UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería
Departamento de Computación

Propuesta de implementación de enseñanza de algoritmos y "Programación sin computadora" en el segundo ciclo de primaria en el sistema educativo guatemalteco.

Trabajo de graduación presentado por Margarita Rosa Suhr Mora y Edna Regina Quinto Méndez para optar al grado académico de Licenciado en Educación, Jeniffer Emilia Mazariegos Morales, Tomás Gálvez Peña, Axel Saul Mayorga Ruballos, Mauricio Juárez Alvarado, Karen Angélica Basegoda Curiel, María Fernanda Bautista Godínez y Samuel Alfredo Chávez Fuentes para optar al grado académico de Licenciado en Ingeniería en Ciencia de la Computación, Oscar Amado Jimenez Zavala y Luis Geovani Rios Rodríguez para optar al grado académico de Licenciado en Ingeniería Industrial.

Guatemala
2011

Propuesta de implementación de enseñanza de algoritmos y "Programación sin computadora" en el segundo ciclo de primaria en el sistema educativo guatemalteco.

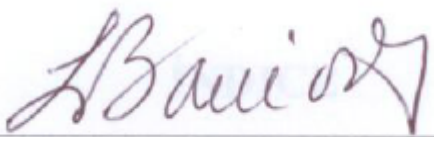
UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería
Departamento de Computación

Propuesta de implementación de enseñanza de algoritmos y "Programación sin computadora" en el segundo ciclo de primaria en el sistema educativo guatemalteco.

Trabajo de graduación presentado por Margarita Rosa Suhr Mora, Edna Regina Quinto Méndez para optar al grado académico de Licenciado en Educación, Jeniffer Emilia Mazariegos Morales, Tomás Gálvez Peña, Axel Saul Mayorga Ruballos, Mauricio Juárez Alvarado, Karen Angélica Basegoda Curiel, María Fernanda Bautista Godínez y Samuel Alfredo Chávez Fuentes para optar al grado académico de Licenciado en Ingeniería en Ciencia de la Computación, Oscar Amado Jimenez Zavala y Luis Geovani Rios Rodríguez para optar al grado académico de Licenciado en Ingeniería Industrial.

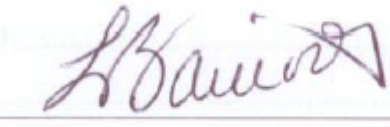
Guatemala
2011

Vo. Bo.:

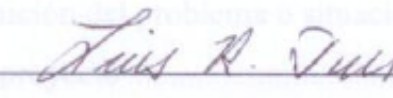
(f) 

Ingeniero Douglas Leonel Barrios

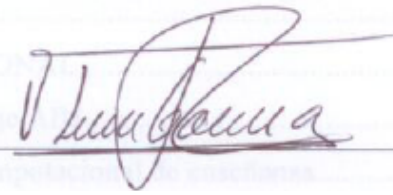
Tribunal Examinador:

(f) 

Ingeniero Douglas Leonel Barrios

(f) 

Ingeniero Luis Furlán Collver

(f) 

Doctora Violeta García Rodríguez

Fecha de aprobación:

Guatemala, 18 de noviembre de 2011

ÍNDICE

ÍNDICE VI
LISTA DE CUADROS VII
LISTA DE ILUSTRACIONES X
RESUMEN XIII
I. INTRODUCCIÓN 1
II. MARCO CONTEXTUAL 3
A. La educación en Guatemala (el contexto educativo) 3
III. MARCO TEÓRICO 10
A. Educación 10
B. Computación 34
C. Industrial 53
IV. MARCO METODOLÓGICO 59
A. Justificación y definición del problema o situación 59
B. Objetivos del Megaproyecto 61
C. Antecedentes 61
D. Metodología 62
V. DISEÑO COMPUTACIONAL 74
A. Paradigma del Lenguaje ABJ 74
B. Diseño del modelo computacional de enseñanza 77
C. Diseño del software 78
D. Gramática 102
E. Diseño del sitio de comunicación (Wiki) 104
F. Selección de Herramientas de implementación 106
VI. RESULTADOS Y DISCUSIÓN 109
A. Resultados cuantitativos 109
B. Resultados cualitativos 109
C. Entregables de software 115
D. Resultados logísticos y administrativos 125
VII. CONCLUSIONES Y RECOMENDACIONES 147

VIII. BIBLIOGRAFÍA 152
IX. APÉNDICE 159
A. Seguimiento de la aplicación de la herramienta 159
X. GLOSARIO 166

LISTA DE CUADROS

Cuadro 1. Porcentaje de estudiantes con respuestas correctas para Matemática de tercer grado de primaria, por proceso cognitivo y por país. 4

Cuadro 2. Porcentaje de estudiantes con respuestas correctas para Matemática de sexto grado de primaria, por proceso cognitivo y por país. Fuente: SERCE, 2007 5

Cuadro 3. Indicadores educativos de cobertura y de eficiencia interna. Fuente: DIPLAN, Ministerio de Educación, 2011 5

Cuadro 4. Distribución de alumnos de nivel primario. Fuente: DIPLAN, Ministerio de Educación, 2011 6

Cuadro 5. Distribución de docentes a nivel nacional. Fuente: http://www.mineduc.gob.gt/estadistica/2008/main.html 7

Cuadro 6. Resumen de los elementos involucrados en el aprendizaje y rol que desempeñan. Fuente: elaboración propia a partir de Bruner (197 2) 16

Cuadro 7. Etapas en el proceso de resolución de problemas. Fuente: elaboración propia a partir de Mayer (1982) 18

Cuadro 8. Porcentaje de estudiantes con respuestas correcta para matemática de sexto grado, por proceso cognitivo y por país. Fuente: (Laboratorio Latinoamericano de Evaluación de la Calidad Educativa (LLECE), 2009) 32

Cuadro 9. Comparación entre ventajas y desventajas del uso de XML. 44

Cuadro 10. Atajos con el teclado 89

Cuadro 11. Experimento de algoritmo propuesto para resolver el problema “¿Cómo hacer limonada?”. 104

Cuadro 12. Presupuesto de gastos 130

Cuadro 13. Actividades fase 4. 130

Cuadro 14. Matriz de marco lógico plan piloto 131

Cuadro 15. Análisis de varianzas por institución y general grupo experimental 132
Cuadro 16. Análisis de varianzas por institución y general grupo control 132
Cuadro 17. Comparación de promedios grupo experimental 132
Cuadro 18. Comparación de promedios grupo control 133
Cuadro 19. Matriz de Marco Lógico a Nivel Nacional. 138
Cuadro 20. Resultados del costeo 139
Cuadro 21. Análisis beneficio / costo 140
Cuadro 22. Costo de capacitaciones y revisiones 161

LISTA DE ILUSTRACIONES

Figura 1. Esquemas claves de la conducta habitual. Fuente: (Comellas, 2002) 24

Figura 2. Objetivos globales del aprendizaje de hábitos de autonomía. Fuente: (Comellas, 2002) 25

Figura 3. Componentes de la autonomía. Fuente: (Comellas, 2002) 26

Figura 4. Fase preparatoria para el aprendizaje. Fuente: (Comellas, 2002) 26

Figura 5. Proceso de apoyo al aprendizaje. Fuente: (Comellas, 2002) 27

Figura 6. Interiorización del hábito. Fuente: (Comellas, 2002) 27

Figura 7. Fases del proceso de traducción. (Aho, Lam, Sethi, & Ullman, 2007) 39

Figura 8. Árbol generativista para una gramática simple en español. X y X' pueden ser cualesquiera elementos de la oración, por ejemplo Nombre, Verbo, Objeto Directo, etcétera. 41

Figura 9. Representación genérica de una arquitectura Modelo Vista Controlador (Wikipedia, 2011). 44

Figura 10. El proceso de planeación de proyectos se muestra a continuación: 53

Figura 11. Gráfica de Gantt 54

Figura 12. Fases de planeación de un proyecto con CPM o PERT. 56

Figura 13. Diseño experimental utilizado para las pruebas. 63

Figura 14. Sistema de Abastecimiento 69

Figura 15. Nube de ideas, utilizada para el diseño de los objetos (sustantivos concretos) y acciones (verbos). 76

Figura 16. Interacciones entre cada uno de los niveles de arquitectura Modelo-Vista-Controlador propuesto como patrón de diseño para el megaproyecto ABJ 79

Figura 17. Diagrama de casos de uso de la solución de software para el Megaproyecto. 80

Figura 18. Proceso de inicialización del Intérprete de lenguaje ABJ. 82

Figura 19. Proceso de traducción interpretada de una entrada en lenguaje ABJ. 82

Figura 20. Proceso de análisis léxico para el intérprete del lenguaje ABJ. 83

Figura 21. Proceso de análisis sintáctico para el intérprete del lenguaje ABJ. 83

Figura 22. Proceso de ejecución semántica para el intérprete del lenguaje ABJ. 84

Figura 23. Diagrama de cajas del ciclo de despliegue de la ventana del problema. 85

Figura 24. Diagrama de clase para el controlador de acciones. 94

Figura 25. Diagrama de clase para el diseño de los escenarios. 97

Figura 26. Primera parte del diagrama de clases propuesto para la solución de software del megaproyecto ABJ. 100

Figura 27. Segunda parte del diagrama de clases propuesto para la solución de software del megaproyecto ABJ. 101

Figura 28. Diagrama de secuencia representando lo sucedido durante la ejecución de la instrucción runCommand 101

Figura 29. Arte propuesta para el diseño gráfico de la Wiki. 104

Figura 30. Muñecos de tusa Fuente: Margarita Suhr M. 110

Figura 31. Algoritmo con dos procesos distintos Fuente: Margarita Suhr M. 111

Figura 32. Ejemplo de árbol sintáctico generado dada la oración de entrada: “voy hacia la derecha 3 pasos” 118

Figura 33. Maquetación final de la interfaz gráfica 120

Figura 34. Botones 120

Figura 35. Escenario 122

Figura 36. Resultado obtenido de los módulos de diseño e implementación de la wiki del proyecto 123

Figura 37. Wiki promocional 124

Figura 38. Página ejemplo sobre la documentación técnica generada para la clase ABJLabel. 125
Figura 39. Actividades fase 1. 126
Figura 40. Gantt Actividades fase 1. 127
Figura 41. Actividades fase 2. 127
Figura 42. Gantt Actividades fase 2. 127
Figura 43. Actividades fase 3. 128
Figura 44. Gantt Actividades fase 3. 128
Figura 45. Gantt Actividades fase 3. 129
Figura 46. Gantt Actividades fase 3. 129
Figura 47. Gantt Actividades fase 3. 129
Figura 48. Gantt Actividades fase 4. 130
Figura 49. Actividades proyecto a nivel nacional 141
Figura 50. Diagrama Gantt 141

RESUMEN

Se ha comprobado que el aprendizaje de algoritmos promueve el desarrollo del pensamiento lógico, la autonomía e iniciativa personal en los educandos, habilidades indispensables para enfrentar los retos del nuevo siglo. Este trabajo contiene los marcos teórico, contextual y metodológico que fundamentan la ejecución del Megaproyecto de la Facultad de Ingeniería en Ciencias de la Computación de la Universidad del Valle de Guatemala: *Propuesta de implementación de enseñanza de algoritmos y Programación sin computadora en el segundo ciclo de primaria en el sistema educativo guatemalteco.* Los productos del Megaproyecto, contenidos en el presente informe, incluyen la herramienta computarizada, así como un manual del docente, un instructivo de capacitación del docente y un cuadernillo de trabajo para uso del alumno. La herramienta fue aplicada en cinco instituciones educativas de la ciudad de Guatemala. Se observó que los alumnos que recibieron el tratamiento experimentaron mejoras en sus hábitos y habilidades de estudio, algunos estudiantes mejoraron su rendimiento académico en varias materias.

Palabras clave: Algoritmos, aprendizaje, pensamiento lógico, autonomía e iniciativa personal, resolución de problemas, programación

I. INTRODUCCIÓN

Eric Mazur, académico de la prestigiosa Universidad de Harvard afirmó en una conferencia dictada en la Universidad del Valle de Guatemala en el año 2010 que <<*la forma tradicional de enseñanza (clase magistral) ha reducido la educación a únicamente la transferencia de información*>>, enfatizando que la educación es <<mucho más que sólo eso>>. Según Mazur, desde la invención de la imprenta, la función de los libros ha sido la transmisión de la información. La clase magistral duplica, en su opinión, esa función, y los niños expuestos a este tipo de práctica educativa pierden la oportunidad de desarrollar competencias y habilidades básicas requeridas por el mundo moderno.

En esa misma línea, la Comisión Internacional sobre Educación, Equidad y Competitividad Económica establece en el informe: *El futuro está en juego* (PREAL, 1998), que la crisis de la educación en América Latina es el resultado de la interacción de cuatro factores, siendo el tercero de ellos una mala calidad de la enseñanza. Se establece que los docentes aplican métodos pedagógicos anticuados, en donde sólo exponen el material para que el niño lo memorice, una práctica que no promueve el pensamiento crítico y la resolución de problemas.

Este informe describe el Megaproyecto propuesto por el departamento de Ciencias de la Computación de la Facultad de Ingeniería, en el cual se realizó y aplicó una herramienta para ayudar a los estudiantes a desarrollar un pensamiento estructurado que mejore en ellos las habilidades y competencias requeridas para la resolución de problemas, a través de la enseñanza y aplicación de algoritmos.

Aunque el Megaproyecto contempla el desarrollo de una herramienta computarizada, también toma en cuenta el contexto educativo nacional e incluye el desarrollo de una herramienta que no requiere el uso de computadoras. De esa forma, el Megaproyecto fue dividido en varios módulos principales que comprendieron el desarrollo de la herramienta computarizada, el desarrollo de la herramienta no computarizada, y la administración general del proyecto.

La herramienta computarizada fue utilizada por 40 alumnos de una institución púbica durante un período de tres meses. La herramienta no computarizada fue utilizada durante ese mismo período de tiempo por 125 alumnos de quinto primaria en un total de cinco instituciones educativas, cuatro públicas y una privada. Los docentes fueron capacitados en su utilización y acompañados con visitas semanales de seguimiento y retroalimentación.

Para medir los resultados de la herramienta, se llevó a cabo una pre evaluación en el mes de abril y una post evaluación al concluir los tres meses, a finales del mes de julio. Como instrumento de evaluación, se utilizaron dos versiones de la prueba de habilidad general Otis Lennon, la cual fue aplicada por personal del Centro de Investigaciones Educativas de la Universidad del Valle de Guatemala.

Este trabajo contiene los marcos teórico, contextual y metodológico que fundamentan la ejecución del Megaproyecto de la Facultad de Ingeniería en Ciencias de la Computación de la Universidad del Valle de Guatemala: *Propuesta de implementación de enseñanza de algoritmos y Programación sin computadora en el segundo ciclo de primaria en el sistema educativo guatemalteco.*Contiene también los productos obtenidos de los distintos módulos que conformaron el Megaproyecto, así como los resultados cuantitativos y cualitativos de su utilización en cinco instituciones educativas de la ciudad de Guatemala.

II. MARCO CONTEXTUAL

A continuación, se presenta una descripción del contexto educativo de Guatemala, que incluye el marco legal, la calidad educativa del país, la reciente reforma curricular, y las nuevas exigencias del proceso enseñanza-aprendizaje de la matemática en el CNB.

A. La educación en Guatemala (el contexto educativo)

1. Marco legal. Los lineamentos básicos de la educación en Guatemala se encuentran contenidos en la Constitución Política de la República de 1985 y en la Ley de Educación Nacional de 1991. Son obligaciones del Estado asegurar el derecho a la educación y su gratuidad (Art. 71 y 74 de la Constitución Política, respectivamente) en los niveles establecidos por la Carta Magna y en la Ley de Educación de 1991 (educación inicial, pre-primaria, primaria y básica), y define como objetivos permanentes del Estado la educación científica, tecnológica y humanística. De la misma forma, el artículo 72 dice que: "La educación tiene como fin primordial el desarrollo integral de la persona humana, el conocimiento de la realidad y cultura nacional y universal." La Ley de Educación Nacional establece como fines de la educación el cultivo de cualidades físicas, intelectuales, morales, espirituales y cívicas de la población, incluyendo ciudadanos capaces de contribuir al fortalecimiento de la democracia con actitud crítica e investigativa para enfrentar cambios sociales.

A pesar de que la ley de Educación Nacional establece que la asignación de recursos del Estado para la educación debe ser del 35% de los ingresos ordinarios del Estado y que la misma debe ser de hasta un 7% Producto Interno Bruto (PIB), en la actualidad los recursos asignados a educación en Guatemala son equivalentes al 13% del presupuesto nacional, y únicamente el 2% del PIB. (Finlandia, considerado el país con la mejor educación de acuerdo a los resultados obtenidos de las evaluaciones internacionales *PISA –Program for International Student Assessment*-- invierte el 6% de su PIB en educación). El Ministerio de Educación distribuye estos recursos por nivel como sigue:

- Preprimaria 8%
- Primaria 66%
- Secundaria 10%
- Superior 9%

Es evidente que los fondos asignados a la educación son limitados por lo que es importante que la herramienta propuesta en este trabajo de graduación sea de bajo costo, a la vez que permita alcanzar las habilidades esperadas.

2. Calidad educativa. <<Escuela de calidad es la que promueve el progreso de sus estudiantes en una amplia gama de logros intelectuales, morales y emocionales, teniendo en cuenta su nivel socioeconómico, su medio familiar y su aprendizaje previo. Un sistema escolar eficaz es el que maximiza la capacidad de las escuelas para alcanzar esos resultados>> (Mortimore, 1998)

En el año 2006, el Laboratorio Latinoamericano de Evaluación de la Calidad Educativa (LLECE) realizó en Guatemala el *Segundo Estudio Regional Comparativo y Explicativo* SERCE que incluyó el área de matemática, dirigido a medir no sólo los conocimientos de los alumnos de tercero y sexto grado de Primaria sino también su capacidad para aplicar dichos conocimientos para comprender e interpretar el mundo, en distintas situaciones y contextos cotidianos.

En este estudio, los resultados de Guatemala están por debajo de la media con relación al resto América Latina y el Caribe, como lo ilustran los cuadros que siguen.

Cuadro 1. Porcentaje de estudiantes con respuestas correctas para Matemática de tercer grado de primaria, por proceso cognitivo y por país.

		Resolución de problemas	
País	**Reconocimiento de objetos y elementos**	**Simples**	**Complejos**
Cuba	80,29	70,02	83,12
Guatemala	43,36	31,32	32,91
El Salvador	51,65	36,92	36,75

Fuente: SERCE, 2007

Cuadro 2. Porcentaje de estudiantes con respuestas correctas para Matemática de sexto grado de primaria, por proceso cognitivo y por país.

País	Reconocimiento de objetos y elementos	Resolución de problemas	
		Simples	Complejos
Cuba	74,57	65,17	53,50
Guatemala	47,88	32,88	23,58
El Salvador	53,33	35,66	28,01

Fuente: SERCE, 2007

De toda América Latina y el Caribe, solamente Cuba supera la media de la región en más de una desviación estándar; y, Guatemala se encuentra con un rendimiento por debajo de la media de la región en todas las áreas. Según las conclusiones del informe, <<las escuelas con bajo rendimiento y bajo nivel socioeconómico y cultural suelen tener débiles capacidades para promover el aprendizaje. En estas escuelas es frecuente observar que los docentes tienen una formación insuficiente, con poca especialización para atender estudiantes en contextos vulnerables y un limitado repertorio de prácticas de enseñanza para atender las necesidades educativas individuales de los estudiantes. Por ello, se pueden impulsar políticas educativas encaminadas a mejorar la capacidad de los docentes para promover el aprendizaje>> (UNESCO, 2010).

En el siguiente cuadro se describen los indicadores educativos de Guatemala para el año 2010:

Cuadro 3. Indicadores educativos de cobertura y de eficiencia interna.

Indicador: Tasa	Preprimaria	Primaria	Básico	Diversificado
Bruta de Escolaridad	70.06%	116.21%	70.90%	36.71%
Neta de Escolaridad	54.87%	95.81%	42.94%	22.33%
de Repitencia	0.00%	11.91%	2.96%	0.91%
de Deserción	9.23%	5.99%	10.29%	11.95%
De Promoción	100.00%	85.08%	66.23%	74.36%

Fuente: DIPLAN, Ministerio de Educación, 2011

3. Alumnos. De acuerdo a cifras correspondientes al año 2010 proporcionadas por el Ministerio de Educación, el gobierno atiende a una población escolar de 3,265,022, con un presupuesto anual de 9.162 mil millones de quetzales.

Cuadro 4. Distribución de alumnos de nivel primario.

Nivel	**Total nacional**	**Hombres**	**Mujeres**
Preprimaria	577,766	290,405	287,361
Primaria	2,653,483	1,366,560	1,286,923
Ciclo Básico	730,923	388,074	342,849
Ciclo Diversificado	351,397	176,799	174,598
Nacional	**Total**	**Urbano**	**Rural**
No indígena	2,889,122	1,418,353	1,470,769
Indígena	1,424,477	289,649	1,134,798

Fuente: DIPLAN, Ministerio de Educación, 2011

Para lograr una herramienta que sea pertinente al contexto cultural del estudiante, debe observarse que la gruesa de estudiantes se encuentra en el área rural, de los cuales un porcentaje considerable de éstos son indígenas. Únicamente el 24% del total asiste a instituciones privadas, municipales o cooperativas.

Es conveniente, para fines del presente trabajo profesional, tomar en cuenta las palabras del economista César García (2007): <<El hambre y la desnutrición crónica infantil tienen connotaciones mucho más graves que las que sus ojos alcanza a observar…un crimen del que todos somos culpables y todos somos víctimas>>.

Si un niño no recibe una adecuada nutrición durante los primeros tres años de su vida, el daño a su desarrollo craneal es irreversible. Queda limitada la capacidad del niño para aprender y para alcanzar la inteligencia que genéticamente estaba dispuesta en él. La desnutrición crónica en Guatemala, según datos de UNICEF para el 2007 es del 48.7% (UNICEF: El Enemigo Silencioso), la más alta en América Latina en comparación con Honduras (36.2%), El Salvador (23.3%), Haití (22.7%), Costa Rica (6.1%), entre otros.

La herramienta desarrollada tomó en cuenta la población que la utilizará: una población culturalmente muy diversa con numerosas deficiencias educativas que se ven agravadas por las limitaciones físicas causadas por la desnutrición crónica infantil.

4. Docentes. De igual manera, de acuerdo con las estadísticas de MINEDUC del año 2008 para el nivel primario, los beneficiarios indirectos del aporte de este trabajo de graduación, los docentes, estarían distribuidos de la siguiente manera:

Cuadro 5. Distribución de docentes a nivel nacional.

Docentes	Total nacional	Sector público	Sector privado
General	84,980	70,892	13,998
Indígena	24,516	23,104	1,412
% alumnos por docente	29.43	31,25	20,18

Fuente: http://www.mineduc.gob.gt/estadistica/2008/main.html

De acuerdo con las estadísticas de MINEDUC, cerca del 29% son indígenas, de los cuales casi el 94% labora en el sector público. Es necesario tomar en consideración las características de los docentes, su contexto y cultura, ya que ellos serán los que incluyan la enseñanza de algoritmos en sus aulas y la utilización del cuadernillo por los alumnos.

5. La población docente y sus conocimientos en el área matemática. Guatemala es uno de los dos países de América Latina en donde el docente aún no es egresado de una institución universitaria: obtiene su título al egresar del nivel secundario. El doctor en Educación Óscar Hugo López, director de la Escuela de Formación de Profesores de Enseñanza Media de la Universidad de San Carlos de Guatemala, expresó en entrevista de prensa en el año 2010:<<También hay que darle atención a la crisis de paradigmas, porque estamos acostumbrados a que el maestro es el que sabe y el estudiante no, a que el maestro da la clase y el estudiante escucha, a que el maestro dicte y el estudiante escriba. No se les enseña a pensar, ser crítico, opinar, reflexionar, sino que simplemente a obedecer y seguir instrucciones. >> (López, 2010). Propone que la formación del maestro esté enfocada dentro del contexto cultural y que promueva en los estudiantes el pensamiento crítico y las habilidades para la vida.

El licenciado Salvador Paiz, presidente de la organización Empresarios por la Educación, describe las deficiencias en el área matemática de los docentes del sistema educativo guatemalteco como sigue:

- Solamente la mitad de los docentes son capaces de: Identificar valor absoluto y relativo, calcular área y perímetro, operaciones básicas y promedio aritmético.
- 49% de los docentes son capaces de usar conceptos de: figuras geométricas, plano cartesiano, fracciones, múltiplos, números romanos, números primos y compuestos.
- 49% de los docentes son capaces de realizar operaciones usando conceptos de: suma y resta con potenciación, porcentajes, regla del tres, cálculos aritméticos combinados, área, medidas, decimales, subconjuntos, suma y resta de fracciones.

6. Reforma curricular. También deben mencionarse que dentro de los compromisos contenidos en los Acuerdos de Paz está la Reforma Educativa, la cual contiene las políticas y estrategias que buscan la transformación del ciudadano para la construcción de una sociedad incluyente, solidaria y más justa, en la que toda persona contribuya a la construcción del bien común, mejorando así la calidad de vida de cada guatemalteco. Se busca lograr esto por medio de la transformación curricular, que contempla <<la actualización y renovación técnico pedagógica de los enfoques, esquemas, métodos, contenidos y procedimientos didácticos; de las diversas formas de prestación de servicios educativos y de la participación de todos los actores sociales. >> (CNB, 2007) El nuevo CNB contiene cambios profundos. Su eje principal es el individuo con una visión intercultural y bilingüe, en su contexto sociocultural. Además, el nuevo CNB incluye el diseño y desarrollo curricular basado en competencias.

Dentro de los aspectos más importantes de la transformación curricular que el Megaproyecto ABJ apoya es <<el impulso a procesos educativos basados en el aprender a hacer, aprender a conocer y pensar, aprender a ser, aprender a convivir y aprender a emprender.>>

7. Nuevas exigencias del proceso enseñanza-aprendizaje de matemática en el CNB. El CNB distingue dos áreas principales que se denominan en fundamentales y de formación. La primera se refiere a las áreas base para la construcción de nuevos aprendizajes e incluyen el conocimiento de las ciencias, artes y tecnologías. La segunda se refiere al desarrollo de habilidades para la vida, entre ellas el desarrollo de destrezas para el aprendizaje y formación.

El nuevo CNB establece que el aprendizaje de las matemáticas debe estar enfocado al desarrollo del pensamiento crítico, lógico y creativo. Las actividades aplicadas en el aula deben ser parte un proceso constructivo y explorador para el alumno, el cual se logra con la práctica de ejercicios prácticos y dinámicos. Es fundamental que el proceso de enseñanza aprendizaje de matemática esté relacionado con el contexto real de los alumnos.

Tomando en cuenta todos los aspectos anteriores que enmarcan el contexto educativo de Guatemala, la herramienta a desarrollar deberá considerar todos estos factores del contexto educativo nacional: que esté de acuerdo con los fines de la educación establecidos en la Ley de Educación Nacional, que apoye la transformación curricular contenida en el nuevo CNB, que considere los limitados recursos económicos asignados al Ministerio de Educación y que tome en cuenta la población docente y estudiantil a la que está dirigida: una población culturalmente muy diversa con numerosas deficiencias educativas que se ven agravadas por las limitaciones físicas causadas por la desnutrición crónica infantil.

III. MARCO TEÓRICO

A. Educación

A continuación se presenta la sustentación teórica para el desarrollo de la herramienta no computarizada, la cual fue desarrollada por miembros dela Facultad de Educación. Incluye las principales teorías constructivistas, la base teórica sobre la resolución de problemas matemáticos y el proceso de enseñanza aprendizaje de algoritmos.

1. Las teorías constructivistas. El constructivismo es un conjunto de teorías educativas que tienen como denominador común el concepto de que es el individuo el que "construye" su conocimiento basado en su interacción con su medio ambiente. Cattaneo (s.f.) define que el constructivismo es aquél en donde <<el conocimiento es el resultado de un proceso de construcción o reconstrucción de la realidad que tiene su origen en la interacción entre las personas y el mundo. Por tanto, la idea central reside en que la elaboración del conocimiento constituye una modelización más que una descripción de la realidad. >> (Cattaneo, s.f.). A continuación se describen las teorías constructivistas más relevantes para este trabajo de graduación.

a. Piaget. La teoría de Jean Piaget explica el desarrollo cognoscitivo del niño desde su nacimiento hasta llegar a adulto, haciendo énfasis en la formación de estructuras mentales. Plantea la importancia de comprender la formación de mecanismos mentales del niño como una forma para conocer su naturaleza y funcionamiento en la edad adulta.

Establece que la formación del pensamiento se desarrolla de forma progresiva hasta alcanzar un equilibrio en la edad adulta. Ahora bien, ese equilibrio está en constante cambio, y sufre de modificaciones constantes, pues se ve influenciado por las actividades del individuo que varían según la edad. (Riego Gaora, 1986).

Riego (1986) cita a Piaget, quien dice que toda actividad es impulsada por una necesidad, y que ésta, no es otra cosa que un desequilibrio, por lo que toda actividad tiene como finalidad principal recuperar el equilibrio perdido. En este desequilibrio surgen los constructos de la teoría piagetiana conocidos como "asimilación y acomodación". Una vez sucede un desequilibrio, el individuo lleva a cabo un proceso cognitivo donde "asimila" esa situación que produjo el cambio para poder llevar a cabo modificaciones en sus estructuras cognitivas para "acomodar" esos elementos, creando así nuevas estructuras cognitivas necesarias para lograr recuperar el equilibrio perdido.

La asimilación cognitiva puede compararse a la función que realiza el cuerpo humano con los alimentos que se ingieren. Se toma de ellos las sustancias nutritivas que necesita el cuerpo con el propósito de satisfacer las necesidades fisiológicas. Piaget define el término "asimilar" como: <<... incorporar las cosas y las personas a la actividad propia del sujeto y, por consiguiente,... 'asimilar' el mundo exterior a las estructuras ya construidas...>> (Riego Gaora, 1986).

La función de acomodación es complementaria a la de asimilación. Una vez que las experiencias han sido incorporadas a las estructuras cognitivas del sujeto, es necesario modificar dichas estructuras, es decir, <<...reajustar (las estructuras construidas) en función de las transformaciones sufridas, y, por consiguiente, a 'acomodarlas' a los objetos externos>>. (Riego Gaora, 1986).

1) Etapas del desarrollo del niño. Las observaciones hechas por Piaget, lo llevó a concluir que el desarrollo cognoscitivo de un individuo se define por medio de cuatro etapas principales; cada etapa es el resultado de la interacción entre la maduración física y el medio ambiente en donde se desenvuelve el individuo. Cada una de las cuatro etapas está relacionada hasta cierto grado con la edad cronológica de la persona, aunque la misma debe considerarse aproximada, reconociendo la diversidad que cada individuo tiene en su propio desarrollo.

A continuación se describe brevemente cada una de las etapas. Se hará énfasis, por la naturaleza del presente trabajo de graduación, en la etapa de las operaciones concretas, que corresponde a niños desde los siete a los once años.

a) Etapa sensoriomotora. Empieza con el nacimiento y termina alrededor de los dos años. En su interacción con el medio ambiente, el niño primero responde a los reflejos, hasta llegar, al final de esta etapa, al aprendizaje por medio de prueba y error y a la resolución de problemas sencillos.

b) Etapa preoperacional. Ocurre entre los dos y los siete años de edad. El niño empieza a utilizar símbolos que representan objetos y personas de su entorno. Son egocéntricos y desde el punto de vista de un adulto, ilógicos.

c) Etapa de operaciones concretas. Ocurre entre los siete años y once años. El niño empieza a desarrollar el pensamiento lógico, ya comprende los conceptos que utiliza de forma sistemática que le permiten una mejor comprensión e interacción con su medio ambiente.

d) Etapa de operaciones formales. Sucede entre los doce y los quince años. En esta etapa el niño desarrolla el pensamiento abstracto. En esta etapa el adolescente puede relacionarse en situaciones hipotéticas como reales.

En la etapa de las operaciones concretas el niño ya puede utilizar símbolos en sus actividades mentales. Es capaz de clasificar y manejar números. Utiliza representaciones mentales de las cosas y los hechos y también es capaz de tomar todos los aspectos de una situación y sacar conclusiones de ellos. Comprende la característica reversible de las operaciones físicas.

El egocentrismo del niño empieza a disminuir en esta etapa y empieza a percibirse como algo distinto del resto del universo y de otras personas. Eso le permite una mejor comprensión de los puntos de vista de otros. Esto último también hace que mejore su capacidad de comunicarse. Su pensamiento es ahora más estable y lógico, pero no posee aún la capacidad de manejar conceptos abstractos.

b. Vigotsky. Lev Vigotsky (1896 – 1934) fue un destacado psicólogo bielorruso, teórico de la psicología del desarrollo. La Guerra Fría impidió que sus trabajos fueran conocidos por el mundo occidental, hasta hace aproximadamente 20 años. La idea principal de su teoría es que el desarrollo del ser humano se entiende solamente cuando éste es relacionado con su interacción social. El desarrollo del individuo es el resultado de la interiorización de elementos culturales (como el idioma, por ejemplo), que son parte del grupo humano en donde nace.

1) Andamiaje. Según Cattaneo (s.f.), la teoría de Vigotsky afirma que durante el aprendizaje se da una interacción colaborativa entre un individuo que tiene mayor experiencia y otro de menor experiencia y cuya finalidad es que el novato se convierta en experto. Definió esta situación como "andamiaje." Sus características principales son:

- Es ajustable: es decir, debe tomar en cuenta el nivel de competencia del novato y lo que vaya logrando.
- Es temporal: no puede ser permanente, porque se deja de cumplir el objetivo de que el novato sea autónomo.
- Es audible y visible: Esto le permite al novato reconocer y lograr un control progresivo sobre las actividades de aprendizaje.

2) Zona de desarrollo próximo. Vigotsky observó que existe un espacio entre las habilidades adquiridas del individuo y su potencial. Elabora, pues un constructo a su teoría

que define como Zona de Desarrollo Próximo (ZDP), que Caldeiro (2005) define como: <<… la distancia entre el nivel real de desarrollo determinado por la capacidad de resolver independientemente el problema y el nivel de desarrollo potencial determinado a través de la resolución de un problema bajo la guía de un adulto o en colaboración con otro compañero más capaz. >> (Caldeiro, 2005)

Considerada la ZPD como un rasgo básico del aprendizaje, Vigotsky establece que el aprendizaje estimula procesos de desarrollo internos del individuo que no ocurren sino hasta que el individuo se relaciona con personas de su entorno y con la asistencia de otra persona o guía. Indica que en el momento en que estos procesos se interiorizan, pasan a ser logros evolutivos independientes del niño, que luego se realizan sin necesidad de tal asistencia.

3) El juego y la zona de desarrollo próximo. Según Cattaneo, Vigotsky considera que la actividad lúdica es una de las formas como un niño participa mejor de la cultura, ya que el juego constituye una de sus actividades principales. Cuando una actividad lúdica crea una situación imaginaria que se rige a determinadas reglas de conducta, se crea una ZDP. De esta manera, se identifica:

- Un escenario imaginario en donde se representan roles pertinentes al medio ambiente del niño.
- La presencia de reglas socialmente establecidas.
- La presencia de roles que cumplen su función dentro de un contexto.

Indica Cattaneo que el juego adquiere una fuerza motriz ya que permite a los niños situarse en diversos roles sociales del mundo adulto, sujetándose a la observancia de determinadas reglas sociales.

c. Ausubel. El concepto de aprendizaje significativo desarrollado por Ausubel es un componente importante de las teorías constructivistas. A continuación se detallan las partes más importantes de su aporte.

1) Aprendizaje significativo. El aprendizaje significativo es aquél en donde el alumno relaciona de forma significativa los nuevos conceptos con conocimientos previos. La teoría establece que la información es retenida por más tiempo, al mismo tiempo que facilita la adquisición de nuevos conocimientos.

Ausubel diferencia tres tipos de aprendizaje significativo, donde los últimos dos dependen del primero:

- Aprendizaje de representaciones: Se refiere al aprendizaje del vocabulario. El niño aprende palabras que representan objetos reales que tienen significado para él.
- Aprendizaje de conceptos: Se aprende que un concepto está representado por una nueva palabra.
- Aprendizaje de proposiciones: Se aprende el significado de nuevas ideas expresadas por medio de proposiciones.

Según Ausubel, el aprendizaje significativo, requiere de dos condiciones: a) disposición del sujeto a aprender significativamente y b) material de aprendizaje que tenga sentido lógico y donde el aprendiz posea conocimientos previos relevantes con los cuales pueda relacionarse el material nuevo.

Ausubel describe la incorporación de la nueva información a la ya existente de tres formas: derivativa (sólo un ejemplo de las ideas que ya tiene una personas), correlativa (se extiende o modifica una idea), o combinatorial (no está relacionada con una idea o ideas específica(s), sino con el fondo general del conocimiento).

Establece que para que se suceda un aprendizaje significativo, los nuevos contenidos deben presentarse respetando lo que define como "diferenciación progresiva", en donde deben presentarse las ideas generales primero y luego las específicas o particulares y la reconciliación integradora, donde los conocimientos ya presentes se reorganizan y adquieren un significado distinto.

Beltrán (2002) establece que «La idea central de la teoría de Ausubel es, pues, que la información nueva, potencialmente significativa, se incorpora dentro de la estructura cognitiva del estudiante. Para ello es imprescindible la existencia de una estructura cognitiva y, cuando ésta no existe hay que recurrir a los organizadores previos.» (Beltrán 2002:18)

d. Bruner. «El curriculum debería entrañar el dominio de destrezas que a su vez conducirían a otras más fuertes, al establecimiento de series autorecompensadoras» (Bruner, 1972)

Jerome Bruner, psicólogo estadounidense, nació en Nueva York el 1 de octubre de 1915. Se graduó de la Universidad de Duke en 1937 para luego continuar sus estudios de doctorado en la Universidad de Harvard, graduándose en doctor en Psicología en 1941.

Impartió la cátedra de Psicología en la Universidad de Harvard de 1952 a 1972, y luego en la Universidad de Oxford de 1972 a 1980. En 1960 fundó el Centro de Estudios Cognitivos de la Universidad de Harvard, dándole un fuerte impulso a la psicología cognitiva. Su contribución más relevante hacia el constructivismo es su Teoría de la instrucción.

1) Teoría de la instrucción. Bruner establece en su teoría que <<el principal hincapié de la educación debería hacerse en las destrezas: habilidad para manipular (enactiva), para ver e imaginar (icónica), y en las operaciones simbólicas (simbólica). >> Ejemplo: Tema: el equilibrio (física), Enactiva: columpiarse Icónica: un dibujo simbólica: F= 3.1 y M=10.

El autor describe las cuatro características de su teoría de instrucción:

1. Describir las experiencias más efectivas que permiten que la persona posea una predisposición a aprender, que responda pregunta, pe.: ¿Qué cosas en el medio ambiente preescolar tenderán a hacer que el niño esté dispuesto y sea capaz de aprender cuando entre a la escuela? Se consideran predisposiciones los factores culturales, motivacionales y personales.

2. Especificar las formas en que un conjunto de conocimientos deben estructurarse para que pueda ser comprendido por el alumno de la forma más rápida ("estructura óptima"). Una estructura efectiva deberá simplificar la información, generar nuevas proposiciones y aumentar la manipulación de un cuerpo de conocimientos y debe ser acorde a la edad y capacidades del niño.

3. Debe especificar el orden en que los contenidos deben ser presentados.

4. Debe especificar la naturaleza (intrínseca o extrínseca) y tiempo (inmediato o diferido) de las recompensas y castigos en el proceso de enseñanza aprendizaje.

2) Constructivismo social. Bruner afirma que la mejor forma de adquirir el conocimiento es teniendo la experiencia personal de descubrirlo. El aprendizaje se concibe como un proceso social, y se convierte en aprendizaje significativo cuando se logra llevarlo a sus relaciones sociales.

El cuadro siguiente resume los elementos involucrados en el aprendizaje y el rol que desempeñan:

Cuadro 6. Resumen de los elementos involucrados en el aprendizaje y rol que desempeñan.

Elemento	Rol
Docente	• Facilita el aprendizaje • Media entre el alumno y el contexto • Guía en la presentación de problema real o simulación. • Anima al alumno a descubrir por sí mismo • Propicia la colaboración entre estudiantes
Estudiante	Una vez que se alcanza la Concepción Humana dentro de un contexto social: • Participa activamente al sugerir, construir su propio aprendizaje (basándose en sus propias experiencias) y crear • Desarrolla su ser como persona • Logra un aprendizaje significativo.
Medio	• Es real y externo a la persona • Propicia una presentación mental de la realidad • Se incluyen las experiencias previas y presentes • Lleva a un aprendizaje significativo que luego retorna a lo real y externo.
Currículo	• La metodología es un conjunto de técnicas, procedimientos y estrategias que contemplan el aprendizaje colaborativo. Debe ser variada y relacionar a todos los miembros. Tener en cuenta en todo momento los conocimientos previos. • El contenido es un recurso, un punto de partida para que el estudiante adquiera sus propios conocimientos y no los limita a sólo algunos aspectos. • Los objetivos, son metas concretas que indican un nivel de logro. Hacen énfasis en: transferencia de información, desarrollo de habilidades, cambio de estado y de procesos. • La evaluación es un proceso que permite: la toma de decisiones, buscar causas e intereses, monitorear el proceso y el resultado. Se aplica de manera diagnóstica, formativa, sumativa. Utiliza coevaluación, heteroevaluacion, auto evaluación.

Fuente: elaboración propia a partir de Bruner (197 2)

2. La resolución de problemas

a) La psicología cognitiva en la resolución de problemas. La psicología cognitiva es la ciencia que estudia la forma como los seres humanos adquieren, procesan, guardan y manejan el conocimiento. Busca responder varias interrogantes, entre ellas: "¿cómo piensan las personas?" y "¿cómo se desarrollan las destrezas de pensamiento?" Las investigaciones que se han llevado a

cabo en el área cognitiva relacionadas con el aprendizaje de la matemática replantea estas interrogantes así: "¿Cómo piensan las personas en la matemática?" "¿Cómo se desarrolla la comprensión de conceptos matemáticos?"

Resnick y Ford (1981) señalan que la psicología de la resolución de problemas matemáticos requiere de combinar la comprensión del tema estudiado con el conocimiento sobre cómo piensan las personas en general. La psicología cognitiva ha señalado varios aspectos que deben considerarse al estudiar los procesos de pensamiento para el desarrollo de destrezas matemáticas. A continuación se presentan los más relevantes para este proyecto.

Mayer (1982) divide su investigación en tres aspectos primordiales. El primero de ellos se refiere al hecho de que la psicología cognitiva considera el aprendizaje como la adquisición de nuevos *conocimientos,* no sólo de nuevos comportamientos.

Los tipos de conocimiento necesarios para la resolución de problemas son:

- El lingüístico y de hechos, que está relacionado con la comprensión del idioma y el conocimiento de estructuras gramaticales.
- El de esquemas, que se refiere a las relaciones entre los distintos tipos de problemas.
- El de algoritmos: procedimientos que deben aplicarse en los distintos tipos de cálculo, como la suma, por ejemplo.
- El estratégico: cómo abordar un problema.

El segundo aspecto es identificar la diferencia entre las dos etapas del proceso de resolución de problemas: (1) la representación (la comprensión del problema) y (2) la solución (búsqueda del camino). La primera implica un proceso mental que interioriza el problema en su estado actual que busca alcanzar un logro o una meta y los operadores que lo permiten. El segundo examina los distintos caminos hacia la resolución del problema. El estudiante aplica distintos operadores hasta alcanzar la meta.

El siguiente cuadro resume estas dos etapas y muestra cómo son utilizados los tipos de conocimiento en cada etapa del proceso.

Cuadro 7. Etapas en el proceso de resolución de problemas.

Tipo de Conocimiento	Representación (Comprensión)	Solución (Búsqueda)
Lingüístico y de hechos	X	
Esquemático	X	
Algorítmico		X
Estratégico		X

Fuente: elaboración propia a partir de Mayer (1982)

Las investigaciones han mostrado que las dificultades que se encuentran en la resolución de problemas matemáticos a menudo resultan de una representación inadecuada del problema (etapa 1). Sin embargo, Mayer (1982) establece en su informe que la mayor parte del tiempo de clase se dedica a la enseñanza de procedimientos de solución, en vez de a la *aplicación* de esos procedimientos o a la representación de los problemas.

El tercer aspecto señalado por Mayer (1982) indica que la enseñanza para la resolución de problemas debe enfocarse más en *el proceso,* por ejemplo, cómo resolver problemas, que en el *producto,* o la respuesta final. Se ha observado que los estudiantes pueden aprender a efectuar procesos matemáticos utilizando distintos conocimientos y lograr las mismas respuestas correctas. Este es el aprendizaje significativo: una situación en donde el procedimiento para la resolución de problemas está relacionado con otros conocimientos, y es comprendido. La evidencia muestra que el aprendizaje significativo permite una mejor transferencia hacia niveles superiores de pensamiento, así como a una mejor retentiva.

b. La organización del conocimiento en la resolución de problemas. <<Un conjunto bien equipado y bien organizado de conocimientos es un activo para el solucionador de problemas. La buena organización que permite que el conocimiento esté inmediatamente disponible es tal vez más importante que la cantidad de conocimiento. >> (George Polya, citado por Nieto, 2004: 14).

Cuando se habla de la organización del conocimiento, se deben abordar dos fenómenos cognitivos: los esquemas matemáticos, la elaboración cognitiva y los metacognitivos: los sistemas de selección de estrategias y creencias. Los esquemas se definen como el conjunto de

conocimientos que describen las propiedades del concepto que representan. La noción de estos esquemas ayuda en la comprensión de muchos aspectos sobre la organización del conocimiento y memorización humana. El proceso de categorización involucra interpretación y clasificación de la información, además permite el acceso a información previamente procesada, lo cual es propio de cada individuo. Por ejemplo, se ha demostrado que el pasaje de un texto a menudo se presta a distintas interpretaciones, las cuales son el reflejo de los esquemas propios de cada lector.

Una manera de procesar nueva información es utilizando la elaboración cognitiva. Cuando se utiliza esta estrategia, el estudiante hace una construcción simbólica que combina la nueva información con sus conocimientos previos para hacerla significativa. A su vez, incluye la construcción de relaciones lógicas entre los distintos componentes de la información. Esto permite que la nueva información sea más significativa, estableciendo una relación con el conocimiento previamente adquirido. Las investigaciones han señalado también que las diferencias individuales (genéticas), de desarrollo temprano (ambientales) y la forma como ambas son relacionadas pueden ser el resultado de la capacidad de cada individuo para elaborar conocimientos. Un ejemplo de cómo se puede ayudar a un estudiante en su proceso de elaboración es comparando un problema dado con otro similar o distinto que se haya enseñado previamente.

La metacognición se refiere al conocimiento propio de los procesos y productos cognitivos (de la persona) y todo lo relacionado con ellos. Se refiere no sólo a la conciencia de cómo se dan estos procesos sino también a su monitoreo, regulación, evaluación y dirección. En años recientes se le ha dado atención a la manera en que la metacognición influye en la solución de problemas matemáticos.

Al enfrentarse con un problema, el estudiante puede elegir entre diversas estrategias para encontrar la respuesta. Estas decisiones son de naturaleza metacognitiva e influyen en la forma cómo la información del problema es clasificada y qué otra información útil (conocimientos) se extrae de la memoria para ayudar a la resolución del mismo. La toma de decisiones en cuanto a la utilización de determinadas estrategias está influenciada por las creencias y valores propios de la persona. Esto quiere decir que un estudiante examinará casos y buscará patrones únicamente si en realidad él cree que una determinada estrategia podría tener alguna posibilidad de éxito.

Varias investigaciones antropológicas y sociológicas han demostrado que las creencias culturales que posee cada individuo tienen una profunda influencia en la memoria, la percepción y el conocimiento. Muchos componentes de estas creencias y la visión que la persona tiene del

mundo son de naturaleza metacognitiva. Estos son el resultado de grupos de experiencias cognitivas o son adquiridos en procesos de culturalización y socialización. En el contexto matemático, las creencias que un estudiante tenga sobre su proceso de aprendizaje y de resolución de problemas en particular pueden ser agentes que influyen al clasificar y recordar contenidos matemáticos. Un estudiante que *sabe* que existe una estructura matemática fundamental que es más importante que los detalles superficiales, tendrá una perspectiva del estudio de los contenidos matemáticos muy diferente al estudiante que carezca de esta certeza.

c. La metodología de Polya. Polya (1971) establece una metodología de cuatro etapas para resolver problemas. La primera etapa se refiere a la comprensión del problema. Es imposible resolver un problema si no se comprende el enunciado. La segunda etapa no sólo está relacionada con los conocimientos, sino también con la imaginación y la creatividad. Polya busca llevar a la persona a un terreno que le es familiar. La tercera etapa se refiere más a la parte técnica: emplear conocimientos y procesos para resolverlo. Polya dice que la cuarta etapa es con frecuencia ignorada, pero se debe corroborar y revisar tanto la solución como los pasos.

En cada etapa aparecen una serie de preguntas y sugerencias que estimulan el pensamiento divergente. Las mismas se resumen a continuación:

- **Etapa I.** Comprensión del problema: ¿Cuál es la incógnita? ¿Cuáles son los datos? ¿Cuál es la condición? ¿Es la condición suficiente, insuficiente, redundante o contradictoria para determinar la incógnita?

- **Etapa II.** Concepción de un plan: ¿Se ha encontrado con un problema semejante? ¿Ha observado si es el mismo problema planteado en forma ligeramente diferente? ¿Conoce un problema relacionado con éste? ¿Conoce algún teorema que le pueda ser útil? Observe atentamente la incógnita e intente recordar un problema que le sea familiar y que tenga una incógnita igual o similar. ¿Recuerda algún problema relacionado con éste que haya resuelto anteriormente? ¿Podría utilizarlo? ¿Podría utilizar su resultado o método? ¿Podría utilizarlo introduciendo algún elemento adicional? ¿Podría enunciar o plantear el problema en forma diferente? Si no puede resolver el problema propuesto, trate primero de resolver uno similar. ¿Habrá un problema análogo, uno más accesible, uno más general, o uno más particular? ¿Puede resolver una parte del problema? ¿Y si considera sólo una parte de la condición, en qué medida puede determinarse ahora la incógnita? ¿Puede usted deducir algún elemento útil de los datos? ¿Puede cambiar la incógnita y/o los datos de

manera que la nueva incógnita y los nuevos datos están más cercanos entre sí? ¿Ha utilizado todos los datos? ¿Ha considerado todas las nociones esenciales del problema?

- **Etapa III.** Ejecución del plan: Al ejecutar el plan, compruebe cada uno de los pasos. ¿Puede observar que el paso está correcto? ¿Puede demostrarlo?

- **Etapa IV.** Visión retrospectiva: ¿Puede usted verificar el resultado y/o el razonamiento? ¿Puede obtener el mismo resultado de forma diferente? ¿Puede emplear el resultado o el método en algún otro problema?

d. Factores afectivos y bloqueos mentales. Los factores afectivos no deben excluirse en el desarrollo de habilidades para la resolución de problemas. También las emociones y el deseo de resolver un problema influyen en el desarrollo de éstas y otras habilidades.

Se ha observado que la incapacidad que se observa en algunos alumnos puede ser el resultado de la falta de interés y/o motivación. En ocasiones el alumno no tiene el deseo de comprender el problema y por lo tanto, menor es el deseo de resolverlo.

Otras razones afectivas podrían ser una baja autoestima, y quienes la sufren no desean ni siquiera intentar resolver problemas pues consideran desde un inicio, que no serán capaces de hacerlo. Por el contrario, un alumno que confía en su capacidad y considera los problemas como desafíos que valen la pena enfrentar, tendrá una mejor posibilidad de encontrar exitosamente la solución. Alcanzar esa meta producirá en él una satisfacción emocional, y con esto logrará fortalecer su autoestima y de forma espiral, estar en condiciones afectivas adecuadas para enfrentar nuevos retos.

3. Aprendizaje de algoritmos. De acuerdo a López (2009) los orígenes del término "algoritmo" se encuentran en el nombre del matemático persa Mohamed ibn Musa al Khwarizmi" (825 d.C.) La traducción de su apellido (Khwarizmi) al latín es Algorismus y al español, Algoritmo. El pensamiento matemático se vio influenciado por sus contribuciones al álgebra, a la aritmética y a las tablas astronómicas.

Numerosos algoritmos se encuentran en el mundo natural, y con frecuencia pasan desapercibidos por el individuo por la misma naturalidad con la que suceden. El proceso digestivo es un algoritmo, como lo es el proceso de gestación, las estaciones y los ciclos cósmicos. El área científica que más ha utilizado los algoritmos es la matemática, utilizando esos

procesos como estructura para la solución de problemas. Algunos ejemplos de ellos son el teorema de Pitágoras y la suma y resta de fracciones. Estos constituyen pasos secuenciales, que permiten llegar a una solución, independientemente del conjunto de datos con que se cuenta.

López (2009) define el término algoritmo como <<una secuencia ordenada de instrucciones, pasos o procesos que llevan a la solución de un determinado problema>> o a la realización de una tarea determinada. Pueden ser tan sencillos y cotidianos como seguir una receta de cocina o la realización una tarea.

A continuación se detalla un algoritmo para "lavarse los dientes":

- Tomo la pasta de dientes
- Destapo la pasta de dientes
- Tomo el cepillo de dientes
- Aplico la pasta al cepillo
- Tapo la pasta
- Abro la llave del lavamanos
- Remojo el cepillo con la pasta de dientes
- Cierro la llave del lavamanos
- Froto los dientes con el cepillo
- Abro la llave del lavamanos
- Enjuago la boca
- Enjuago el cepillo
- Cierro la llave del lavamanos
- Me seco la cara y las manos con una toalla

López (2009) indica que un algoritmo debe tener las siguientes tres características:

- Debe ser realizable: El algoritmo termina después de una cantidad finita de pasos. Es inaplicable cuando empieza con datos iniciales y el proceso secuencial resulta infinito o cuando se encuentra con algún obstáculo que hace que el algoritmo termine con un resultado.

- Debe ser comprensible: Sus instrucciones son claras, no puede ambigüedad en las mismas, de manera que quien ejecuta los pasos sabe exactamente qué, cómo y cuándo hacerlo.

- Debe ser preciso: El orden adecuado para la ejecución de los pasos permite que el resultado final sea siempre el mismo.

El desarrollo del pensamiento algorítmico es requerido en la programación de computadores. Sin embargo, también puede aplicarse en otras disciplinas, pues le permite al alumno una aproximación guiada y disciplinada a los problemas o una mejor comprensión de algún tema en particular. "En pocas palabras, la programación de computadores aporta al ámbito escolar un laboratorio para desarrollar habilidades." López (2009:22)

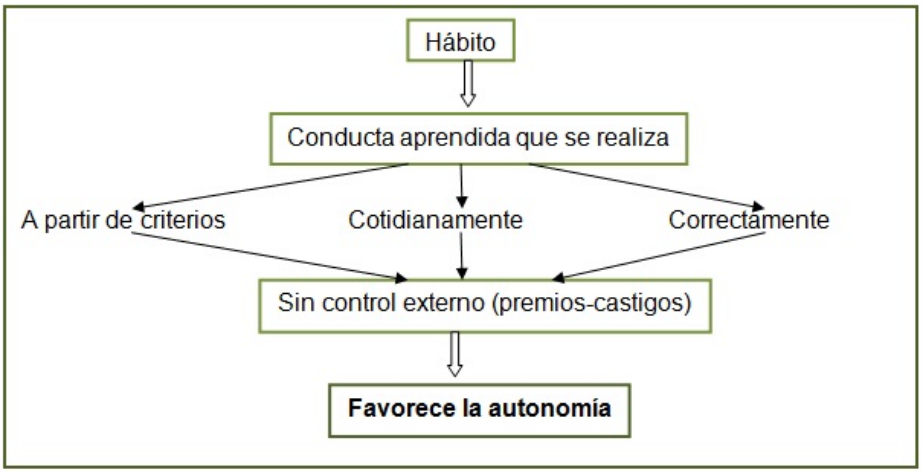
López (2009) indica que David Moursund (2006) propuso un planteamiento que relaciona la computación y el desarrollo del pensamiento algorítmico con los cuadros estadios del desarrollo cognitivo planteado por Piaget. Según Moursund (2006), durante la etapa de las operaciones concretas (7 a 11 años) el niño empieza la manipulación lógica y sistemática de los símbolos en un computador. Eso les permite utilizar lenguajes y manipular ambientes gráficos. Durante la etapa de las operaciones formales (12 – 18 años), el alumno utiliza de forma lógica los símbolos relacionados con conceptos abstractos.

Según López (2009) los estudiantes deben aprender a diferenciar entre comunicación informal y la precisión de la comunicación formal. Esto depende de elementos como el vocabulario utilizado, el uso de comodines en vez de términos o expresiones precisas, el uso de adverbios coloquiales en vez de adverbios formales. No puede haber ambigüedad en la forma en que se expresan las instrucciones.

4. Proceso de adquisición de hábitos de autonomía. Saber resolver un problema o el conocer el mecanismo para llegar a una solución no tiene mayor importancia si no se hace un hábito de este mecanismo. Es importante que los docentes comprendan que es necesario formar un hábito de resolución de problemas en sus alumnos, para que estos, por iniciativa propia, en su momento, resuelvan conflictos sin la ayuda de un mentor. Para esto, es necesario comenzar por

definir qué es un hábito. Un hábito es la ejecución de una conducta aprendida previamente de manera oportuna, adecuada y sin necesidad de control externo (Comellas, 2002).

Figura 1. Esquemas claves de la conducta habitual.

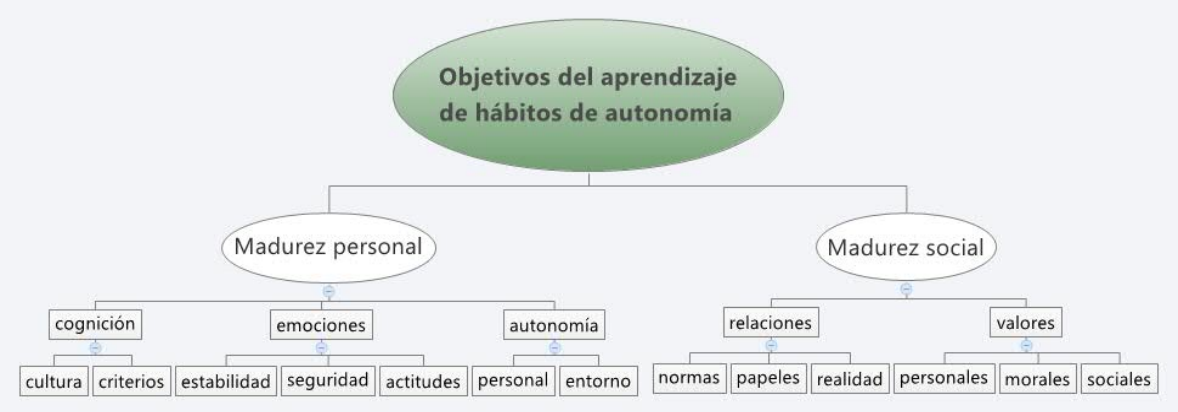


Fuente: (Comellas, 2002)

Comellas (2002) explica que los hábitos tienen sentido desde una doble perspectiva: como aprendizaje funcional que favorece la adquisición de autonomía del individuo, por lo que es un aprendizaje válido por sí mismo, y como factor fundamental de madurez personal y social. Siendo este último el objetivo educativo general en el que se basa el aprendizaje.

El aprendizaje incide en el proceso madurativo global del individuo: en el proceso de humanización (propio de la especie), en el proceso de singularización (identidad personal) y en el proceso de socialización (como miembro de la comunidad en la que se está implicado y en la que se ocupa un lugar) (Charlot, 1997 citado por Comellas 2002). Por lo tanto, partiendo de que el aprendizaje de hábitos de autonomía perdura a lo largo de toda la vida, se pueden lograr objetivos favorecedores en la calidad de vida personal y social.

Figura 2. Objetivos globales del aprendizaje de hábitos de autonomía.



Fuente: (Comellas, 2002)

El desarrollo global de las potencialidades depende de la madurez personal, que a su vez depende tanto de la edad como del entorno cultural. La cognición, emociones y autonomía son aspectos básicos que suponen dicha madurez personal, los cuales Comellas (2002) analiza de la siguiente manera:

- Cognición: para que una conducta se constituya en un hábito espontáneo es preciso que, durante la etapa de aprendizaje, se garantice no sólo la comprensión del motivo de dicha conducta sino, también, el contexto en que se debe dar y el proceso para su correcta ejecución. Es preciso que el niño interiorice la conducta como una situación con significado (aprendizaje significativo), a la vez que es importante que se le presente como medio para lograr una mejor relación con el entorno.

- Emociones: la madurez emocional incide en la globalidad de la conducta humana, por lo que estará presente en cualquier situación cotidiana. Para incidir en la educación de las emociones es necesario considerar los deseos y las necesidades individuales y grupales, ya que se constituyen en aspectos claves que orientan el comportamiento.

- Autonomía: determina el comportamiento en diferentes situaciones escolares, familiares y sociales a lo largo de la vida del individuo. Lo posibilita para la toma de decisiones y para la resolución activa de los requerimientos y exigencias con que se encuentre, basándose en el análisis de la realidad y los factores determinantes.

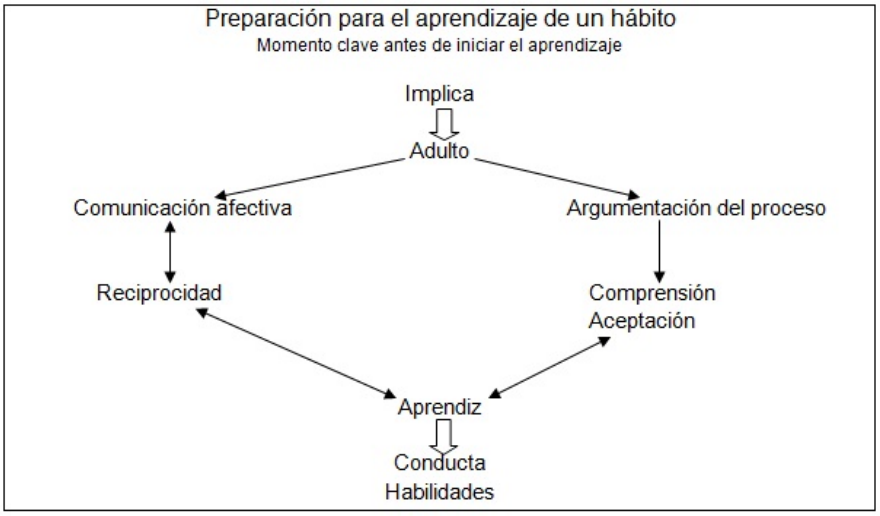
Figura 3. Componentes de la autonomía.

Autonomía	Personal	- Funcional - Ocio - Salud
	Entorno	- Estudio - Trabajo

Fuente: (Comellas, 2002)

Los componentes de la autonomía permiten al individuo realizarse en la vida. Una persona autónoma es metacognitiva, aprende de sus errores, conoce sus fortalezas y debilidades. Sabe escoger o desarrollar intereses que le satisfagan. Es responsable, tanto en sus labores como en los estudios, además de tener la capacidad de tomar medidas concretas y realistas encaminadas hacia su salud y bienestar personal.

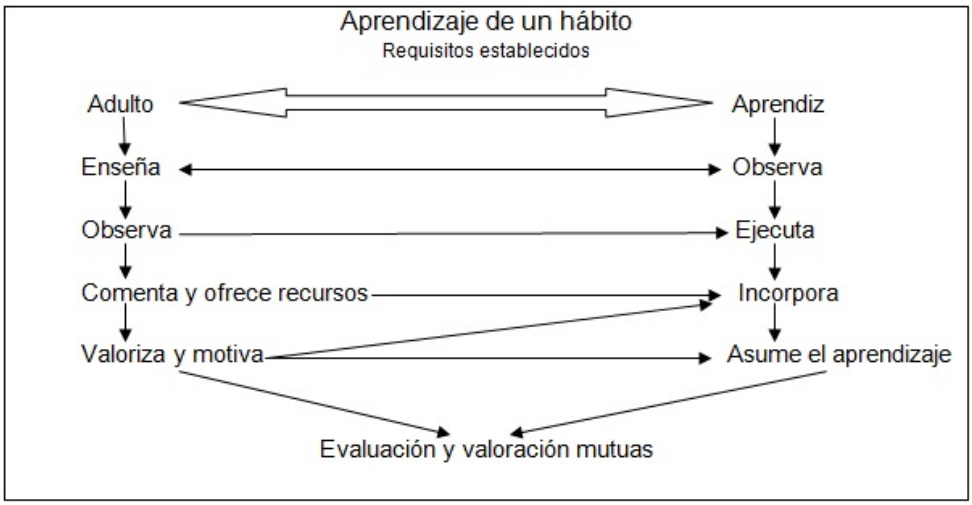
Figura 4. Fase preparatoria para el aprendizaje.



Fuente: (Comellas, 2002)

La actitud y el enfoque del docente al momento de iniciar la enseñanza de un hábito es esencial en el proceso enseñanza-aprendizaje. Este es el momento en el que se hace evidente la importancia de las características personales de los encargados de la enseñanza, los educadores. Si bien es cierto que hace falta reciprocidad en el procedimiento, ésta tiene grandes posibilidades de darse como resultado de la actitud y las acciones del maestro.

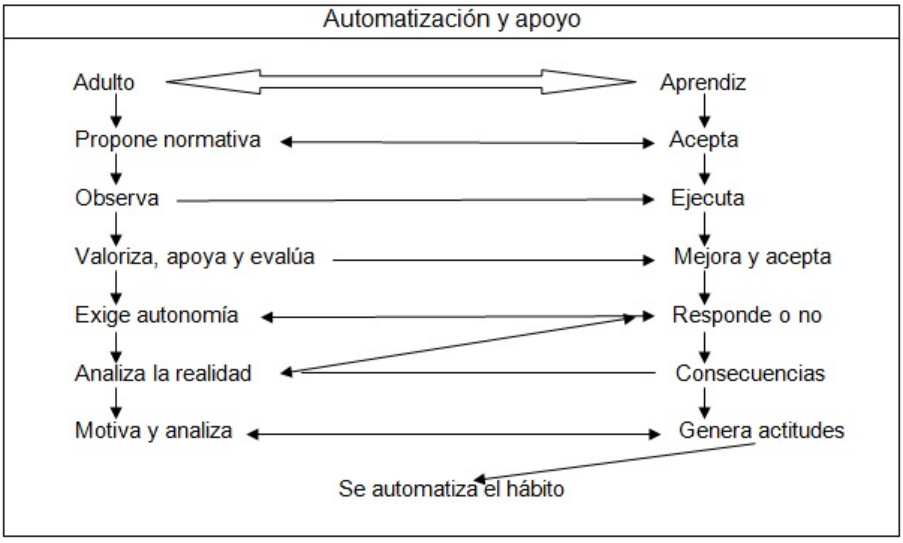
Figura 5. Proceso de apoyo al aprendizaje.



Fuente: (Comellas, 2002).

La función del docente es la de acompañar el proceso en el aprendizaje de un hábito. Debe estar atento para ofrecer los comentarios y recursos que el alumno necesita en el momento justo. Además es necesario que valore los esfuerzos y avances a lo largo del proceso. Toda esta atención y apoyo motivará al educando, favoreciendo la asimilación y el aprendizaje.

Figura 6. Interiorización del hábito



Fuente: (Comellas, 2002)

Al momento de la automatización de un hábito el rol del docente es la de observador. Debe estar atento a cómo el alumno se desenvuelve al momento de plantearle una situación para resolver, siempre apoyando, valorizando los esfuerzos del educando, además debe evaluar y exigir las actitudes o procesos que considere el estudiante ya es capaz o debe ser capaz de realizar. Es importante notar que la motivación es un eje central en todo el proceso. El alumno no debe sentirse acorralado o presionado, sino acompañado.

5. Formación docente. La formación docente de calidad es un factor indispensable en el sistema educativo. El Ministerio de Educación, a través de la Dirección de Calidad y Desarrollo Educativo (DICADE), diseñó un Currículum Nacional Base para la Formación Inicial de Docentes del nivel primario (CNB-FID), encontrando entre sus características principales: <<*ser flexible, lo que hará posible una gama de adaptaciones y concreciones, según el contexto de cada institución educativa; y perfectible, es decir, susceptible de ser mejorado y perfeccionado de acuerdo con las situaciones cambiantes de la sociedad y del mundo global. Estas características abren el espacio para una autonomía institucional y docente responsable, con capacidad de autoconducirse y libertad de saber elegir con sentido de asumir las responsabilidades implícitas*. >> (Dirección de Calidad y Desarrollo Educativo. (DICADE, 2006)

Según éste, el perfil del docente egresado de la carrera de magisterio ha desarrollado las siguientes competencias:

- *Organiza el aprendizaje de la niñez en atención a su proceso de desarrollo particular y del ser humano en general, con base en conocimientos pedagógicos y didácticos.*

- *Desarrolla estrategias para analizar e interpretar situaciones y proponer soluciones y alternativas viables, eficaces y efectivas de mejoramiento de los procesos de formación para la infancia y la niñez.*

- *Utiliza métodos para facilitar los aprendizajes, aplica fundamentos pedagógicos, brinda atención a las necesidades, intereses y expectativas de los alumnos y alumnas.*

- *Vincula los contenidos curriculares con la experiencia de vida y los intereses y el entorno inmediato de los alumnos y alumnas.*

- *Genera respuestas eficaces y válidas a los múltiples y variados contextos, situaciones y demandas que enfrenta en su quehacer profesional.*

- *Actúa con autonomía en la formación de los alumnos y alumnas realizando las adecuaciones administrativas, pedagógicas y curriculares pertinentes y necesarias con base en el conocimiento apropiado de la pedagogía y de las áreas académicas.*

- *Desarrolla proyectos educativos que contribuyan al mejoramiento de la calidad de vida de la comunidad donde está inserta la escuela.*

- *Aplica el pensamiento lógico y crítico para la resolución de problemas de la cotidianidad docente.*

a. Perfil del docente que enseña matemáticas. De manera general, el perfil de un docente es: <<el conjunto de competencias organizadas por unidades de competencias requeridas para realizar una actividad profesional, de acuerdo con criterios valorativos y parámetros de calidad>> (Rodríguez, 2010 cita a Galvis, Fernández y Valldivieso, 2006).

Al ser docente especializado en una materia es necesario tener dominio tanto teórico, histórico y filosófico, como didáctico y pedagógico de dicha materia. Las Matemáticas no son la excepción. El objetivo de la Matemática es la formación de un ser humano crítico, pensante y transformador de su realidad. Para lograr dicho objetivo el perfil del docente debe poseer una sólida base en Historia, Filosofía, Psicología y Sociología de las Matemáticas. Que, al ser presentadas todas, motiven al alumno al estudio de la misma a través de su aplicación y utilidad en problemas propios de su contexto.

Lo ideal sería un docente humanista, apasionado por el legado de la Matemática, que no use su saber Matemático como objeto de poder en el salón de clases, y que además considere que "esta ciencia puede ser aprendida por todos y no que sólo por unos pocos la tienen a su alcance; los que tradicionalmente se denominan *inteligentes"* (Rodríguez, 2010).

Rodríguez (2010) afirma que no se trata de definir un listado de competencias para el perfil del docente de Matemáticas, y, cita a Marrone (2008), quien afirma que al docente se le concibe <<*como un ser total en el que se integran simultáneamente las dimensiones cognitivas, afectivas, sociales, espirituales, un ser innovador y competente y principal recurso de su formación. Las acciones de formación deberán procurar el protagonismo del docente en todo el proceso, favoreciendo su participación de modo activo, potenciando una imagen del docente más cercana*

a su papel de agente de desarrollo curricular o animador del curriculum y brindando oportunidades tanto para el diálogo y la reflexión individual y colectiva como formas de aprendizaje y de crecimiento intelectual y profesional, como para la adecuación de las propuestas curriculares a su realidad escolar. Asimismo, deberán potenciar actitudes hacia la investigación que contribuyan a profundizar en la construcción y reconstrucción de su profesionalidad, y hacia el trabajo conjunto, con miras al establecimiento de una cultura docente más colaborativa>>.

Lo imperativo es que el docente posea la capacidad para vivir la cultura de la Matemática en el aula. Que éste sea capaz de transmitir, a través de la pasión que siente por dicha ciencia, el amor hacia la misma, haciéndola comprensible hasta para aquellos que no se sienten cómodos con ella.

b. Enseñanza de las matemáticas en Guatemala. Según el Ministerio de Educación, en el documento *Contenidos de Matemáticas para docentes de primaria que van a optar a plazas bilingües y monolingües*, *<<las Matemáticas ayudan a organizar el conjunto de conocimientos, modelos, métodos, algoritmos y símbolos necesarios para propiciar el desarrollo de la ciencia y la tecnología en las diferentes comunidades del país>>*. Sin embargo, en cuanto a la enseñanza de las mismas, en Guatemala, aún existen muchas deficiencias. Ésta se basa únicamente en la automatización de las operaciones básicas, sin que se llegue a la comprensión de las mismas a tal grado que se apliquen o utilicen como herramienta para resolución de problemas. <<En las evaluaciones del 2008, el 21% de los estudiantes de sexto primaria respondieron correctamente las preguntas de resolución de problemas>> (Quiñonez, y otros, 2010).

El Ministerio afirma que por medio de las Matemáticas se desarrollan habilidades, destrezas y hábitos mentales, tales como el cálculo, estimación, observación, representación, argumentación, investigación, comunicación, demostración y autoaprendizaje. Por lo que instituye los contenidos siguientes para lograr el desarrollo de dichas destrezas y habilidades Matemáticas:

Formas, patrones y relaciones. *Ayudan en la construcción de elementos geométricos y en la aplicación de sus propiedades en la resolución de problemas, desarrollan la capacidad de identificar patrones y relaciones, de observarlas y analizarlas no sólo en situaciones matemáticas sino en actividades cotidianas.*

Matemáticas, ciencia y tecnología. *Es el componente por medio del cual se aplican los conocimientos de la ciencia y la tecnología en la realización de acciones productivas, utiliza métodos alternativos de la ciencia para construir nuevos conocimientos, etc.*

Sistemas numéricos y operaciones. *En este componente se estudian las propiedades de los números y sus operaciones para facilitar la adquisición de conceptos y la exactitud en el cálculo mental. Estudia los fundamentos de las teorías axiomáticas para expresar las ideas por medio de signos, símbolos gráficos y términos matemáticos.*

La incertidumbre, la comunicación y la investigación*. Utiliza la estadística para la organización, análisis, representación gráfica y la probabilidad para hacer inferencias de hechos y datos de su cotidianidad. Utiliza, también, la construcción y comunicación de predicados matemáticos y el uso del razonamiento en la investigación, para resolver problemas y generar nuevos conocimientos.*

El contenido de la evaluación para docentes está basado en términos de destrezas y habilidades complejas a un nivel de sexto grado del Nivel Primario. Para resolverla el docente debe poner en práctica las capacidades para analizar, razonar y comunicar ideas de manera efectiva mediante el planteamiento, la formulación y la resolución de problemas matemáticos propios de este grado.

Sin embargo, los informes de estudios nacionales, tanto de maestros como de alumnos, revelan que ambos tienen un rendimiento deficiente.

Y, lógicamente, la mala preparación de los primeros da como consecuencia el bajo rendimiento de los segundos. Gálvez-Sobral (2010) afirma que los docentes guatemaltecos tienen una preparación deficiente. En las pruebas del Ministerio de Educación del año 2006 únicamente un 15% obtuvo un nivel favorable en Lectura, mientras que en Matemáticas esa cifra desciende a un 2%. Este último resultado (el más grave de los dos) se ve reflejado en el Informe para educadores de las Evaluaciones Nacionales del Nivel Primario de sexto grado de Quiñónez (2011), el mismo indica que en las pruebas generales de matemática, únicamente el 8% de los estudiantes obtuvo resultados “excelentes” y un 12%, “satisfactorio”, el porcentaje restante o “debe mejorar” o se encuentra en el nivel “insatisfactorio”. En resolución de problemas específicamente, sólo el 28% obtuvo respuestas correctas a problemas de operaciones combinadas, el 22% a problemas de proporcionalidad y 18.8% a problemas de porcentaje.

Los resultados de las pruebas realizadas en el 2006 por el Laboratorio Latinoamericano de Evaluación de la Calidad Educativa (LLECE), el Segundo Estudio Regional Comparativo y Explicativo (SERCE), que fue coordinado por la Oficina Regional de Educación de la UNESCO para América Latina, en el documento Aportes para la enseñanza de la Matemática (2009), confirman que Guatemala se encuentra lejos de alcanzar los niveles de óptimos de desempeño. De hecho, de los 17 países que participaron en dicha evaluación, sólo Nicaragua, Panamá y República Dominicana se encuentran por debajo de los resultados de Guatemala en la solución de problemas simples, y en la de complejos, esto se reduce a los últimos dos.

En el mismo informe se reporta que en el nivel superior de desempeño, <<se ubica aproximadamente, el 11% de los estudiantes tanto de tercer como de sexto grado de primaria. Es decir, sólo ese porcentaje de estudiantes de ambos grados puede responder correctamente la mayoría de las preguntas de mayor demanda cognitiva de las pruebas de Matemática. Ello acusa un significativo déficit de calidad de la educación en este campo que se está ofreciendo a los estudiantes de primaria en América Latina y el Caribe>>.

Cuadro 8. Porcentaje de estudiantes con respuestas correcta para matemática de sexto grado, por proceso cognitivo y por país.

PAÍS	Solución de problemas	
	Simples	Complejos
Argentina	43,12	35,37
Brasil	41,87	31,15
Colombia	37,89	29,89
Costa Rica	48,47	39,43
Cuba	65,17	53,5
Chile	40,72	32,42
Ecuador	33,22	23,8
El Salvador	35,66	28,01
Guatemala	32,88	23,58
México	49,18	40,85
Nicaragua	32,7	24,94
Panamá	32,49	23,27
Paraguay	36,59	25,12
Perú	40,51	31,13

Continuación Cuadro 8

País	Simples	Complejos
República Dominicana	25,88	20,73
Uruguay	52,11	49,23
Estado Nuevo León	49,72	40,15
Región	41,16	32,7

Fuente: (Laboratorio Latinoamericano de Evaluación de la Calidad Educativa (LLECE), 2009)

c. Procesos de formación en matemáticas para docentes en servicio. La Matemática es una ciencia en constante remodelación, por lo que, según Rodríguez (2010) el maestro de matemática debe ser <<un matemático-docente-investigador que enseñe lo que investiga y que haga de su práctica docente objeto de estudio, uno que enseñe lo que practica y transmita criterios y procedimientos para superar su propia práctica profesional>>.

Un docente que se forma o se capacita se mantiene actualizado y competente, sin embargo, <<el profesor de Matemática raramente reconoce su deficiencia didáctica, más bien, racionaliza el hecho achacando su fracaso a los estudiantes porque `son malos para la matemática´>> (Rodríguez, 2010 cita a Martínez 2006).

Guatemática es un Proyecto de Mejoramiento de la Enseñanza de la Matemática en Guatemala auspiciado por la Agencia de Cooperación Internacional del Japón (JICA), en apoyo al Ministerio de Educación para la formación de docentes en servicio. Como lo afirman los estatutos de la Sociedaad del Conocimiento de la Enseñanza de la Matemática con Guatematica, para ser miembro de dicha sociedad es requisito *tener interés en "aprender a enseñar" y "enseñar a aprender" matemática con Guatemática*.

Según el informe sobre *Cómo hicieron los sistemas educativos con mejor desempeño del mundo para alcanzar sus objetivos* realizado por McKinsey & Company (2007), <<*las experiencias de estos exitosos sistemas educativos resaltan la importancia de tres aspectos:*

- Conseguir a las personas más aptas para ejercer la docencia;
- Desarrollarlas hasta convertirlas en instructores eficientes; y
- Garantizar que el sistema sea capaz de brindar la mejor instrucción posible a todos los niños>>.

Es posible generalizar que en una empresa, para que cualquier labor se lleve a cabo eficientemente, es necesario contar con personas idóneas y capacitarlas hasta que tengan las herramientas que las conviertan en trabajadores eficientes, lo que a su vez, garantiza el óptimo funcionamiento de dicha empresa. En docencia estos aspectos cobran mayor importancia. Contar con las personas idóneas garantiza, hasta cierto punto, el que éstas vayan a estar prestas, o que incluso posean la motivación intrínseca, y por ende, la iniciativa de buscar las capacitaciones necesarias de acuerdo a sus deficiencias personales o a las necesidades de los alumnos a su cargo.

B. Computación

A continuación se presenta la sustentación teórica para el desarrollo de la herramienta computarizada, la cual fue desarrollada por miembros de la Facultad de Ingeniería, departamento de Computación. Incluye temas relevantes sobre construcción de intérpretes e interfaces gráficas.

1. Paradigmas de programación. Un paradigma de programación es un marco de referencia ampliamente adoptado que trata de definir la forma de programar estándar, para resolver varios problemas bien definidos. Se basa en una serie de tecnologías y conceptos desarrollados para la adecuada aplicación y difusión de la técnica. En el transcurso de la historia los paradigmas de programación han surgido como respuesta a los paradigmas predominantes en épocas anteriores, con el fin de resolver de forma elegante algunos de los mayores problemas que dichos paradigmas dejaron abiertos. (The George Washington University)

Hasta la fecha se han desarrollado muchos paradigmas de programación, pero de forma canónica se entiende que los más destacados son:

a. Declarativo. Propone que un programa es una serie de declaraciones de proposiciones y condiciones diseñadas para que oculten o abstraigan de la lógica de alto nivel del programa, el detalle de su implementación. Este paradigma engloba algunas corrientes importantes, descritas a continuación:

1) Programación lógica. Define a un programa como una serie de premisas que son necesarias para inferir algún resultado deseado (The George Washington University). Este tipo de programación es de especial utilidad en *Inteligencia Artificial* (para el desarrollo de sistemas expertos). Un lenguaje representativo de dicha corriente es Prolog.

2) Programación funcional. Se basa en el desarrollo de funciones aritméticas, cuya definición limita el tipo de datos a tratar; no es posible almacenar datos de estado, ni utilizar

datos que cambien en el tiempo (The George Washington University). Una característica importante de este paradigma es la forma de entender el término "función". Una función recibe una entrada, la procesa y produce una salida, las salidas pueden variar en tipo, desde un dato numérico hasta otras funciones. Cada función procesa dependiendo del contexto en el que fue llamada. Algunos lenguajes representativos de esta corriente con Haskell y Scheme.

b. Imperativo. Entiende a un programa como un estado delimitado por variables de control que son alteradas en el transcurso de su ejecución a través de sentencias que el programador especifica (The George Washington University). El programador detalla un algoritmo exhaustivo para que la computadora resuelva un problema determinado. Algunos lenguajes de programación representativos de éste paradigma son BASIC y C.

c. Orientado a objetos. Un objeto es una entidad con estado y comportamiento. El estado de un objeto es el conjunto de valores de las variables internas (de instancia) que el objeto posee en un momento determinado de la ejecución de un programa; y el comportamiento del objeto es el conjunto de acciones que puede realizar, estas acciones están definidas por un tipo especial de función llamada método. Este paradigma introduce una serie de conceptos que cambiaron la forma de entender las interacciones entre los datos que ocurren en un programa (The George Washington University). Algunos conceptos importantes de dicho paradigma se esbozan a continuación:

1) Clase. Especificación de tipo que define el dominio del estado de un objeto y su comportamiento.

2) Herencia. Relación entre dos clases en la que se afirma que A hereda de B, si y sólo si el estado y comportamiento de B es extendido hacia A.

3) Objeto. Ejemplificación de una clase, es la forma en que una clase se hace efectiva y puede manejar su estado y realizar tareas mediante su comportamiento. Los objetos son los bloques básicos de un programa orientado a objetos, la mayoría de los cálculos se realizan mediante una o más interacciones entre objetos.

4) Método. Función algorítmica perteneciente al comportamiento de un objeto, especificado por la clase del mismo.

5) Evento. Suceso que el ambiente detecta debido al cumplimiento de un conjunto de condiciones determinadas por el programador o por el mismo ambiente de desarrollo.

6) Mensaje. Conjunto de órdenes y datos enviados a un objeto para que realice una operación.

7) Interfaz. Conjunto de métodos provistos por un objeto, visibles y aplicables fuera de el, que permiten el acceso a su comportamiento.

8) Atributo y estado. Cada una de las variables internas a un objeto, en el ámbito de todos sus métodos, son conocidas como atributos del objeto. En conjunto, los valores de todos los atributos en un momento dado de la ejecución del programa determinan el estado del objeto. Por convención un objeto debe ocultar su estado (solo el objeto puede cambiarlo) y proveer de una interfaz para su manejo y modificación.

El paradigma de orientación a objetos se caracteriza por un conjunto de reglas y principios que acompañan a los anteriores conceptos, a continuación se esbozan los más importantes:

9) Abstracción. Un objeto esconde la implementación y provee una interfaz de alto nivel para ejecutar las tareas y comunicarse con este. Es decir un objeto debe funcionar como una caja negra.

10) Encapsulamiento. Un objeto contiene todos (de forma exhaustiva) los métodos y atributos que definen su comportamiento y estado respectivamente.

11) Modularidad. Un programa debe estar inherentemente dividido en partes más pequeñas, cada cual es independiente de las demás. Cada una de estas partes es conocida como módulo.

12) Principio de ocultación. Un objeto es un módulo por definición, es decir, sólo el mismo tiene acceso directo a su estado, y para interactuar con el estado del mismo es necesaria una interfaz.

13) Polimorfismo. El nombre (o referencia) de un objeto puede comportarse como distintos tipos de objetos, lo cual se determina en tiempo de ejecución del programa. Esta característica está definida en gran parte por la jerarquía de un objeto en su árbol hereditario.

2. Lenguajes formales. Conjunto de palabras. Un lenguaje formal se compone de varios elementos (Sakharov):

a. Alfabeto. Conjunto no vacío de todos los símbolos, caracteres, y dígitos que pueden conformar las palabras de un lenguaje. Colectivamente se conoce como letras a los elementos que componen un alfabeto.

b. Palabras. Secuencia finita de letras del alfabeto. Por definición todo alfabeto define una sola palabra de longitud cero llamada *épsilon*.

c. Reglas de formulación. Éstas definen la forma en que una serie de palabras es construida de manera que se considere que la serie pertenece al lenguaje; estas reglas pueden definirse de forma enumerativa. La forma más común de presentar dichas reglas es mediante una gramática formal.

En general entendemos una gramática como el conjunto de reglas que definen el discurso bien formado en un lenguaje determinado. Específicamente define la forma correcta de formular oraciones. Las gramáticas formales son un conjunto de producciones (reglas de construcción) que definen todas las posibles hileras de palabras que pertenecen a un lenguaje.

Las gramáticas formales de mayor interés en el desarrollo de lenguajes de programación son las gramáticas libres de contexto (Sakharov).

d. Gramáticas libres de contexto. Es un tipo de gramática formal que se compone por:

1) Conjunto de símbolos terminales. Conocidos como tokens. Son los símbolos básicos del lenguaje.

2) Conjunto de no terminales. Variables sintácticas que describen un conjunto de hileras de caracteres. Generalmente estas descripciones se dan en forma de expresiones regulares.

3) Conjunto de producciones. Una producción se compone por un no terminal conocido como cabeza y una secuencia de terminales y no terminales conocidos como cuerpo, que describen una forma de expresar la construcción de la cabeza.

4) Símbolo inicial. Símbolo al que se debe reducir una hilera de caracteres para ser considerada parte del lenguaje.

3. Traducción de lenguajes. El proceso de traducción toma de entrada una hilera de palabras en un lenguaje fuente y las convierte en su equivalente en un lenguaje destino (generalmente código máquina). Para entender dicho proceso es necesario explicar las fases necesarias para la

traducción y las herramientas de software utilizadas para realizarlo (Aho, Lam, Sethi, & Ullman, 2007).

a. Procesos. Dentro de la tarea de traducción se ejecutan varios procesos, presentados gráficamente en la Figura 7.

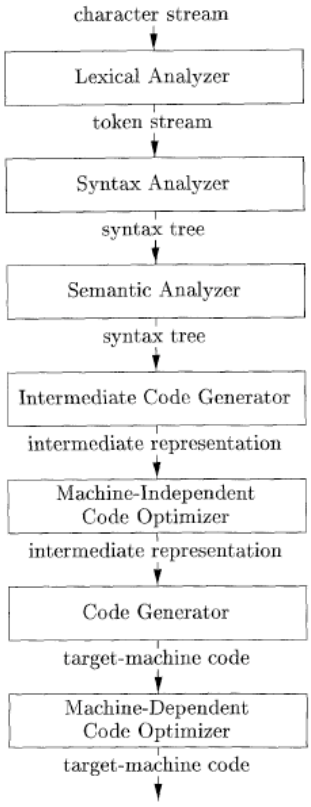
1) Análisis léxico. Conocido también como escaneo. Se abastece de la hilera de caracteres de entrada y produce una hilera de tokens en orden según la entrada. Cada token contiene la información correspondiente a su identificador y el valor leído.

2) Análisis sintáctico. Conocido también como parseo. Basado en la gramática del lenguaje fuente y mediante un proceso de derivación, trata de asignarle a la hilera de tokens entregada por el análisis léxico, una estructura. Dicha estructura es genéricamente conocida como árbol sintáctico.

3) Análisis semántico. Utilizando el árbol sintáctico producido en la fase anterior, esta fase trata de verificar las relaciones de significado entre los nodos de dicho árbol. Su objetivo es determinar el *sentido* programa, para así en las fases posteriores impregnar dicho sentido en el código de salida. En esta fase se verifica la concordancia de tipos, entre otros.

4) Síntesis de código. Las fases de generación de código (intermedio y máquina) y optimización de código se resumen como la síntesis de código. En estas fases se construye, con base en los resultados de las fases de análisis, el código de salida en el lenguaje destino.

Figura 7. Fases del proceso de traducción.



Fuente:(Aho, Lam, Sethi, & Ullman, 2007)

b. Herramientas. El proceso de traducción se lleva a cabo por varios tipos de herramientas de software, entre ellas mencionamos las más importantes:

1) Compilador. Un compilador es un programa computacional que se encarga de traducir una hilera de caracteres en un lenguaje determinado (fuente) hacia otro lenguaje de programación (generalmente a código de máquina) que la plataforma de desarrollo soporta y puede ejecutar (Aho, Lam, Sethi, & Ullman, 2007). El compilador lee la entrada completa y ejecuta todas las fases de traducción sobre dicha entrada.

2) Intérprete. Un intérprete es un programa computacional similar a un compilador, pero se diferencia principalmente en que éste último ejecuta todas las fases de traducción, sentencia por sentencia (Aho, Lam, Sethi, & Ullman, 2007).

La principal ventaja de un compilador respecto de un intérprete es que el proceso de traducción completa se ejecuta una sola vez, y la ejecución de un programa ya compilado es mucho más rápida que uno interpretado. Por otra parte, un lenguaje interpretado provee varias facilidades inherentes: el proceso de depuración es más sencillo y directo, y también facilita la implementación de consolas interactivas.

4. Gramática del español. Para facilitar el aprendizaje en los usuarios del tema de los algoritmos se propone que es necesario no introducir complicaciones más allá de las inherentes al problema, por ello se debe desarrollar un lenguaje lo más parecido al que los usuarios manejan (para el caso del megaproyecto ABJ será el español plano). Para definir una estructura condensada del español se toman en cuenta dos enfoques principales: el funcionalismo y el generativismo.

a. Funcionalismo. Tiene un enfoque pragmático y adecuado para aplicaciones fuera de la computación, debido a que cada constructo del lenguaje se enuncia y define según su contexto (Burgos). En términos de computación, este hecho hace bastante difícil utilizar el enfoque funcionalista para definir una gramática (Functional Grammar, 2009). Es por ello que los resultados más importantes de esta investigación son los referentes al generativismo.

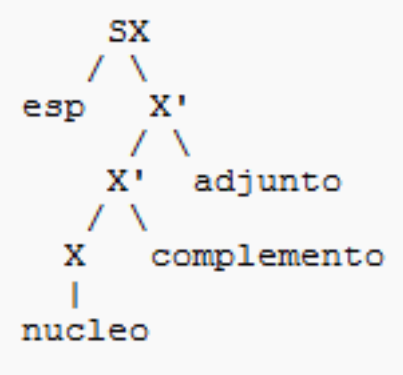
b. Generativismo. Propone un enfoque basado en la Teoría Estándar propuesta por Noam Chomsky en la década de 1960. El extenso trabajo de Noam Chomsky en la lingüística ha propuesto que la gramática es independiente del significado, y que por lo tanto la sintaxis está caracterizada por una gramática formal con reglas de transformación (Chomsky, Syntactic Structures, 2002). Chomsky creó la gramática generativa (Chomsky, Topics in the theory of generative grammar, 1966) (inicialmente llamada gramática transformacional), que separa los elementos del lenguaje y les da una estructura formal, sugiriendo la existencia de una "gramática universal" de la cuál todas las gramáticas heredan sus propiedades. Cada elemento es una regla o una unidad que en conjunto definen una oración genérica de la cual se pueden desprender una variedad infinita de otras oraciones (Ministerio de Educación de la Nación Argentina). Este enfoque se ajusta bastante al usado en cursos como Construcción de Compiladores, para la definición de gramáticas libres de contexto.

En un lenguaje de programación (o de forma más general, lenguaje formal) normalmente se trata con las llamadas gramáticas formales, las cuáles poseen una definición en función de conjuntos y conceptos que relacionan los elementos de dichos conjuntos (Norvell, 2002). Las gramáticas, y los lenguajes formales se diferencian de los lenguajes naturales en aspectos tales

como que los formales pueden tomarse como modelos de lenguajes naturales (Stokhof, 2007) o, fundamentalmente, en que un lenguaje formal cumple con ciertos criterios de diseño, por lo cual puede ser circunscrito y estudiado en su totalidad (Cornell University Department of Computer Science, 1994).

Por lo anterior, se decidió construir la gramática del proyecto bajo la idea del generativismo. Se debe aclarar que no se adoptará el enfoque integral generativista, debido a que nuestro lenguaje no pretende adaptar el idioma español completo; será suficiente que se defina la gramática tomando en cuenta cierto tipo de oraciones especiales. Al final la definición de la gramática se conceptualiza en árboles del tipo generativistas como el que se muestra en la siguiente figura:

Figura 8. Árbol generativista para una gramática simple en español. X y X' pueden ser cualesquiera elementos de la oración, por ejemplo Nombre, Verbo, Objeto Directo, etcétera.



5. Lenguajes formales y su aplicación en la educación. El megaproyecto está orientado a los niños guatemaltecos que hablan español y otros idiomas con orígenes mayas; es necesario que la forma de enseñar algoritmos no traiga consigo una abstracción o dificultad más allá de la implicada por el tema. Lenguajes Naturales son todos aquellos que surgen sin previa planificación en los humanos debido a una capacidad nata. Se caracterizan por poseer estructuras flpo los, en las que sucede el fenómeno de la ambigüedad (Association for the Advancement of Artificial Intelligence, 2011)(Center for Natural Language Processing).

El funcionamiento de las computadoras se reduce a sumas, restas y direccionamientos de memoria, las computadoras no fueron diseñadas para resolver dilemas ni para inferir más allá de lo programado. Por ello, una computadora necesita de instrucciones precisas para poder ejecutar programas (no existe cabida para la ambigüedad).

He ahí el problema de fondo que se tiene al tratar que una computadora interprete un lenguaje natural. A diferencia de las computadoras, los humanos utilizan más que el mensaje "verbal" o "escrito" para identificar la idea detrás de la instrucción. Los sentidos le permiten usar información extra para así descifrar el verdadero contenido y cometido de lo que se está solicitando (gestos, tono de voz, expresión facial, etcétera) (Association for the Advancement of Artificial Intelligence, 2011).

Por otro lado, un lenguaje formal es un conjunto de palabras (cadenas finitas de letras), símbolos y tokens. Las reglas que definen la estructura de un lenguaje formal están descritas por una gramática formal. Los lenguajes formales normalmente son dotados de semántica, la cual define un conjunto externo de reglas que le dan un sentido a la estructura definida por la gramática. La gramática y la semántica del lenguaje formal son definidas por estructuras llamadas expresiones regulares, producciones y reglas semánticas (Aho, Lam, Sethi, & Ullman, 2007).

Un tipo de gramática formal son las gramáticas libres del contexto. Se llaman "libres de contexto" debido a que las estructuras que pertenecen al lenguaje descrito por la gramática tienen un solo significado, no importa en qué momento ni en qué lugar se usa (no depende del contexto). Estamos interesados en este tipo de gramática debido a que la mayor parte de los lenguajes de programación pueden ser generados por una gramática libre de contexto (Sakharov).

Así pues, se usaría la teoría de los cursos de *Lenguajes formales y Compiladores* para construir un programa que interprete el lenguaje que usaremos para enseñar algoritmos a los usuarios.

El megaproyecto ABJ no es el primer proyecto en usar los lenguajes de programación como medio de enseñanza a los jóvenes. Ejemplos famosos lo son Pascal, Squeak y Logo. Aún así, el producto final que se les presenta a los estudiantes generalmente no hace tan evidente la formalidad bajo la cual se han construido; es precisamente esta característica la que se busca al usar un lenguaje de programación para educar (Junauza, 2009) (Heng, 2009).

- Las características más importantes de los lenguajes orientados a la educación son:
- Presentan un desafío de distinto nivel según la etapa de aprendizaje en la cual se usan.
- Sus estructuras son simples, fáciles de definir y estables.
- La recursión y ciclos infinitos están limitados.

- Implementan al menos un módulo de gráficas por computadora o animación.
- Algunos no necesitan que el usuario escriba para poder programar (Heng, 2009).

Lo que en realidad se necesita es implementar un lenguaje de programación similar a un lenguaje natural con las herramientas de lenguajes formales que se han adquirido durante la carrera. Además, se debe tomar en cuenta que los niños aprenden y ven el mundo de forma distinta a la que los adultos (quienes normalmente programan). Por ello surge la siguiente pregunta de enfoque: ¿Es necesario pensar en un cambio del paradigma de programación para luego así implementar un lenguaje que facilite el desarrollo de las habilidades de autonomía e iniciativa personal en los jóvenes?

La respuesta que se propone es SÍ.

6. Arquitectura Modelo Vista Controlador. Es un patrón de diseño que separa los componentes lógicos, de la interfaz de usuario, haciendo más fácil el proceso de desarrollo, pruebas unitarias y mantenimiento. Originalmente éste patrón de diseño se especificó en 1979, y se aplicó por primera vez en el lenguaje orientado a objetos Smalltalk (Wikipedia, 2011).

a. Componentes. La arquitectura MVC está compuesta por tres elementos (según indica su nombre), en la Figura 9 se propone una representación gráfica de la interacción entre las partes del modelo MVC:

1) Modelo. Segmento que trabaja en el dominio de los datos, almacena el estado del sistema de forma coherente y provee una interfaz de comunicación suficientemente estrecha y robusta para que tanto el controlador y la vista puedan interactuar con éste. Normalmente atiende consultas de la vista usadas para desplegar en pantalla el estado del sistema. Además recibe llamadas del controlador para actualizar su estado.

2) Vista. Despliega el modelo del sistema en la pantalla. Provee de una interfaz de usuario que será la vía mediante la cual los usuarios utilizarán el sistema. La relación Vista / Controlador / Modelo es tal que es posible definir una o más vistas cuando aún se posan sólo una vista o un controlador.

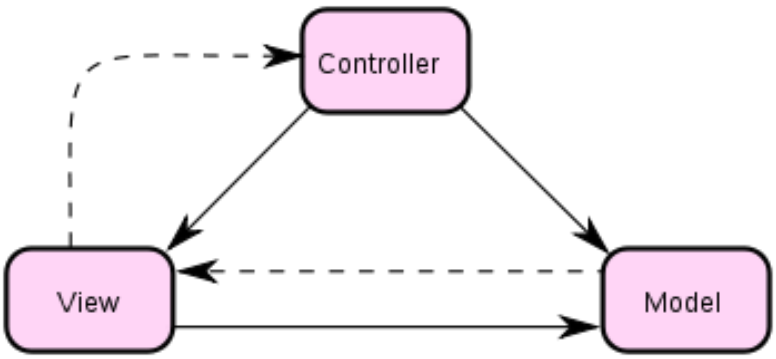
3) Controlador. Encargado de procesar las solicitudes del usuario (recibidas por medio de la vista), actualizar el estado del modelo y notificar a la vista sobre cualquier cambio hecho al modelo para que se actualicen los datos en pantalla.

7. Lenguaje extensible de etiquetas (XML). XML es un metalenguaje estándar. Se le conoce como metalenguaje debido a que su finalidad es la descripción de otros lenguajes a través del uso de etiquetas. XML y las tecnologías derivadas de el (por ejemplo XHTML y SVG) tienen aplicaciones para transferencia de datos, no sólo en el ámbito del internet (W3 Schools). A continuación se presenta un cuadro que resumen las ventajas y críticas que se hacen sobre XML:

Cuadro 9. Comparación entre ventajas y desventajas del uso de XML.

Ventajas	Desventajas
Se basa en una estructura anidada que es fácil de entender para un lector humano.	Alta complejidad espacial, es decir, estructuras de datos sencillas pueden llegar a extenderse mucho en disco (memoria) al ser representados en XML.
Debido a que es un estándar, los intérpretes de XML son lo suficientemente generales para no preocuparse por incompatibilidad entre ellos.	Existen alternativas de bajo peso y muy populares para substituir al XML (como Json)
La definición de etiquetas nuevas puede hacerse en el camino.	

Figura 9. Representación genérica de una arquitectura Modelo Vista Controlador.



Fuente: (Wikipedia, 2011).

8. Modelo de Objetos de Documento (DOM). Es una interfaz de desarrollo que define los objetos que existen en un documento basado en el estándar XML. Además, DOM define la forma en que un programador puede interactuar con el documento a través de consultas al mismo (escritura y lectura) (W3, 2005).

La estructura de un XML posee elementos y atributos, donde los atributos describen características de un elemento y deben ser únicos dentro de cada elemento, mientras que los elementos pueden estar anidados y presentarse múltiples veces en cada nivel de anidación. La información se almacena en el XML dentro de los elementos y atributos. En el DOM, tanto elementos como atributos son llamados "nodos", y la navegación puede realizarse utilizando las funcionalidades del lenguaje de programación; de forma 'manual', al leer y manejar el documento como un archivo de texto; o utilizando XPath. XPath (XML PathLanguage) es un lenguaje especial para referenciar partes de un documento XML, y es una recomendación del W3C (World Wide Web Consortium (W3C), 1999).

La navegabilidad, estandarización y bajo costo de utilizar estructuras XML las hace ideales para almacenamiento de datos clasificados, en forma similar a una tabla relacional (Bos, 1997).

9. PHP. Su nombre es abreviatura para PHP: HypertextPreprocessor, y es un lenguaje de scripting (lenguaje normalmente interpretado, para controlar una o más aplicaciones) orientado al desarrollo web. Tomando su sintaxis básica de los lenguajes C, Java y Perl, PHP es una herramienta ampliamente usada por permitir la codificación de páginas web generadas dinámicamente (PHP Group, 2011). La licencia de uso de PHP es libre (PHP Group, 2011), y la documentación y nuevas versiones se encuentran disponibles a través del sitio oficial del PHP Group, y en diferentes sitios educativos en línea tales como w3Schools.

PHP es el lenguaje en el que está escrita y en el que se desarrollan extensiones sobre la plataforma de software libre para manejo de contenidos llamada MediaWiki.

10. MediaWiki y los manejadores de contenido. Un manejador de contenidos (CMS, por sus siglas en inglés) es un sistema que permite la creación, manejo, distribución, publicación y consulta de información es un sitio. Cubre aspectos como la estructura del sitio y el ciclo de vida de la información contenida en él (Robertson, 2003). Por su parte, el concepto Wiki fue creado por Ward Cunningham originalmente para el descubrimiento del patrón de lenguajes en programación, pero desde siempre basado en el trabajo colaborativo (Cunningham, 2003).

MediaWiki es un CMS para wikis escrito en PHP y provisto como software libre por la WikimediaFoundation. Originalmente, MediaWiki fue desarrollado para la popular enciclopedia en línea Wikipedia (MediaWikiFoundation, 2007). MediaWiki es capaz de manejar una gran cantidad de usuarios a nivel diario, ofreciendo las capacidades de edición de contenido a través de su lenguaje wikitext (MediaWikiFoundation, 2011), y permitiendo un alto nivel de

personalización de la plataforma a través del desarrollo de extensiones en PHP, las cuales pueden ser desde un cambio en el aspecto visual hasta una aplicación web completa que corre sobre la plataforma. Aunque wikitext es una versión simplificada de lenguajes técnicos como HTML, muchos usuarios optan por editores "WhatYouSeeIsWhatYouGet" (WYSIWYG) para compensar la falta de intuitividad que wikitext presenta. Un editor WYSIWYG es aquél que permite la edición no del código fuente de un documento o sitio, sino de su presentación tal y como se ha de ver en su lanzamiento (Lennartz, 2008). Para ABJ se implementó un editor WYSIWYG en el sitio de apoyo para facilitar el manejo y actualización de contenido.

11. PyQT. Es un set de bindings de Python para el framework de aplicaciones QT de Nokia y corre en varios sistemas operativos, e.g., Windows y Linux. Está desarrollado en forma de módulos de Python. (Riverbank Computing Limited, 2010).

12. QtDesigner. Es una herramienta QT utilizada para diseñar y construir interfaces gráficas (en inglés, graphicaluser interfaces – GUIs) a partir de componentes Qt. Se pueden componer y personalizar widgets y/o diálogos en un editor WYSIWYG. Las funcionalidades se pueden cambiar dinámicamente en el código fuente. (Nokia Corporation, 2010)

13. Algoritmos y su aplicación en la educación. El hombre a través de su actividad cotidiana debe resolver una gran cantidad de problemas teóricos y prácticos diferentes. Al resolverlos, actúa con un sistema controlador, transformando información sobre ellos. Para que tenga éxito la solución de los problemas, el hombre debe asimilar métodos específicos para resolverlos pero para hacerlo debe aprender y dominar los métodos necesarios. (Landa, 1978)

Los estudiantes deben lidiar con problemas cuya solución se realiza mediante algoritmos y para los que también es conveniente se enseñen con premeditación. Existen muchísimos de estos problemas que se realizan en las instituciones educativas.

Por ejemplo, algunos problemas de matemáticas son casi imposibles o muy difíciles de resolver sin conocer los algoritmos adecuados. Su resolución consiste en aplicar un algoritmo específico a los datos del problema. Por mencionar algunos, están los problemas de la multiplicación, división, potencias, raíces, transformaciones del álgebra, funciones trigonométricas. Para todos estos problemas se resuelven mediante algoritmos específicos y son esencialmente los que se enseñan a los estudiantes. La circunstancia de que existan problemas solucionables mediante algoritmos no solamente en las matemáticas, sino también en muchas disciplinas, es de gran importancia para la enseñanza. Es mucho más fácil y sencillo resolver un

problema cuando la persona conoce el algoritmo que cuando no lo conoce. Además, es incomparablemente más rápido y confiable.

Descubrir los algoritmos de solución para los diferentes tipos de problemas de distintas materias, permite la posibilidad de enseñar más rápida y exitosamente a los estudiantes a resolver los mismos, además de facilitar el dominio de la materia en su totalidad. Esto es obviamente importante, porque como ocurre en la actualidad, tiende a dársele demasiado trabajo a los estudiantes en las instituciones educativas. Por esta razón, la enseñanza de algoritmos debería desempeñar una importante función en la enseñanza de cualquier materia. (Landa, 1978)

Se ha tratado de crear una teoría que es la de idear algoritmos con propósitos pedagógicos, al igual que los medios para enseñarlos. La duda que surge es si no hay peligro de reprimir sus potencialidades creativos con este tipo de enseñanza. La respuesta es no, no solamente se debe motivar la creatividad sino también la formación de diferentes hábitos que deben manifestarse de manera automática ya que son un componente necesario de cualquier proceso creativo.

Por otro lado, el aprendizaje de algoritmos no debiera reducirse a la memorización de los mismos ya que la elaboración y formulación de los mismos pueden constituir una manera eficaz de cultivar cualidades de pensamiento creativo. (Landa, 1978)

14. Usabilidad. El estándar ISO (International Organization for Standarization) define la usabilidad como "el grado en el que un producto puede ser utilizado por usuarios específicos para lograr metas en específico, con efectividad, eficiencia y satisfacción en un contexto de uso especificado"

La usabilidad del software, a pesar de ser un factor cualitativo, ha ganado importancia a través de los años, hasta el punto de estar al nivel de atributos como el performance y la robustez del software. En Interacción Humano-Computador (HCI, por sus siglas en inglés) la usabilidad estudia la elegancia y claridad con la que la interacción con un programa de computadora es diseñado. (Nielsen, useit.com: Jakob Nielsen's Website, 2011). La usabilidad está definida por cinco componentes de calidad (Nielsen, useit.com: Jakob Nielsen's Website, 2011):

- Facilidad de aprendizaje: Se define como la facilidad con la que un usuario realiza tareas básicas en su primer encuentro con el diseño.

- Eficiencia: Una vez que el usuario está familiarizado con el diseño, puede medirse la eficiencia identificando qué tan rápido puede realizar tareas.

- Memorabilidad: Es la facilidad de un usuario para restablecer su habilidad para manejar el programa cuando lo utiliza nuevamente, después de un período de tiempo prolongado.

- Errores: Es la cantidad y magnitud de los errores que un usuario comete al tratar de realizar una tarea. Éste componente también mide la facilidad con la que el usuario se recupera de los errores cometidos.

- Satisfacción: Determina qué tan agradable es el uso del diseño para el usuario.

15. Pruebas de usabilidad. Una prueba de usabilidad es una técnica utilizada para evaluar la usabilidad de un producto ensayándolo en usuarios. Ésta consiste básicamente en observar a usuarios reales utilizando el producto para descubrir errores y áreas de mejora. Si se realiza correctamente, esta técnica produce realimentación valiosa que no puede obtenerse por medio de otras técnicas. (Nielsen, Usability Engineering, 1994)

Las pruebas de usabilidad se caracterizan por cinco aspectos según Dumas y Redish: El objetivo principal es mejorar la usabilidad del producto, los participantes representan usuarios reales, los participantes realizan tareas reales, se observa y documenta lo que los participantes hacen y dicen durante la prueba y se analiza la información obtenida y se determinan problemas reales (Dumas &Redish, 1999).

a. Diseño iterativo. Según Jakob Nielsen la única forma de lograr experiencia del usuario de gran calidad es iniciar las pruebas desde el inicio del proceso de diseño y realizar sucesivas pruebas a lo largo del ciclo del software. (Nielsen, Parallel&IterativeDesign + CompetitiveTesting = High Usability, 2011)

El diseño iterativo es un acercamiento que consiste en conducir pruebas de usabilidad con las mismas tareas. Después de cada prueba el producto se mejora en base a los resultados obtenidos. (Dumas &Redish, 1999)

b. Prototipo en papel. El prototipado en papel es un método generalmente utilizado en el proceso de diseño orientado al usuario para probar la usabilidad de una interfaz gráfica. Éste consiste en elaborar prototipo de la interfaz utilizando dibujos hechos a mano, sin mucho detalle. La interacción que normalmente ocurriría entre el usuario y el programa se simula utilizando moviendo los pedazos de papel con los dibujos de los componentes de la interfaz.

Este método provee varias ventajas ya que ahorra tiempo y dinero permitiendo al desarrollador probar el producto del diseño de una interfaz antes de iniciar a codificar o iniciar el proceso de desarrollo. Es por esto que el método es especialmente útil en fases tempranas del ciclo de software, permitiendo identificar y corregir problemas de usabilidad desde el inicio.

Otro beneficio ligado a éste método recae en la comodidad que siente el usuario para criticar el mock up. Al ser un prototipo poco elaborado no intimida al usuario de la forma en que podría hacerlo una interfaz de apariencia nítida que involucre programación. Además esto permite que los resultados de la prueba no sean afectados por las habilidades de los usuarios para manejar una computadora. (Klee, 2000)

c. Pruebas de usabilidad en niños. Las pruebas de usabilidad aplicadas a niños son similares a las aplicadas a adultos. Sin embargo existen algunas diferencias principales que deben tomarse en cuenta al diseñar pruebas para niños. (Webcredible, 2006)

1) Stress causado por personas y ambiente nuevo. Los niños son más propensos a sentirse incómodos y estresados ante nuevos lugares y personas.

2) Tendencia a pedir ayuda. Los niños suelen pedir y recibir ayuda de los adultos a menudo, por lo que deben darse instrucciones claras al inicio. El moderador debe establecer que se desea que el niño realice las tareas por sí solo. Además deben considerarse estrategias para evadir las preguntas, tales como contestar con otras preguntas o pedirle al niño que intente de nuevo.

3) Tendencia a cansarse, aburrirse o darse por vencidos más rápidamente. El número de tareas debe poder ejecutarse en un período de duración menor a una hora. Si el niño se desespera o se irrita durante la prueba, los resultados se verán sesgados.

4) Importancia del lenguaje no verbal. No puede confiarse en las habilidades de comunicación de un niño. Es necesario prestar más atención a señales físicas que el niño demuestre al realizar distintas tareas.

5) Diferencias físicas. Debe tomarse en cuenta la altura de la silla y la posición del micrófono para asegurar la comodidad del participante.

6) Nivel de comprensión y alfabetismo. Es necesario redactar las instrucciones en lenguaje que un niño pueda entender. El moderador debe asegurarse de que el niño ha comprendido lo que se requiere de él.

d. Personalización de un ambiente. Existe una escuela de pensamiento que dicta que todo programa debe ser totalmente personalizable y existe otra escuela de pensamiento que dicta que la personalización indica falta de dedicación por parte del programador. Un punto intermedio es deseable en este aspecto ya que una estructura demasiado rígida puede producir frustración, mientras que un ambiente que otorga demasiada libertad para personalizar puede producir un déficit en la productividad. (Nanni, 2004)

16. Estudio de benchmark. Un estudio de benchmark es el proceso de comparar los procesos y métricas de desempeño de una empresa propia con las mejores prácticas de otras empresas. En HCI el estudio de benchmark se utiliza para establecer parámetros de medición de usabilidad. Esto se logra al capturar el nivel de usabilidad y características de productos similares, para luego tratar de imitarlos o superarlos en el software propio. (Berkun, 2003)

Para el estudio de benchmark fueron consideradas los siguientes programas:

a. Alice. Es un ambiente de programación en tres dimensiones que permite la creación una animación para contar una historia, jugando un juego interactivo, o un video para compartir en internet. Alice es una herramienta gratuita diseñada para exponer a los estudiantes a un primer acercamiento a la programación orientada a objetos (POO por sus siglas en inglés). En la interfaz interactiva de Alice los estudiantes pueden arrastrar y soltar componentes gráficos para crear un programa en el cual las instrucciones corresponden a oraciones estándar de lenguajes de programación orientada a objetos como Java, C++ o C#. Alice permite al estudiante observar cómo sus programas corren como una animación, permitiéndoles entender la relación entre las oraciones programadas y el comportamiento de los objetos en la animación (Alice Carnegie Mellon)

b. BlueJ. Es un ambiente desarrollado como parte de un proyecto de investigación para enseñar programación orientada a objetos. El proyecto ha sido desarrollado por La TrobeUniversity y University of Kent con el apoyo de Sun Microsystems.

El objetivo de BlueJ es proveer un ambiente de enseñanza fácil de usar que facilita enseñar el lenguaje de programación Java a estudiantes de primer año. Se ha hecho especial énfasis en la visualización e interacción para crear un ambiente que promueve la experimentación y exploración. (Sun Microsystems, 2011)

c. Everydaymath. Es un programa que hace énfasis en la aplicación de matemáticas en situaciones reales. Los números, habilidades y conceptos matemáticos no se presentan aislados,

sino que están ligados a situaciones y contextos que son relevantes para el diario vivir. El curriculum también provee numerosas sugerencias para incorporar las matemáticas en las rutinas de clase diarias y con otras áreas de estudio. (The University of Chicago School Mathematics Project, 2011)

d. Etoys. Etoys es una herramienta de software educativa para enseñar ideas poderosas a los niños de manera convincente. Utiliza un entorno de autor multimedia y un sistema de programación visual. (Squeakland, 2011)

e. Scratch. Scratch es un lenguaje de programación orientado a niños que les facilita crear sus propias historias interactivas, animaciones, juegos, música y arte; además le permite compartir con otros sus creaciones en la web. <<A medida que los jóvenes crean y comparten proyectos realizados en Scratch, aprenden importantes ideas matemáticas y computacionales; al mismo tiempo que aprenden a pensar creativamente, a razonar sistemáticamente y a trabajar colaborativamente>> (Scratch MIT)

17. Teoría del color. El color, posiblemente, es el aspecto más demandante de cualquier interfaz gráfica, es por esto que debe tomarse en cuenta la teoría del color. El color puede ser fuente de frustración o placer para los diseñadores y los usuarios, por igual. El uso efectivo del color puede favorecer la comunicación y estética de una aplicación, y además contribuir a la satisfacción de usuario. El color debe integrarse bien en los elementos visuales como íconos, imágenes y texto. (Cooper & Reimann, 2003) Éste puede ser utilizado para varios fines, como llamar la atención, mejorar la velocidad de navegación y mostrar relaciones entre elementos. Existen reglas para la efectiva selección de una paleta de colores según Nanni, (Nanni, 2004). Exagerar las diferencias de brillo entre colores en el fondo y colores en imágenes principales. El contraste entre colores se utiliza como método para enfatizar elementos deseados.

Utilizar colores obscuros con un tono tomado de la mitad inferior del círculo de tonalidades contra colores claros tomados de la mitad superior del mismo. Diferencias de contraste entre tonalidades en los hemisferios de la rueda de color (Nanni, 2004).

Evitar contrastar tonos de segmentos adyacentes en el círculo de tonalidades, especialmente si los colores no contrastan.

18. Elementos de la interfaz gráfica

a. Widget. Es un elemento de la interfaz gráfica que despliega una cantidad de información modificable por el usuario. Entre estos se puede mencionar cuadros de texto, cuadros de opciones, barras de navegación, vista de árboles, lista de cuadros de texto, etiquetas, barras de progreso, barras de información, barras de estado, ventanas modales, cuadros de diálogo, y contenedores de widgets.

Los widgets utilizados en la interfaz gráfica fueron:

1) Vista de árboles. Es una vista jerárquica de la información, en dónde cada ítem también llamado nodo tiene sub ítems y así sucesivamente, creando árboles. Se utilizó para el almacenamiento de las acciones llevadas a cabo para resolver los problemas.

2) Contenedores de widgets. Son un tipo de caja que tiene una colección de widgets; estos se utilizan para tener un mejor control sobre el ambiente de desarrolló. Se utilizó los contenedores de widgets como escenario de la aplicación, dónde se llevan a cabo las acciones sobre los objetos, dichos objetos son etiquetas.

3) Barras de navegación. Toda una imagen continua puede ser corrida dentro de una ventana si es que su tamaño excede al tamaño de la ventana o de la pantalla, para esto se manejan las barras de navegación. Se utilizaron barras de navegación para mostrar todo el escenario ya que este podría llegar a ser más grande que la pantalla que los usuarios tuvieran. Las barras de navegación utilizadas fueron horizontales y verticales, estas estaban englobadas dentro de los contenedores de widgets.

4) Menús. Un menú es una lista de comandos o acciones representado por texto y/o símbolos en forma de opciones. En proyecto se utilizaron para representar las acciones que el héroe puede realizar sobre el escenario y los objetos. También se utilizaron menús para la interacción entre objetos.

5) Submenús. Son menús ordenados jerárquicamente permitiendo la navegación dentro de estos de una manera estructurada.

6) Botones. Es un área definida por una forma, texto y/o imagen la cual genera una acción cuando es invocada o presionada. Ésta puede estar representada por distintas formas y por lo general tienen una imagen o texto que representa la acción de la misma, esto es con el fin

de que sea más intuitivo para el usuario. En el proyecto se usaron botones para las acciones del héroe, objetos, mochila y escenario, también para la verificación del estado actual en el problema.

7) Mallas. Son un estilo de cómo serán distribuidos, de manera gráfica, los elementos dentro de una ventana, contendor o widget.

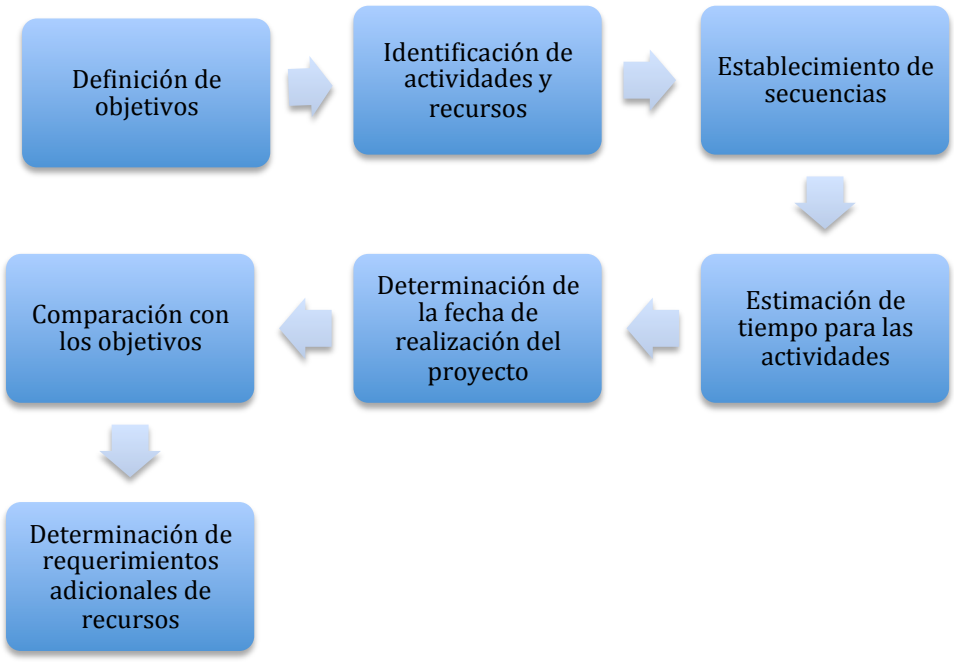
C. Industrial

1. Proyecto. Un proyecto se define como el conjunto de actividades que se llevan a cabo una sola vez, se relacionan entre sí y que tienen un tiempo de inicio y fin definido (Coulter, 2010).

2. Planeación. La planeación es el proceso mediante el cual se definen los objetivos del proyecto, se establecen las estrategias para lograr dichos objetivos y se desarrollan los planes para integrar y coordinar las actividades de trabajo (Coulter, 2010).

3. Administración de proyectos. La administración de proyectos consiste en realizar las actividades de un proyecto en tiempo, con el presupuesto disponible y de acuerdo a las especificaciones establecidas (Coulter, 2010).

Figura 10. El proceso de planeación de proyectos.



4. Estructuración de proyectos. De las tres diferentes estructuras de proyectos que existen, dentro de nuestro megaproyecto se utilizó “El proyecto funcional”, que es una división funcional entre dos o más grupos (Coulter, 2010).

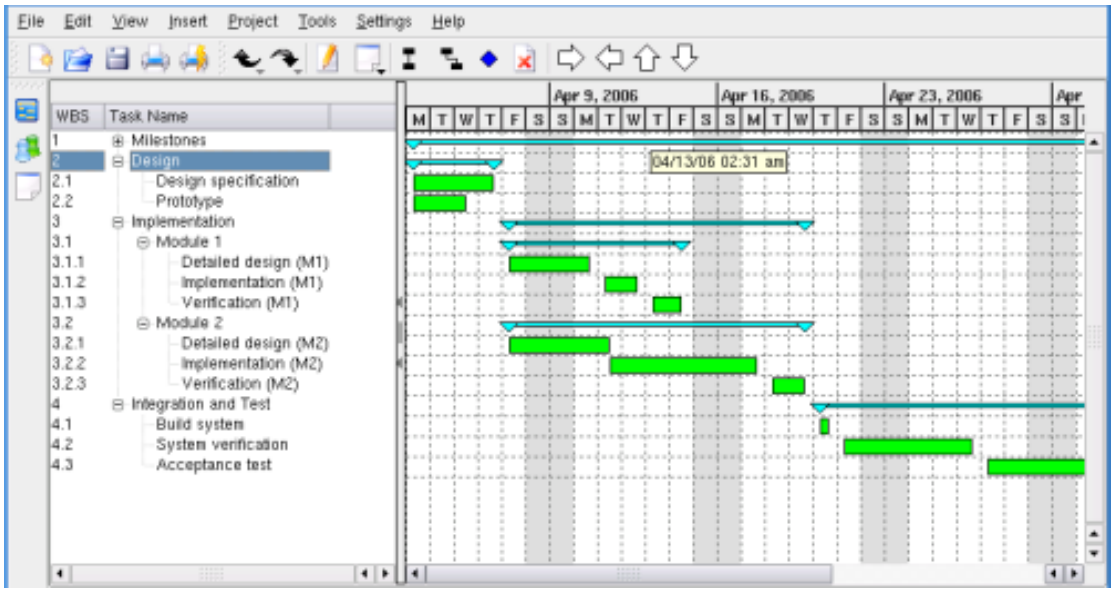
5. División de trabajo en equipos. Adam Smith sostuvo que la productividad, considerada como la capacidad de producir una cierta cantidad de bienes con un conjunto de recursos dados, será mayor si el trabajo se divide entre especialistas que cumplan funciones definidas.

En nuestro megaproyecto se utilizó la división de trabajo con el objetivo de aumentar la productividad de cada una de las personas que integran el equipo y que el trabajo realizado por alguien no fuera repetido por otros (Laza, 2006).

6. Calendarización. La calendarización es el detalle de las actividades que se deben realizar dentro de un proyecto, incluye el orden en que deben realizarse, quien será el encargado de ejecutarlas y cuándo deben estar terminadas. La calendarización se lleva a cabo a partir de los objetivos y metas del proyecto (Coulter, 2010).

7. Gráficas de Gantt. Las gráficas de Gantt son una representación visual de la calendarización. Muestra los resultados tanto reales como planeados dentro de un periodo dado. En las gráficas se puede observar la cantidad de tiempo por actividad, la secuencia de las mismas, a qué persona le corresponde, el porcentaje que se lleva cumplido de dicha actividad, los recursos requeridos(Taha, 2004).

Figura 11. Gráfica de Gantt



8. Planeación de red en proyectos. Los proyectos se pueden modelar en redes y nodos conectados por ramas. Para resolver y determinar la ruta crítica de los proyectos se utilizan algoritmos de optimización de redes. Existen cinco algoritmos para este uso:

- Algoritmo de árbol de expansión mínima
- Algoritmo de la ruta más corta
- Algoritmo de flujo máximo
- Algoritmo del flujo capacitado con costo mínimo
- Algoritmos de la ruta crítica

Las redes son series de nodos enlazados con arcos. Con cada red se asocia algún tipo de flujo. En general, el flujo en una red está limitado por la capacidad de sus arcos, que pueden ser finitos o infinitos (Taha, 2004).

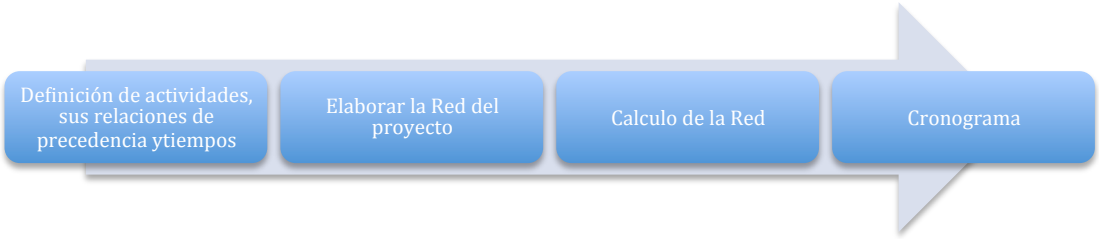
Se dice que un arco es dirigido u orientado si permite un flujo positivo en una dirección y flujo cero en la dirección opuesta. Una red dirigida tiene todos sus arcos dirigidos (Taha, 2004).

Una ruta es una sucesión de arcos distintos que unen dos nodos pasando por otros nodos, independientemente de la dirección de flujo en cada arco. Una ruta forma un ciclo si conecta un nodo consigo mismo, pasan por otros nodos (Taha, 2004).

Una red conectada es aquella en que cada dos nodos distintos están enlazados al menos por una ruta. Un árbol es una red conectada que puede consiste sólo en un subconjunto de todos lo nodos en ella, donde no se permiten ciclos, y un árbol de expansión es un árbol que enlaza todos los nodos de la red, también sin permitir ciclos (Taha, 2004).

9. Método Pert y CPM. Los métodos CPM (Método de la Ruta Critica o del Camino Critico, CriticalPathMethod) y PERT (*programevaluation and reviewtechnique, técnica de evaluación y revisión de programa*) se basan en redes, y tienen por objeto auxiliar en la planeación, programación y control de proyectos.

Figura 12. Fases de planeación de un proyecto con CPM o PERT.



Las dos técnicas difieren en que el CPM supone duraciones determinísticas de actividad, mientras que el PERT supone duraciones probabilísticas (Taha, 2004).

Durante la ejecución del proyecto, podría no cumplirse el programa que estaba planeado, causando que algunas de las actividades se adelanten o se atrasen. En este caso será necesario actualizar el programa para que refleje la realidad (Taha, 2004).

10. Ruta crítica. Es la secuencia más larga o la que toma más tiempo de actividades y eventos en una red. Cualquier retraso en la terminación de eventos de la ruta crítica retrasaría la terminación del proyecto. En otras palabras, las actividades en la ruta crítica tienen un tiempo de holgura igual a cero (Coulter, 2010).

11. Presupuesto. Un presupuesto es un plan de operaciones, integrado y coordinado, que comprende todas las fases de las actividades del proyecto, que en resumen, muestra los resultados financieros de llevar a cabo dicho plan (Gillespie, 1966).

12. Matriz de marco lógico.

a. Marco lógico. El Marco Lógico es una herramienta de análisis estructurado, que facilita el proceso de identificación, diseño, ejecución y evaluación de proyectos, pudiendo aplicarse en cualquier fase de los respectivos procesos de planificación. Se puede modificar y mejorar repetidas veces, tanto durante la preparación como durante la ejecución del proyecto o programa, incluso durante el funcionamiento del mismo (Ministerio de Educación de Chile, 2010).

b. Matriz de marco lógico. Es una plantilla que contiene el resumen del proyecto o programa. Se compone, en general, de cuatro filas (horizontales) y cuatro columnas (verticales). La matriz está basada en dos principios básicos. Primero, las relaciones lógicas verticales de

causa- efecto entre las diferentes partes de un problema, que corresponden a los cuatro niveles o filas de la matriz que relacionan a: las actividades, los componentes, el propósito y el fin, como el conjunto de objetivos jerarquizados del proyecto. Segundo, el principio de la correspondencia (lógica horizontal), que vincula cada nivel de objetivos a la medición del logro (indicadores y medios de verificación) y a las condiciones que pueden afectar su ejecución y posterior desempeño (o supuestos principales) (Ministerio de Educación de Chile, 2010).

13. Análisis de varianzas. El análisis de la varianza es un procedimiento estadístico que nos permite dividir la variabilidad observada en componentes independientes que pueden atribuirse a diferentes causas de interés.

En el planteamiento más simple de análisis de la varianza tenemos una variable numérica cuantitativa, que es el resultado, y queremos determinar en qué medida se puede atribuir la variabilidad de ésta a otra variable cualitativa nominal que vamos a denominar factor. Se está hablando por tanto de análisis de la varianza para un solo factor, que puede tener dos o más categorías o niveles (Runger, 2007).

14. Población. Estadísticamente hablando consiste en la totalidad de las observaciones que son de motivo de interés. Para el proyecto sería la población estudiantil 5to grado de primaria (Montgomery, 2007).

15. Muestra. Es un subconjunto de observaciones que se seleccionan de una población total que sea significativa para la validez del proyecto a realizar. La muestra sería en este entorno la cantidad de alumnos que se van a estudiar para el proyecto (Montgomery, 2007).

16. Costos. Estimación de gastos para la entidad gubernamental para la construcción, operación y mantenimiento del proyecto, menos el valor de salvamento (Blank, 2006).

17. Beneficios. En este tipo de proyectos son las ventajas que experimentará el propietario, pero en este caso es el público en general y los estudiantes del 5to grado de primaria (Blank, 2006).

18. Contrabeneficios. Son desventajas para el público cuando el proyecto se lleva acabo bajo diferentes consideraciones. Se pueden consistir en desventajas económicas indirectas de la alternativa (Blank, 2006).

19. Modelos de transporte. Se utilizarán para poder tener una guía de cómo llegar a cada región de Guatemala, para minimizar los costos del plan piloto (Taha, 2004).

20. Análisis beneficio / costo. Esta razón es la que se considera como la mejor forma para evaluar los proyectos del sector público. Se creó para darle una mayor objetividad a la economía en el sector público.

El análisis B/C es una decisión fácil de decidir. Si la razón es mayor a 1 se considera que el proyecto es económicamente aceptable para los estimados y la tasa de descuento aplicable, por el contrario si el proyecto es menor a 1 el proyecto no es económicamente aceptable (Blank, 2006).

21. Logística de distribución. Es modelo de optimización de la investigación de operaciones que trata como un solo sistema al flujo de materiales a utilizar. Empezando por el pronóstico de ventas, compra y procesamiento de materiales y de su inventario hasta que los productos en este caso los cuadernillos son enviados a los diferentes puntos de Guatemala (Taha, 2004).

IV. MARCO METODOLÓGICO

El marco metodológico está dividido por módulos e incluye la justificación y definición del problema observado.

A. Justificación y definición del problema o situación.

«Sólo son educadas las personas que han aprendido cómo aprender, que han aprendido a adaptarse y cambiar, que advirtieron que ningún conocimiento es firme; sólo el proceso de buscar el conocimiento da una base para la seguridad.»(Carl Rogers, 1977)

El mundo digital y globalizado actual exige que las personas posean un conjunto de habilidades y destrezas específicas que asegure su desarrollo y éxito personal. Estas habilidades han sido planteadas como fundamentales para la educación en el Siglo XXI por la organización The Partnership for 21st Century Skills, entidad fundada por el Departamento de Educación de Estados Unidos conjuntamente con organizaciones como Apple, Microsoft, Cisco, Dell, y la Asociación Nacional de Educación (National Education Association), entre otras. El conjunto de habilidades incluye niveles de pensamiento superior como la creatividad y la destreza para solucionar problemas.

La UNESCO, en su Segundo Estudio Regional Comparativo y Explicativo (SERCE) realizado en 2006, reporta que Guatemala se encuentra con un rendimiento por debajo de la media de la región en todas las áreas. Indica que únicamente el 31.32% de estudiantes de tercero primaria y 32.88% de sexto primaria respondió correctamente a problemas sencillos en la prueba de matemática. Esta prueba midió no sólo los conocimientos de los alumnos, sino también su capacidad para aplicar dichos conocimientos para comprender e interpretar el mundo, en distintas situaciones y contextos cotidianos.

De la misma forma, el artículo de Amanda Quiñónez Castillo publicado en el Boletín No. 9 Ajetab'al 9-2011 de la Dirección General de Evaluación e Investigación Educativa (DIGEDUCA) del Ministerio de Educación de Guatemala señala que los resultados de las pruebas estandarizadas aplicadas a primero, tercero y sexto grados de Primaria en el año 2008 muestran que cerca de un 50% de los alumnos evaluados no dominan las habilidades, capacidades y conocimientos requeridos para el grado que cursan, revelando también que el predominio de aprendizajes es de tipo memorístico. (Quiñónez, 2011).

Los resultados de las evaluaciones citadas evidencian la urgencia de buscar una estrategia que apoye el desarrollo del pensamiento lógico y estructurado en los estudiantes.

En la actualidad, se insiste en la necesidad de convertir el aula en un espacio más dinámico donde se enfatice el desarrollo de habilidades, capacidades y competencias, dejando atrás el concepto de que la educación es únicamente la transmisión de conocimientos. Esto último implica la aplicación de estrategias activas que conviertan al alumno en el protagonista de su aprendizaje.

El módulo temático "Programación de Computadores en Educación Escolar" elaborado con el apoyo de Motorola Foundation, Motorola de Colombia Ltda. y la gestión de la ONG Giveto Colombia en el año 2009 (Eduteka, 2009) recomienda como una buena alternativa para fomentar el desarrollo de la creatividad y la habilidad para resolver problemas la inclusión de un curso de algoritmos y programación de computadores, el cual debe tener como objetivo fundamental el desarrollo de habilidades, y no la formación de programadores. El módulo hace énfasis en la importancia de empezar a desarrollar estas habilidades desde temprana edad, <<dada la urgencia actual para que los estudiantes desarrollen habilidades del Siglo XXI…>> (s.f., 2009)

Atorresi, Macedo, Leymonié & Bronzina (2009) establecen que el estudio de Algoritmos y Programación mejora la competencia de autonomía e iniciativa personal y desarrolla el pensamiento lógico. De la misma forma, el módulo temático menciona que el aprendizaje de algoritmos y programación <<compromete a los estudiantes en la consideración de varios aspectos importantes para la solución de problemas: decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo y, monitorear sus propios pensamientos (meta cognición) y estrategias de solución.>> (s.f., 2009). Esto último fue validado también por el grupo de estudiantes del departamento de Ciencias de la Computación de la Facultad de Ingeniería que integran el Megaproyecto mencionado, quienes observaron que el curso Algoritmos y Programación desarrolló en ellos un pensamiento lógico y estructurado.

Las teorías constructivistas sociales afirman que el niño aprende mejor cuando el es expuesto a experiencias de aprendizaje que guardan relación con su entorno y contexto propio, logrando las estructuras que le permiten alcanzar un aprendizaje significativo. Los conocimientos se forman a partir de los esquemas del alumno como producto de su realidad y de la interacción con otros individuos que conforman su entorno. Integrando esto último con lo expuesto al principio del presente capítulo, se establece urgencia e importancia de crear una herramienta en donde el niño aprenda algoritmos utilizando elementos que encuentra dentro de su propio contexto.

Para apoyar en las aulas guatemaltecas el desarrollo de esquemas de pensamiento que permitan el desarrollo de la lógica y la resolución de problemas se propuso la ejecución del Megaproyecto *Propuesta de implementación de enseñanza de algoritmos y programación sin computadora en el segundo ciclo de primaria del sistema educativo guatemalteco.*

B. Objetivos del Megaproyecto.

- Crear y fomentar la competencia de autonomía e iniciativa personal en los estudiantes del segundo ciclo de primaria del sistema educativo guatemalteco, a través de la inclusión del tema Algoritmos y Programación en el plan de estudios normal.
- Mejorar la comprensión de problemas en los estudiantes, apoyándose en la idea de división en módulos.
- Desarrollar las habilidades necesarias en los estudiantes, para resolver problemas de forma ordenada y completa.
- Hacer de la enseñanza de Algoritmos y Programación un proceso simple, entretenido y accesible inclusive para estudiantes guatemaltecos con pobres condiciones computacionales.
- Lograr que la enseñanza del tema Algoritmos y Programación se adapte al contexto cultural en el que se impartirá (por medio del uso de problemas comunes para el nicho cultural al que pertenezca el estudiante).

C. Antecedentes.

La idea original se basa en la investigación *Competencias Básicas Para la* Vida, auspiciada por USAID y Juarez & Associates. La investigación tuvo como objetivo definir ciertos grupos de habilidades que todo guatemalteco debería poseer al graduarse de diversificado para enfrentar de la mejor manera un ambiente laboral o universitario. Entre las competencias definidas se encuentra Actuar de manera Autónoma e Identidad e Iniciativa personal. El objetivo de la competencia anterior se puede ver reflejado en las siguientes citas de la mencionada investigación:

[...] actuar de forma independiente, generando automotivación, perseverancia, tenacidad, constancia y seguridad sobre los derechos propios. Esto es acompañado de la capacidad de

auto-cuestionarse y auto-criticarse, siendo consciente de que también se tienen obligaciones (Estándares de Guatemala)

[...] deberá mostrar competencias asociadas a la autonomía personal, pues necesitará disciplina y autoestima para completar sus tareas en lugar de descansar (Estándares de Guatemala)

Una forma inteligente de fomentar dicha competencia es enseñar a los jóvenes a razonar de forma organizada y metódica para interpretar de mejor manera los problemas que se puedan presentar en la universidad, el trabajo y la vida en general. La comprensión y la división en módulos de un problema ayudarán a los jóvenes a requerir menos asistencia de superiores y compañeros para realizar una tarea (incidiendo esto de forma directa en la competencia mencionada anteriormente). Por la experiencia en el área de computación de los miembros del megaproyecto, se asegura que la comprensión y la habilidad para dividir un problema en módulos se puede adquirir al aprender a programar, y específicamente al aprender a desarrollar algoritmos, de ahí proviene la premisa sostenida por el megaproyecto Modelo ABJ. Una ventaja importante de este enfoque es que fomenta un proceso de reflexión y metacognición sobre las soluciones desarrolladas (aspecto importante desarrollado en la investigación mencionada con anterioridad).

Entre los proyectos que sin duda inspiraron el desarrollo del megaproyecto podemos mencionar Scratch y Alice, detallados en el marco metodológico.

D. Metodología.

La Universidad del Valle inició en el año 2004 la ejecución de megaproyectos, como una modalidad de trabajo de graduación, que consisten en equipos multidisciplinarios que ponen en marcha proyectos con el objetivo de plantear una solución a una problemática identificada. El presente megaproyecto, conocido como Modelo ABJ, incluyó dos departamentos de la Facultad de Ingeniería (Sistemas e Industrial) y la Facultad de Educación. Inició en el segundo semestre del 2010, y concluyó en el segundo semestre del 2011. Fue dividido en dos módulos principales: la herramienta computarizada (Sistemas) y la no computarizada (Educación). Los miembros del departamento de Ingeniería Industrial tuvieron a su cargo la administración y coordinación del proyecto entre los dos módulos. El modelo ABJ, pues, representa la intersección entre dos ejes principales, el bloque educativo y el computacional.

1. Educación. Para validar la herramienta desarrollada, se eligieron cuatro instituciones públicas y una privada dentro del perímetro de la ciudad de Guatemala. Dos instituciones públicas están localizadas en la Zona 14, una en la Zona 10 y una en la Zona 6. La población escolar de las primeras tres instituciones es similar: alumnos de ambos géneros y de nivel socio económico bajo. La escuela de la Zona 6 de la ciudad atiende una población escolar de sólo varones de nivel socio económico bajo. La institución privada también está localizada dentro del perímetro urbano de la ciudad de Guatemala. Es una organización educativa bilingüe (español-inglés), laica. Su población escolar es mixta de un nivel socio económico medio.

La muestra consistió en 259 alumnos de quinto grado primaria, la mitad de ellos fue el grupo experimental y la otra mitad el grupo control. Aproximadamente 209 fueron alumnos de instituciones públicas y 50 de una escuela privada.

La herramienta se aplicó por un período aproximado de 12 semanas en las que, luego de una corta capacitación, se le dio seguimiento y acompañamiento al educador. Para las visitas semanales efectuadas a las instituciones educativas durante el tiempo de aplicación de la herramienta, se elaboró un cuestionario para facilitar la evaluación cualitativa del proyecto, el cual se adjunta como anexo al presente informe.

Para la medición de los efectos de la misma se utilizó el diseño pretest-postest con grupo control (sólo una de ambas secciones recibirá tratamiento); se escogieron instituciones que contaran con dos secciones de dicho grado. La prueba estandariza que se utilizó fue el Test Otis Lennon de Habilidad Mental, Nivel Elemental II.

Figura 13. Diseño experimental utilizado para las pruebas.

R	O_1	X	O_2
R	O_3		O_4

Decisión que se tomó considerando la opinión de Buendía, Colás y Hernández (1998) en cuanto a que <<…este diseño es de los más completos que se pueden utilizar en investigación experimental. Incluye la medida de la variable dependiente en ambos grupos en situación del pretest (O1 y O2), aplicación del tratamiento al grupo experimental (X) y la medida, de nuevo, de la variable dependiente en ambos grupos en situación de postest (O3 y O4). La medida de la variable dependiente en situación de pretest y postes se hace al mismo tiempo en los dos grupos.

La asignación de los sujetos a los grupos se hace al azar y es lo que viene indicado por la R al comienzo del diagrama>>.

Los autores declaran que <<puesto que este diseño incorpora un grupo de control que tiene las mismas experiencias del grupo experimental, excepto el tratamiento, los factores de validez interna como son la historia, maduración, la selección, la mortalidad y la regresión estadística no son fuentes de invalidez interna. Es decir, éste es uno de los diseños que tienen mayor control sobre los factores que pueden afectar a la validez interna>>.

Además, debido a que los mismos autores consideran que "lo peor controlado en este diseño es la interacción que puede llegar a producir entre el pretest y el tratamiento; es decir, la aplicación del pretest puede producir un efecto de sensibilización en los sujetos que afectará a los resultados del postest", y por no contar con el tiempo suficiente para que exista el mínimo requerido (seis meses entre pre y postest) se aprovechó que la herramienta de medición a utilizar, el Test Otis Lennon de Habilidad Mental, Nivel Elemental II, tiene dos versiones (J Y K). Se pretendió minimizar dicho posible efecto de sensibilización utilizando la forma J en el pretest y la forma K en el postest, partiendo del supuesto de que ambas evalúan exactamente lo mismo, y se dejó, entre ambas pruebas, el máximo de tiempo posible (12 semanas), que nos permitiera obtener resultados dentro del margen de tiempo estipulado para la entrega de este informe.

2. Computación.

a. Investigación inicial. La investigación inicial tiene como objetivo adquirir un panorama general del problema al que se está enfrentando con el desarrollo del megaproyecto ABJ. Durante esta etapa se realiza un pequeño estudio de *punto de referencia*, una investigación sobre las técnicas para enseñar con la computadora, usabilidad de software para niños, lenguajes formales y su aplicación en la educación, plataformas posibles para desarrollo, niveles educativos guatemaltecos y la teoría del desarrollo intelectual de Piaget. Es en esta etapa que se diseña y elabora la primera wiki, con el objetivo de agilizar la comunicación entre los miembros del grupo.

b. Investigación temática. En esta fase se adentra en temas más específicos del desarrollo de la solución como: la gramática del español, herramientas útiles para el desarrollo del software, primer acercamiento al lenguaje propuesto y su tabla de símbolos. Esta fase tiene como objetivo sentar las bases para el diseño del proyecto.

c. Diseño. Con los resultados de las dos etapas anteriores se lograron definir las variables que afectarían al desarrollo del megaproyecto, como lo son: el lenguaje de programación a usar,

la plataforma para la cual se desarrollaría, prototipos de interfaz gráfica, etcétera. Además, tomando en cuenta las limitantes y beneficios de las herramientas seleccionadas, se desarrolló una idea completa del rumbo que se debería tomar para implementar exitosamente la herramienta de software. Para acceder al diseño completo de la herramienta, consultar los anexos / el capítulo de Diseño del Software.

d. Construcción y evaluación. Durante el diseño se utilizaron distintos elementos de las metodologías ágiles para desarrollo de software, como lo es el desarrollo basado en revisiones y ciclos de entregables cortos. Por eso se incluye en este punto tanto la construcción como la evaluación. En esta etapa se desarrolló y probó la herramienta de software, tanto la parte lógica (intérprete) como la parte gráfica (IDE). Además se llevaron a cabo una serie de pruebas de usabilidad con niños del colegio Suger Montano, y la creación de la documentación del proyecto.

e. Medición. Una de las partes más importantes del megaproyecto es la medición, puesto que es ésta quién dará la pauta si la herramienta desarrollada de verdad es efectiva. El resultado de esta fase (y de la siguiente) dará un fuerte sustento y base para pensar en un futuro presentar al Ministerio de Educación guatemalteco un plan completo de inclusión del tema de *Algoritmos y Programación* en el sistema educativo nacional. La medición se hizo usando la prueba estandarizada Otis Lennon, aplicándose con el método de preevaluación y postevaluación con un grupo experimental y uno control.

f. Análisis. Después de realizadas las pruebas y al tener los resultados de las mismas, se procedió a evaluar si efectivamente la herramienta y el sistema produjo un beneficio a los estudiantes en la competencia *Actuar de manera Autónoma, Identidad e Iniciativa personal*. Para ello se aprovechará la experiencia de las profesionales en educación que están trabajando en el megaproyecto ABJ.

g. Desarrollo de recomendaciones y trabajo escrito. Para concluir el megaproyecto se redactó un reporte con las experiencias obtenidas al desarrollar la etapa de construcción de la herramienta y medición de la misma. Con ello se pretende poner al alcance público las experiencias ganadas y ahorrar trabajo a futuros esfuerzos que vayan en líneas similares a la propuesta por el megaproyecto ABJ. Además en este reporte se incluyó una sección de recomendaciones sobre acciones que probablemente el equipo del megaproyecto pueda pensar que habrían sido de beneficio para alcanzar los objetivos del mismo, pero por limitaciones de tiempo o presupuesto no se pudieron implementar. Además se diseñó y publicó una nueva wiki, incluyendo toda la información que podría ser útil para futuros miembros de megaproyecto y para

el público en general, ya que se incluyen las instrucciones para descargar, instalar y utilizar la herramienta de software.

3. Industrial.

a. Planificación.

- Establecer los objetivos del proyecto, tanto el general como los específicos.
- Establecer el título formal y comercial del megaproyecto.

b. Planificación de actividades.

- Definición de todas las actividades que se realizarán durante el megaproyecto.
- Definición de los tiempos para cada una de las actividades.
- Planificación de actividades para cada unas de las fases.
- Realizar un Diagrama de Gantt por fase del megaproyecto.

c. Presupuesto.

- Investigar sobre los costos que se incurrirán en el megaproyecto.
- Realizar el presupuesto total de gastos del megaproyecto y encargarse de la ejecución del mismo.

d. Administración y logística.

- Administración de las actividades por medio de QDPM Project on-line.
- Establecer las instituciones educativas en donde se realizarán las pruebas.
- Administración del recurso humano para las actividades, tales como: Realización de Pre-test, vistitas, capacitaciones.
- Revisión sobre el cumplimiento de las actividades asignadas a cada grupo o persona individual, según el cronograma.

e. Análisis de datos.

- Tabulación de datos
- Análisis de varianzas para cada institución.
- Análisis de varianzas para el grupo experimental y control, en general (sin importar la institución educativa).
- Analizar los datos obtenidos del análisis de varianza y verificar si el uso de la herramienta tuvo algún impacto al finalizar el plan piloto.

f. Matriz de marco lógico.

- Realizar el análisis de involucrados, el análisis de problemas, el análisis de objetivos y el análisis de alternativas.
- Realizar la matriz de marco lógico, unificando el análisis de involucrados, el análisis de problemas, el análisis de objetivos y el análisis de alternativas.

g. Logística de distribución. En esta etapa del trabajo se planteará el manejo de logística en un sistema de distribución propuesto para Guatemala, y se manejarán todos los productos terminados necesarios, los cuales serán distribuidos a todo el país.

Todo este sistema tomara en cuenta todas las limitaciones para aplicar a dicho trabajo y diferentes modelos de administración, investigación de operaciones y lógica, como consideraciones con respecto a los problemas burocráticos para transporte de mercadería.

Se trabajará para enviar todos los productos necesarios, para llevar a cabo de la mejor manera el plan piloto y tener una idea base para que con está investigación se pueda desarrollar diferentes formas de distribución de los cuadernillos de trabajo. En esta parte el producto más importante son los documentos mencionados con antelación.

Para las localizaciones nos tenemos que dar un panorama de dónde estará ubicada cada una de nuestras escuelas, tomando en cuenta la población de cada una de las escuelas y la distribución óptima de todo el material de apoyo que se tiene para los alumnos como para los maestros.

h. Estructura de la distribución. Se presentará la estructura de red de distribución propuesta, se tomarán en cuenta varios factores importantes.

En todo el país se buscará distribuir de una manera eficiente todos los materiales necesarios para que el plan del proyecto se cumpla de la mejor manera, teniendo en cuenta todo tipo de retrasos, rutas críticas, tiempos de holgura y todo tipo de datos importantes para el mismo.

Se usaran tres tipos de escuelas para el almacenamiento de material:

- Escuela Central,
- Escuela Regional,
- Escuelas de Sector.

Con estos tres tipos de lugares de almacenamiento se irá hilvanando una red que logrará según los objetivos, toda la distribución de los materiales en toda la República de Guatemala.

Las funciones de las escuelas serán las siguientes:

1) Escuela central. Esta escuela es el punto de partida de toda la red, de ésta depende que todo se lleve de la mejor manera, además de la misma saldrán todos los materiales hacia las demás escuelas, todo el inventario de materiales se centralizara en ella. También funcionará como Escuela regional, ya que abastecerá a las bodegas del sector de la región central.

No es aconsejable centralizar todo en un solo lugar, se debe analizar el sitio que más convenga para que todo llegue en el tiempo correcto, con la cantidad correcta y en el lugar correcto. En este caso la ciudad será el lugar donde saldrá todo, ya que es donde se tiene la mayor información en cuanto a centralización de transportistas, huelgas y todo tipo de atrasos para una entrega.

Todo lo anterior hace recomendable ubicar la Escuela Central en la ciudad capital, ya que es el lugar con mayor desarrollo y donde los problemas no desaparecen pero se minimizan.

2) Escuela regional. Esta escuela tiene una función tan importante como lo tiene la central. Aunque es considerada como un intermediario, será a quien se dirige directamente la distribución por parte de la Escuela Central y será la encargada de abastecer a una región. Adelante se tomarán varios criterios dónde y porqué es necesario almacenar los materiales en la misma. Será más de una escuela, ya que sería difícil, por el tamaño del país, abastecerse solo de una Escuela regional.

3) Escuela de sector. Representa la cola de la distribución, en estas se busca abastecer lugares donde es conveniente tener un punto para suministrarlos, pues la distancia de la Escuela regional y el movimiento de los materiales justifican la creación de las mismas.

A continuación se presenta un organigrama de cómo se pretende que sea el sistema de abastecimiento de todos nuestros materiales a las diferentes escuelas de la república.

Figura 14. Sistema de abastecimiento

1) Ubicación de sistemas de almacenamiento. Se deben tomar en cuenta todo tipo de factores como accesibilidad, transporte, servicios básicos, lluvia y destrucción de los materiales para que se pueda almacenar de la mejor manera. Las escuelas deben contar con instalaciones adecuadas para que no se tenga el riesgo de destrucción de los materiales de todas las escuelas.

Se deben tomar en cuenta los factores anteriores para los tres tipos de escuelas que se utilizarán. Considerando todos los puntos que se plantearon, se tomaron en cuenta varias ciudades del país para la ubicación de las Escuelas, que se mencionan a continuación:

a) Escuela central: Ésta fue escogida en la ciudad capital porque es más su nivel desarrollo, disponibilidad de recursos y varios puntos de accesibilidad, además dan las características de infraestructura necesarias para poder almacenar todos los materiales, siendo esto el centro logístico de la operación de distribución.

Esta escuela también funcionará como una Escuela regional, ya que de aquí se manejará la operación para abastecer la Región Central del país. Tanto ésta como las demás Escuelas

regionales servirán como almacenaje para cada sector, pues también se abastecerá todos los alrededores de la zona.

b) Escuelas regionales: Para decidir sobre cuáles serían la Escuelas Regionales, se debe primero dividir el país en regiones, para lo cual se consideraron las siguientes cinco:

- **Región Central:** esta región comprende los departamentos de Guatemala, Sacatepéquez, Chimaltenango, Escuintla, Jutiapa, Jalapa, Santa Rosa; la parte oeste de El Progreso y El Petén, que por la difícil accesibilidad a este departamento se recomendaría vía aérea, pero se estará enviando por cuestión de costos vía terrestre. La Escuela regional es la misma que la Escuela Central que se ubico en la ciudad capital. Al igual los departamentos de Jutiapa, Jalapa y Santa Rosa serán atendidos desde esta región, pues se tomaron en cuenta las condiciones y la infraestructura de carreteras que se tienen en Guatemala, y se hace más sencillo transportarlas desde la capital que de otra Escuela regional.

- **Región Sur-Occidente:** esta región comprende los departamentos de Suchitepéquez, Retalhuleu y el sur de San Marcos. La Escuela Regional se decidió ubicar en la cabecera departamental de Retalhuleu, Retalhuleu, pues es un punto céntrico para esta región. Será desde aquí que se abastecerá los diferentes departamentos mencionados con anterioridad. Estas ciudades oscilan entre 40 y 100 kilómetros, por lo cual la Escuela regional está ubicada en el lugar correcto. Si el lector observa un mapa de Guatemala, notará que la región Sur-Occidente y Nor-Occidente casi se traslapan una con la otra, por lo que podría pensarse que no es necesario dividir esta zona en dos regiones, sino que sería posible trabajarla únicamente como el occidente del país. En efecto, no sería necesario dividirlas si sólo se enfocaran desde el punto de vista de distancias, ya que las ciudades de estas regiones se encuentran cercanas, pero si se analiza, se concluirá que por la cantidad de escuelas es mejor tenerlas divididas.

- **Región Nor-Occidente:** la creación de esta región, que comprende los departamentos de Sololá, Totonicapán, Quetzaltenango, Huehuetenango, la parte sur del Quiché y la parte norte de San Marcos, se explicó anteriormente. La escuela regional se decidió ubicar en la cabecera departamental de Quetzaltenango, Quetzaltenango, ya que esta ciudad también ha alcanzado mucha gente en el área

urbana y desarrollo económico, se considera clave para la distribución de todos los materiales hacía dicha región.

- **Región Norte:** esta región abarca Alta Verapaz, Baja Verapaz y la parte norte del Quiché, puede ser que se piense en abastecer algunas escuelas de Petén vía terrestre. La Escuela Regional se ubicó en la ciudad de Cobán, cabecera departamental de Alta Verapaz, el movimiento de escuelas en esta ciudad es grande, y el desarrollo de las mismas, está en constante crecimiento. De estás abastecerán toda la región.

- **Región Nor-Oriente:** esta región comprende los departamentos de Izabal, Zacapa, Chiquimula y la parte este de El Progreso. La Escuela Regional se ubica en Zacapa, ya que esta ciudad se ha desarrollado por toda su infraestructura y el crecimiento estudiantil, teniendo todas las cualidades para abastecer a las partes aledañas a la región que tiene a su cargo. Dicha zona fue necesaria dividirla por los crecimientos en infraestructura que están teniendo, en cuanto a carreteras para llegar a las ciudades que la forman, abasteciendo ciudades desde la capital, como se vio anteriormente.

- De esta manera queda el país dividido en cinco regiones, cada una con la responsabilidad de abastecer a las Escuelas de Sector. A continuación se propondrán lugares de ubicación para estás.

- **Bodegas de sector:** debe recordarse primero que cada Escuela Regional funciona como una Escuela de sector, desde el momento que ella misma sirve para abastecer el sector donde se encuentra.

- Las Escuelas de sector ayudarán a las Escuelas regionales a abastecer su zona. Para este tipo de Escuelas se sugieren los siguientes lugares estableciendo que cada Escuela de sector se encuentra en las cabeceras departamentales, ya que es aquí donde se encuentran la mayoría de escuelas del área urbana, que son las que estudiaremos durante el proyecto. Es importante mencionar que algunas Escuelas regionales, también funcionarán como Escuelas de sector, sin perder sus obligaciones de Escuela Regional. Existen muchos departamentos que tiene múltiples municipios, por lo que la Escuela de Sector facilitará el abastecimiento a todas estas escuelas.

c) Rutas de Guatemala. Se debe tener en cuenta que las rutas de toda Guatemala, tienen dificultades para poder acceder a todos los departamentos, debido a algunos eventos como lo son huelgas, paros de maestros, accidentes vehiculares y eventos de masacres de personas.

Al tener en cuenta estos eventos ajenos a la compañía de transporte, más la posibilidad de un desperfecto mecánico, influye en las llegadas a tiempo de los materiales a utilizar durante el proyecto. Alguna de las cosas que se pretenden es que todo llegue a su lugar, en el tiempo correcto, con la cantidad correcta, relacionado al tipo de escuelas que se definió anteriormente. Primero abastecer la Escuela Central, para luego distribuirla a las Escuelas Regional, pasando luego a las Escuelas de Sector, para por último pasar abasteciendo a las demás escuelas urbanas de cada Escuela de Sector.

La ventaja que se tiene es que para llegar a las diferentes escuelas los caminos están asfaltados, no se tienen kilómetros en terracería, entonces es mucho más fácil y factible distribuir todo.

d) Pre y Post Test. Al analizar los exámenes Pre y Post Test se concluyó que estaban fuera de nuestro alcance, ya que dichos exámenes tienen que ser aplicados por gente que tiene alguna especialidad o Licenciatura en Psicología o carrera a fin de la descrita anteriormente. En nuestro equipo de Megaproyecto se tiene únicamente a dos personas con Licenciatura en Educación, por lo cual se decidió pagarle a terceros para que ellos se encarguen de pasar los test. La prueba que se estará realizando será la Otis Lennon en formato A para el Pre Test y en formato B para el Post Test.

La logística de todos los exámenes, estará a cargo de la compañía contratada para la misma, es importante señalar que ellos se tendrán que movilizar para llegar a cada parte del país, y también son los encargados de la entrega de resultados.

La entrega de resultados y que se lleven a cabo en las fechas establecidas las diferentes pruebas, es vital, para ver el impacto que el proyecto tuvo durante todo el año escolar.

Debido a que se están trabajando con escuelas publicas, se debe tomar en cuenta los paros de magisterio por inconformidades, por sueldos, por paros o cosas por el estilo. Otra situación son la cantidad de actividades que manejan las escuelas, como excursiones, visitas a algunos lugares turísticos, semana de medio año, semana del día del maestro, ya que muchas actividades de esta índole pueden afectar el buen desempeño de dicho proyecto.

El factor tiempo es de vital importancia, se deben programar de la mejor manera el tiempo, ya que solo se puede contar con el tiempo de los maestros de 7:00 am a las 12:00 pm, los mismos no dan tiempo extra ya que atribuyen a los bajos sueldos.

La situación del país hacen que muchos maestros trabajen en jornada doble, lo cual hace que los rendimientos de los mismos no sean los más aptos, solo se puede trabajar con ellos de 7:00 am a 12:00 pm, por lo cual se tienen tiempos más largos para todo lo relacionado con capacitaciones y visitas.

V. DISEÑO COMPUTACIONAL

A. Paradigma del Lenguaje ABJ

El paradigma final se basó en el primer acercamiento al diseño del paradigma del lenguaje ABJ presentado en el marco teórico. Aún así se hicieron cambios sustanciales al mismo.

Es necesario resaltar un dato importante obtenido del primer acercamiento al diseño del paradigma, y es la ventaja de diseñarlo basado en conceptos concretos, familiares a los usuarios finales. Desde éste punto descartamos la posibilidad de desarrollar el proyecto basado en el paradigma integralmente *Orientado a Objetos* (OO), debido al grado de abstracción y complejidad que presenta. Ahora bien, no es necesario descartar la orientación a objetos por completo, argumento de ello es que si quitamos el elemento de abstracción que OO hace, aún existen varios elementos útiles para alcanzar los objetivos del proyecto, por ejemplo: diseño sensible a eventos, definición de estado y comportamiento de los objetos (concretos), entre otros. A esta especialización del paradigma Orientado a Objetos le llamaremos "Dirigido por la Historia" (DH), el cual se detalla a continuación:

1. Definición. DH es un paradigma derivado de OO, en el cual se propone una reducción en la abstracción del concepto *objeto*. Se le llama *Dirigido por la Historia* debido a que se propone que el usuario, al programar en DH, perciba la actividad como la narración de una historia en la que es necesario el planteamiento de un algoritmo para resolver una situación cotidiana.

2. Elementos.

a. Escenario y problema. Escenario es el ámbito de desarrollo en el que se presenta una disposición determinada de objetos, que en conjunto describen una situación (problema) que se desea resolver. Para resolver un problema es necesario que el usuario interactúe con los objetos en el escenario hasta lograr una configuración de estados tal que satisfaga una serie de reglas previamente definidas como parte del escenario mismo.

b. Satisfacción de condiciones. Es importante dejar toda la libertad posible en términos de las soluciones propuestas por los usuarios, esto es para que el programa sea utilizado de una forma no monótona, y para que potencie la creatividad y desarrollo de soluciones inteligentes a problemas comunes. Es por ello que un problema se considerará resuelto si un conjunto mínimo de condiciones es cubierto (satisfecho), no se pretende que exista una única solución a los problemas, ni que la alcanzada sea correcta únicamente si es optima.

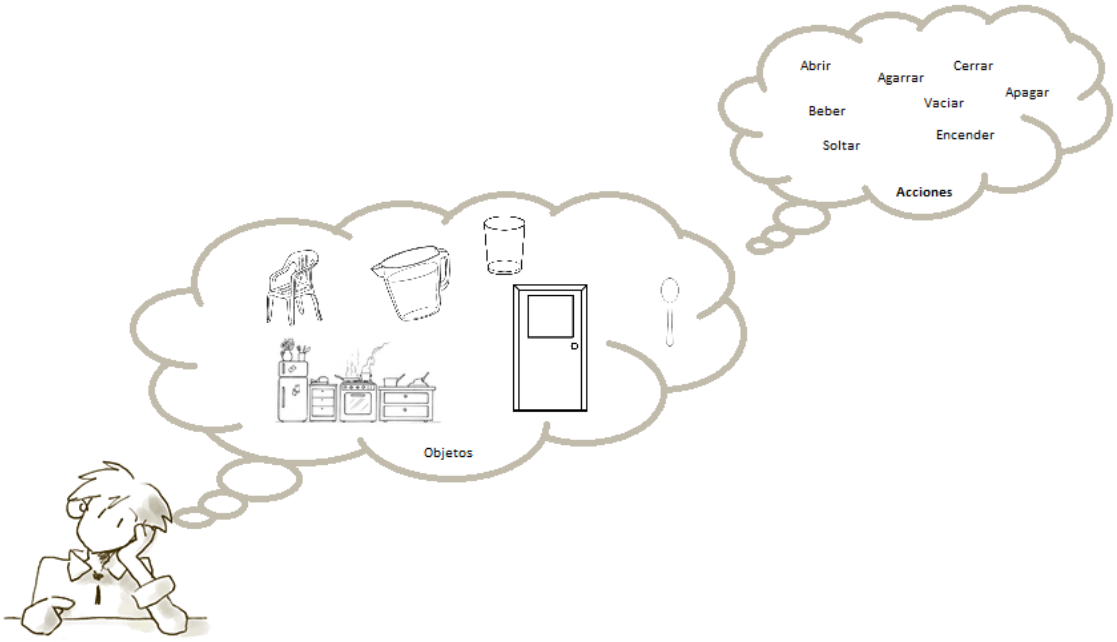
c. Héroe. Es un concepto que se utiliza como una referencia directa al término Historia. El héroe es un objeto concreto cuyo papel es el desarrollo de la escena, es decir, ejecutar los pasos necesarios para resolver la situación en discusión. Así pues, es natural para los usuarios acomodarse en el rol de héroe de la historia, puesto que comparten el mismo objetivo. Este concepto funciona como una referencia que ayuda a determinar las posibles acciones a realizar en la escena.

d. Sustantivos concretos. Son los elementos y herramientas de los cuales el usuario dispone para resolver la situación presentada por el escenario. Los sustantivos poseen propiedades como peso y tamaño, las cuales son utilizadas para determinar estados futuros luego de la aplicación de una o más acciones. Cada sustantivo debe referirse a un objeto concreto y visible.

e. Verbos y acciones. La manera en que el usuario interactúa con los objetos en el escenario es a través de diversos tipos de acciones disponibles según el objeto con el que se interactúa. Los verbos son los equivalentes a los métodos en OO, y tienen el objetivo de cambiar el estado de los sustantivos sobre los cuáles se aplican.

f. Mochila. Para lograr que el usuario interactúe de forma más natural con un escenario, se propuso la inclusión, dentro del mismo paradigma, del concepto de mochila. Este elemento es un contenedor adjunto al héroe, en el cual se pueden almacenar distintos elementos del escenario, con el objetivo de utilizarlos, cambiar su posición y completar tareas. Cada escenario provee al héroe de un distinto tipo de mochila, cuya variante depende de su contenido máximo.

Figura 15. Nube de ideas, utilizada para el diseño de los objetos (sustantivos concretos) y acciones (verbos).



3. Reducción de la abstracción. Se propuso e implementó una estrategia para la reducción de la abstracción del concepto *objeto* propuesto por OO, dirigida por las siguientes ideas:

a. Completa instanciación. Todos los objetos necesarios para resolver un problema se encuentran en el escenario. Es decir, no es posible instanciar los objetos debido a que ya fueron previamente instanciados todos los necesarios para resolver el problema.

b. Objetos de estado cerrado. Todos los objetos que se encuentran en un escenario pueden ser alterados únicamente a través de una interfaz pública provista. Esta interfaz responde a las preguntas ¿Qué puedo hacer con este objeto? Y, ¿qué acciones afectan a este objeto? Con este acercamiento se evitan manipulaciones directas sobre las variables de instancia de un objeto dado.

c. Carencia de valores de retorno. Las acciones o verbos no retornan valores al usuario, más bien cambian el estado de uno o más elementos en el escenario. Cada objeto es responsable de notificar al entorno su nuevo estado.

d. Otros conceptos. Conceptos prestados del paradigma OO son: interfaz, encapsulamiento y el principio de ocultación de la información.

B. Diseño del modelo computacional de enseñanza.

El diseño de programas es tarea difícil y a su vez, creativo. No existe un conjunto completo de reglas para escribir un programa, sin embargo, existen fases para la resolución de un problema. Este modelo solo se enfocará en las primeras tres fases, ya que el objetivo no es formar programadores como se dijo anteriormente, sino que el estudiante utilice metodologías y estructuras secuenciales, repetitivas y condicionales para analizar problemas y diseñar procesos o algoritmos para su solución. Con esto se asegura que los estudiantes desarrollen competencias de comprensión lectora.

1. Proceso de diseño de un programa (Hernández, 2010). Es importante resaltar que es imposible enseñar a resolver problemas, pero es posible que el estudiante aprenda a resolverlos, siempre con una actitud positiva y de manera entretenida en su desarrollo. Y ésta es la finalidad del modelo ABJ, que es el modelo que adaptará la forma de enseñanza de algoritmos en los estudiantes de primaria.

2. Prototipo de ejercicios. Para los ejercicios que se incluirían en la herramienta computarizada se utilizó la gramática ABJ presentada en el marco teórico.

Diseñar ejercicios los cuales puedan ser resueltos por el estudiante tanto con la herramienta computarizada como la no computarizada, esto con el fin de practicar el diseño de algoritmos y poder así aplicarlos en su vida cotidiana. Algunos de los ejercicios serían previamente cargados en la herramienta de software, es por eso que al obtener el prototipo final de estos, el equipo de computación los utilizará para realizar experimentos y poder así incluirlos exitosamente en la herramienta.

- Los criterios utilizados para la elaboración del prototipo fueron los siguientes:
- El estudiante podría tener experiencias pasadas sobre los ejercicios o actividades.
- Experimentos o tareas de grados anteriores.
- No requiera de gran elaboración, ya que el estudiante podría aburrirse y desesperarse al resolverlo.
- Entretenido y útil, por ejemplo la realización de actividades extracurriculares como muñecos de trapo, juguetes, entre otros.

- Que aliente a los estudiantes a analizar otras posibles soluciones y sembrar curiosidad por compararlas con los demás estudiantes.

C. Diseño del software.

1. Diagrama de casos de uso. Ver Figura 11 en página 54.

2. Arquitectura de software propuesta. Para desarrollar los entregables se propone trabajar utilizando el patrón de diseño *Modelo-Vista-Controlador (MVC)*. Se decidió por éste patrón de diseño porque provee la ventaja de separación de tareas y modularidad. Además propone una manera bastante natural de dividir el trabajo. A continuación se describen cada una de las capas de dicha arquitectura aplicada al megaproyecto.

a. Modelo. El modelo representa el conjunto de datos y las relaciones que los rigen, cuya finalidad es dirigir la ejecución del programa. Para el megaproyecto ABJ, el modelo representará el estado inicial de un problema, las características ambientales que regirán la solución algorítmica del mismo y las descripciones de estado e interfaz provistas por los elementos contenidos en el escenario.

b. Vista. Clases encargadas del despliegue en pantalla de la información pertinente para la resolución de un problema. Tienen como objetivo adicional la recepción y reporte de eventos. Está compuesta por la ventana del problema y los complementos de interfaz.

c. Controlador. Todos los eventos recibidos de la vista son procesados por el intérprete. Para que el intérprete pueda analizar la entrada, es necesario que los comandos sean convertidos en comandos del lenguaje ABJ. Para ello se introduce el módulo de *Traductor inverso*, descrito a continuación:

1) Generador de comandos. Este módulo se encarga de convertir un conjunto de eventos reportados por la vista, en el comando en lenguaje ABJ que realiza las acciones solicitadas por dichos eventos. Se le denomina Traductor inverso debido a que realiza el proceso inverso del intérprete. Este módulo contiene dos componentes:

a) Generador de número y género. Determina los artículos correspondientes al número y género de un objeto dado. Los artículos se obtienen de un diccionario. El artículo generado se transmite al traductor inverso, que es el encargado de armar los comandos.

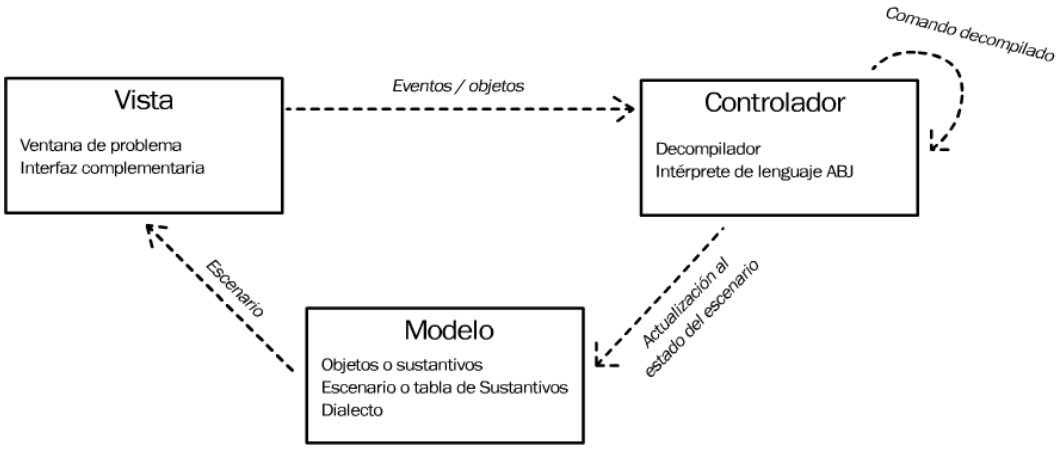
b) Traductor inverso. Este módulo se encarga de convertir un conjunto de eventos reportados por la vista, en el comando en lenguaje ABJ que realiza las acciones solicitadas por dichos eventos. Se le denomina Traductor inverso debido a que realiza el proceso inverso del intérprete.

Éste recibe un objeto que envía al generador de número y género. De esta forma puede concatenar la acción, el artículo correcto y el objeto para generar el comando correctamente.

Se llama desde iAccion, y es allí donde se construye el comando. iAccion es una interfaz implementada por todas las demás clases de acciones.

A continuación se propone un diagrama de la comunicación que se dará entre cada nivel de la arquitectura propuesta:

Figura 16. Interacciones entre cada uno de los niveles de arquitectura Modelo-Vista-Controlador propuesto como patrón de diseño para el megaproyecto ABJ

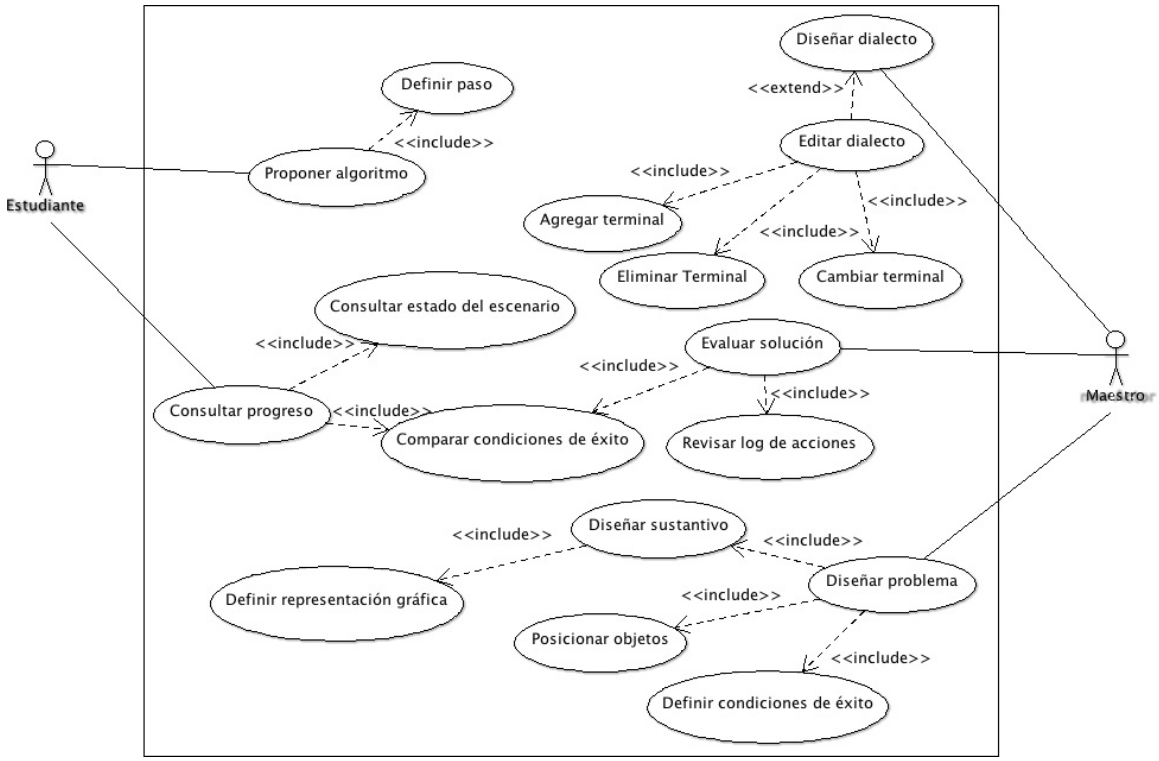


3. Especificación de los entregables. La solución de software consistirá de dos entregables principales:

a. Intérprete del lenguaje ABJ. El intérprete engloba la parte lógica del proyecto, de forma genérica se puede ver cómo el traductor del proyecto, toma la entrada en lenguaje ABJ y la transforma a estructuras reconocibles por Python y su librería NLTK para así ejecutar los comandos solicitados. Tiene carácter autónomo y modular, puede ser utilizado por varias interfaces externas, entre las propuestas están una consola interactiva y una interfaz gráfica de resolución de problemas. Los componentes principales del intérprete son:

1) Jerarquía de sustantivos. Jerarquía de clases que definen las características (estado) y acciones realizables (interfaz) de cada sustantivo dentro de un escenario. Estas características definen la forma en que un objeto interactúa con el entorno (escenario) y con el usuario (héroe). Además, dentro de esta jerarquía debe incluirse la descripción del sustantivo especial héroe.

Figura 17. Diagrama de casos de uso de la solución de software para el Megaproyecto.



2) Lector de escenarios de problema. Descripción espacial de los elementos que componen un problema, incluyendo el héroe del mismo. Para cada problema propuesto debe existir un escenario que rige su estado inicial. Durante la ejecución del sistema, el escenario fungirá como una *tabla de sustantivos*, en la que mientras el usuario interactúa con el sistema, se mantiene record del estado actual del problema.

3) Lector de dialectos. Es posible definir distintos conjuntos de palabras que regirán la ejecución del sistema, y definirán el lenguaje en el cual se puede expresar el algoritmo de solución al problema cargado. Cada uno de éstos conjuntos de palabras (específicamente terminales) se denomina *dialecto*.

Una de las consecuencias más interesantes de la inclusión de dialectos en el sistema es que un mismo problema puede tener dos soluciones computacionalmente idénticas, pero que en términos de lenguaje fueron expresadas de formas distintas. Esta descripción debe ser cargada al sistema según los requerimientos de su ejecución.

4) Analizador léxico. Toma una hilera de caracteres en lenguaje ABJ y genera una secuencia de tokens (pares nombre, valor) obtenidos del emparejamiento de un segmento de la hilera de entrada con la expresión regular correspondiente.

5) Analizador sintáctico. A partir de la hilera de tokens devuelta por el analizador léxico, utiliza las facilidades de la librería NLTK de Python para tratar de asignarle una estructura de árbol a cada comando de entrada. Si la entrada pertenece al lenguaje ABJ, se pobla la estructura de árbol original con la información correspondiente a cada token; de lo contrario debe notificarse el problema ocurrido al ambiente de ejecución.

6) Analizador semántico. El analizador semántico toma información de un archivo diccionario en lenguaje XML donde están especificadas los terminales para acciones y artículos permitidos dentro del lenguaje. Debido a que el proyecto fue planificado con la posibilidad de crear y editar problemas en el futuro, no se restringió el vocabulario aceptado, sino que se almacenó en el diccionario con una estructura que, leída por el analizador semántico, crea dinámicamente los objetos utilizados en los problemas a trabajar. Las características de los elementos del diccionario se especifican en sus atributos, los cuales poseen la información pertinente a número (plural, singular) y género del elemento en cuestión. Si una palabra es ejecutable, se le identifica en su género como un verbo. El analizador semántico no recibe parámetros de entrada, y en la salida devuelve tres estructuras de diccionario de Python, las cuales son parejas <clave, valor> donde clave y valor pueden ser objetos o valores alfanuméricos.

7) Ejecución semántica. Durante esta fase se hace un recorrido al árbol generado por el analizador sintáctico para revisar concordancia de género entre artículos y sustantivos, revisión de aplicabilidad de acciones (por cercanía o cualquier otro atributo de los sustantivos en juego), entre otros. Si ocurre un error durante esta fase debe notificarse al ambiente de ejecución, de lo contrario debe actualizarse el estado de la tabla de sustantivos (escenario).

A continuación se presentan cinco diagramas de caja que representan las interacciones a nivel abstracto que ocurren en el intérprete del lenguaje ABJ.

Figura 18. Proceso de inicialización del intérprete de lenguaje ABJ

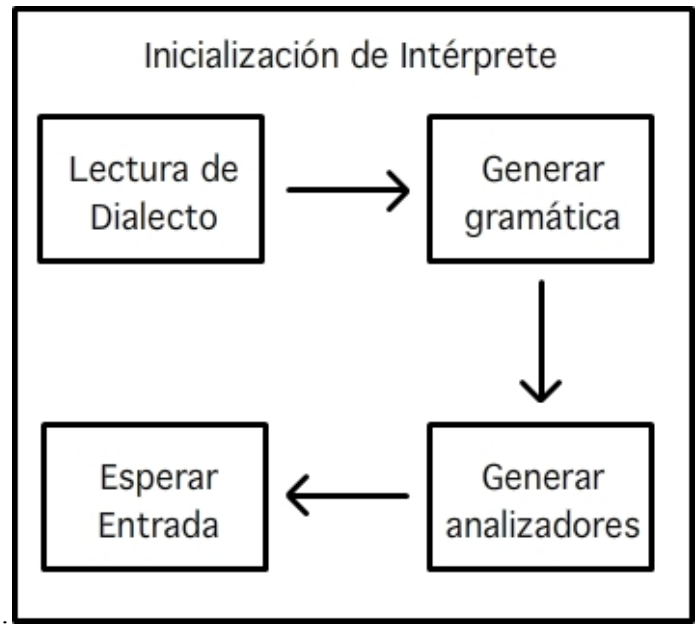


Figura 19. Proceso de traducción interpretada de una entrada en lenguaje ABJ.

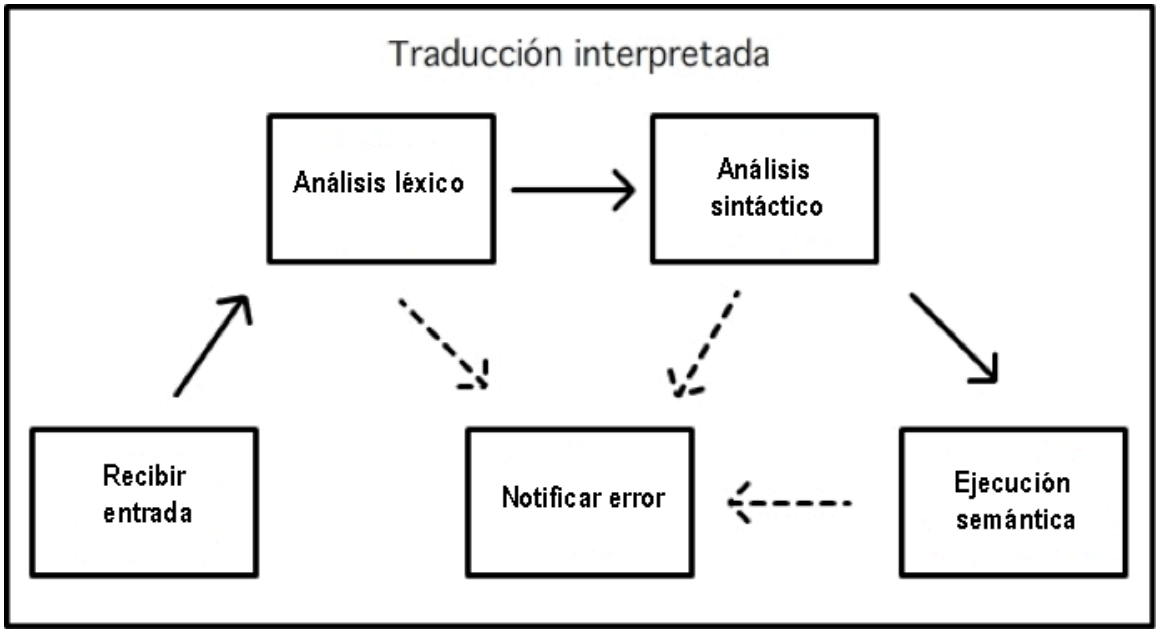


Figura 20. Proceso de análisis léxico para el intérprete del lenguaje ABJ.

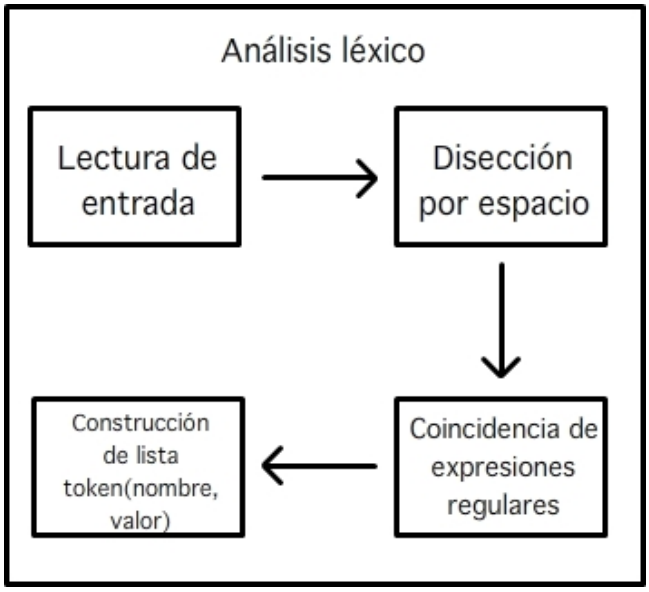


Figura 21. Proceso de análisis sintáctico para el intérprete del lenguaje ABJ

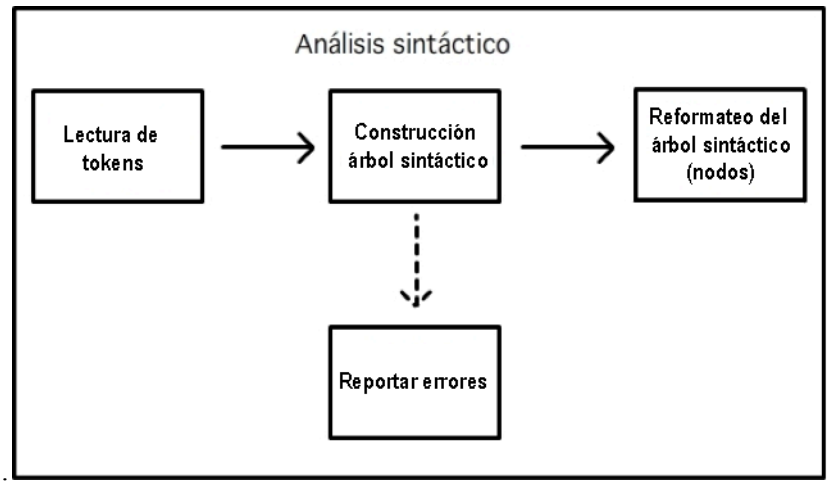
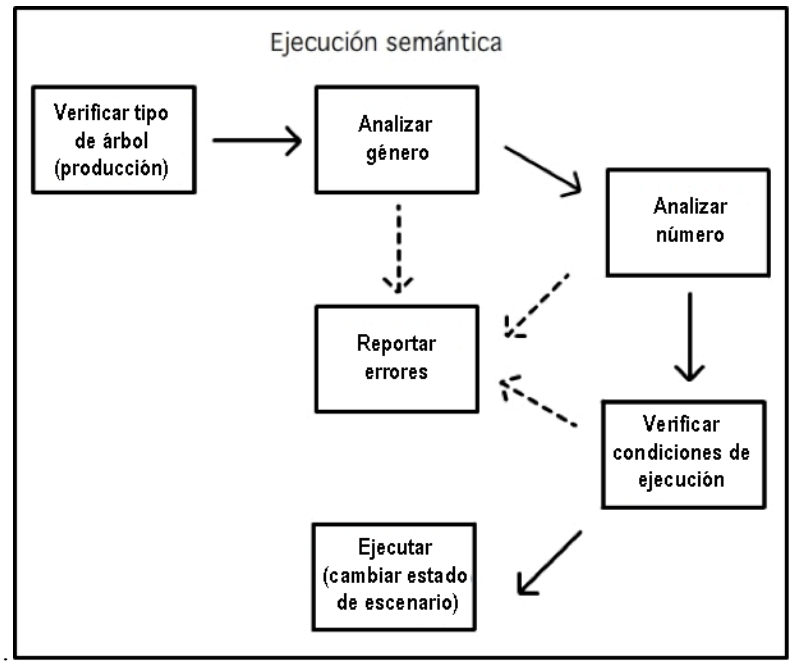


Figura 22. Proceso de ejecución semántica para el intérprete del lenguaje ABJ



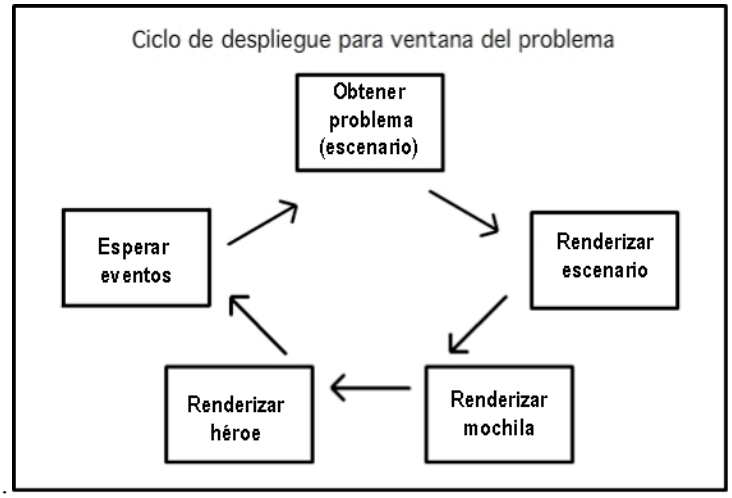
b. Interfaz de resolución de problemas. Este entregable se encarga de conectar las acciones del usuario con el módulo lógico del proyecto. Se divide en dos componentes principales:

1) Ventana de problema. Se encarga de tomar un escenario y desplegar su estado actual en pantalla (desplegar la tabla de sustantivos). Utiliza la información de cada sustantivo, y del escenario en general, para desplegar el estado coherente del problema en pantalla.

2) Complementos gráficos. Junto con la ventana de problema debe de implementarse un conjunto de complementos para ayudar a la interacción con el usuario. Entre estos complementos se pueden mencionar una interfaz especializada de la mochila y una bitácora de acciones realizadas.

A continuación se presenta un diagrama de cajas que explica el funcionamiento de la ventana de problema, específicamente se describe su ciclo de despliegue:

Figura 23. Diagrama de cajas del ciclo de despliegue de la ventana del problema



El diseño final de la interfaz gráfica se basó en las técnicas y teoría sobre usabilidad presentada en el marco teórico. Fue necesario realizar varias iteraciones sobre el diseño inicial para llegar hasta el resultado actual.

El proceso de diseño se detalla a continuación:

1) Estudio de benchmark. El estudio de benchmark se realizó con el objetivo de establecer metas reales para la usabilidad del megaproyecto y determinar los aspectos utilizados por los programas evaluados, para facilitar el aprendizaje de los niños. Entre los programas considerados para realizar el estudio se encuentra, Alice, BlueJ, EToys, Scratch.

Se realizó una investigación sobre los distintos programas orientados a enseñar a un niño a programar. Finalmente se realizó la selección, determinando que Alice y Scratch poseían la interactividad y atractivo visual que quería lograrse con la interfaz gráfica del megaproyecto.

Se elige Alice por su orientación a contar historias, mientras que Scratch llama la atención por el aspecto desprolijo de las imágenes que utiliza. Además ambos ambientes cuentan con la opción de arrastrar y soltar elementos gráficos de forma libre. Etoys permite realizar acciones similares, sin embargo la forma de presentar opciones es confusa y resulta una herramienta poco usable. La orientación de BlueJ resulta ser inadecuada para niños y Everydaymathematics, a pesar de ser un programa de estudios con características deseables está fuera de los objetivos del benchmark.

En el diseño del estudio de benchmark se establece el perfil del usuario final. Inicialmente, se plantea la idea de realizar el estudio por medio de pruebas de usabilidad a usuarios que cumplieran con el perfil definido, para determinar el nivel de usabilidad de ambos. Sin embargo,

por motivos de optimización de tiempo y recursos, se opta por realizar una investigación profunda, tanto teórica como práctica, por medio de la exploración de ambas herramientas.

2) Diseño de pruebas de usabilidad. Se opta por realizar un diseño de pruebas de usabilidad de tipo iterativo. Así que el diseño de este módulo se divide en dos pruebas de usabilidad realizadas, la inicial, realizada sobre un prototipo en papel, y la final, sobre la herramienta completamente funcional.

En ambos casos surge la necesidad de la adaptación de la prueba tradicional para hacerla apta para niños, y de este modo lograr resultados que no estén contaminados por interferencia del evaluador o bien por la complejidad de las instrucciones.

Para el mismo fin, se realiza una pequeña capacitación a los evaluadores, haciéndoles comprender la forma en la que debían comportarse con los niños y evitar sesgar sus opiniones o intimidarlos. Además se recalca la importancia de observar tanto las acciones como los gestos de los niños al recibir instrucciones y realizar tareas. Se advierte a los evaluadores que los niños pedirán ayuda y se muestran las formas en las que deben evadirlos o explicarles con gentileza que no les es posible suministrar ayuda.

3) Usabilidad en prototipo en papel. En la primera prueba se definieron tareas para probar la configuración de las opciones y los menús dentro de la interfaz. Al utilizar un prototipo en papel surge la necesidad de extender la explicación inicial para hacer entender al niño, la forma en la que se evaluará. La utilización de símiles es importante en ésta fase, ya que se le indica al infante que su dedo es equivalente al puntero del mouse.

El objetivo para el diseño de ésta prueba residió en determinar los siguientes aspectos:

- Facilidad y efectividad
- Utilidad de contenido
- Efectividad de la presentación
- Comodidad del usuario

a) Perfil de usuario. Para esta prueba se eligió una muestra de cinco niños, en un rango de edades de ocho a trece años. Evaluados por dos personas.

4) Usabilidad en herramienta funcional. Una vez estuvo terminada la herramienta gráfica, se unificó con la herramienta lógica, permitiendo realizar una segunda iteración en el ciclo de pruebas. El diseño para la segunda prueba se enfoca en medir la usabilidad de la herramienta como un todo, ya que la usabilidad de la interfaz gráfica se midió en la prueba sobre el prototipo en papel.

Se definen tareas indispensables para la resolución de un problema ABJ, buscando determinar la respuesta a las siguientes preguntas:

- ¿El niño se siente cómodo con el ambiente gráfico (ubicación de los elementos, aspecto de las imágenes)?
- ¿El niño puede realizar las acciones básicas (Buscar, agarrar, mover)?
- ¿La forma en la que se presentan las opciones es intuitiva (click derecho y botones)?

a) Perfil de usuario. Para esta prueba se elige una muestra de quince niños, cursando quinto primaria.

5) Análisis y aplicación. Una vez diseñadas las preguntas, el objetivo y las actividades que querían ser puestas a prueba durante las pruebas de usabilidad se debía pasar a la fase de implementación.

Se dirigiría al colegio Suger Montano de la z. 10, para obtener la autorización de utilizar su laboratorio de computación para hacer pruebas sobre la herramienta a aproximadamente 10 niños. A cada niño se le explicaría a grandes rasgos lo que se iba a hacer, y se le harían una serie de preguntas previamente diseñadas. Es muy importante no influir en la respuesta del niño ni explicarle cómo contestar las preguntas ni cómo utilizar la interfaz detalladamente.

Al finalizar, se debían tabular las respuestas y determinar los cambios que se le debían hacer a la interfaz gráfica de la herramienta.

6) Diseño de mochila. La mochila fue planteada al inicio del megaproyecto y verificada en los resultados de usabilidad y de benchmark en los cuales los niños reconocieron el papel de la mochila como un lugar donde puede guardar los elementos mientras se mueven por todo el escenario.

La mochila es un elemento tipo ABJFrame que fue hecho a la medida para que fuera capaz de aceptar objeto ABJLabel al inicio, durante y el fin de arrastrar y soltar.

La mochila es un elemento que siempre está presente durante toda la ejecución del programa y esto para facilitar el manejo de objetos dentro del escenario: como moverlos dentro de contenedores o guardarlos para futuros usos dentro del problema.

7) Bitácora de acciones. Inicialmente se diseña la bitácora como un complemento gráfico flotante que puede ser posicionado al gusto del usuario. Se utiliza un objeto tipo vista de árbol llamado QtreeWidget dónde se crea y agrega un QtreeWidgetItem como topLevelItem para cada acción realizada.

a) Recuperación de acciones realizadas. Para determinar si una acción genera una entrada en la bitácora, se utiliza el método runCommand () que devuelve distintos valores según la visibilidad de la acción en la bitácora. Luego son desplegadas en una vista de la bitácora.

8) Ligadura de acciones a íconos. Cada acción tiene asociada una imagen que se agrega a cada QTreeWidgetItem de la bitácora. Cada clase de acción almacena la imagen que se despliega en la bitácora.

9) Diseño de imágenes. Para el diseño de imágenes se toman en cuenta los resultados del estudio de benchmark, ya que se desea lograr un aspecto en los objetos, parecido al de Scratch.

Se presta especial atención en los colores utilizados, aplicando la teoría del color para hacer énfasis en los objetos para simular una posición en el eje z. Además es resulta necesario utilizar colores neutros, evitando un posible rechazo de los niños si se utilizan colores que generalmente son utilizados por niñas, y viceversa.

10) Posicionamiento dentro de la herramienta gráfica. Después del segundo test de usabilidad se determina que el diseño original como un complemento gráfico flotante siempre visible, es confusa para el usuario. Es así que se decide que esté invisible por defecto. Al buscar la opción para desplegar la bitácora, ésta se hace visible y es posible reposicionarla en cualquiera de los bordes de la ventana, para integrarla a la vista principal.

Las acciones presentadas en la bitácora se exportan a un archivo de texto que se almacena en la carpeta “Soluciones” al guardar el problema.

11) Implementación de funcionalidades de usabilidad de la herramienta.

a) Implementación de cuadros de diálogo de la herramienta computarizada. Uno de los aspectos determinados durante las pruebas de usabilidad de la herramienta computarizada era la necesidad de poder retroalimentar al niño el estado de la herramienta, problemas y/o errores.

Para esto se diseñó un método que recibe una cadena de caracteres a desplegar y éste despliega el diálogo con la cadena de caracteres enviados. Además, contiene un ícono de alerta para simbolizar el tipo de diálogo que es.

Se utilizó la herramienta QtDesigner para poder determinar el tamaño del diálogo necesario. Luego con el framework de PyQT de Python, se desarrolló el código para implementarlo.

b) Implementación de atajos con el teclado para realizar acciones en la herramienta computarizada. Para la implementación de los keybindings primero se elaboró un estándar a seguir en relación a qué teclas iban a corresponder a qué acciones, por lo que se llegó al siguiente consenso:

Cuadro 10. Atajos con el teclado.

Teclas	Acción
"A" o "a"	Abrir
"C" o "c"	Cerrar
"G" o "g"	Guardar
"S" o "s"	Soltar
"M" o "m"	Moverse

c) Implementación de menús contextuales a través del click derecho en la herramienta computarizada. Para poder completar este módulo se utilizó una función de PyQT que se llama ContextMenuRequested. Esta función me permite capturar el evento del click derecho para desplegar un menú contextual.

Gracias al buen manejo de la lógica del sistema, cada objeto en la pantalla contiene su propio set de acciones que se pueden realizar sobre él. Para cada acción se elaboró un ícono representativo para mayor usabilidad y ser más agradable a la vista de los niños.

d) Implementación de visualización de objetos seleccionados. El usuario debía ser capaz de visualizar el objeto que tenía seleccionado, ya sea en el área de acción o en la mochila. Para esto se diseño un método que al momento de hacer click sobre un objeto, cambiara la imagen actual por una con el mismo nombre pero dentro de un directorio relativo con el nombre de "seleccionados", la diferencia es que esta nueva imagen tiene los colores invertidos.

e) Implementación del arrastrar y soltar. Los usuarios pueden tomar y arrastrar los elementos que estén en el escenario y la mochila. Es decir el usuario con facilidad puede tomar cualquier elemento que tenga la propiedad de ser arrastrable y depositarlo tanto en la mochila como en el escenario.

Para hacer esto solo se tiene que presionar el botón izquierdo sobre la imagen y mantenerlo presionado hasta que la imagen se empiece a mover, este elemento se puede mover entre la mochila y el escenario.

Para soltar el elemento depende de su estado anterior, es decir que si un elemento estaba en el escenario solo puede ser soltado en los contenedores que tiene cerca o puede ser soltado en la mochila, de lo contrario el elemento regresará a su posición original.

4. Diccionario de clases y métodos. A continuación se describen las clases, atributos y métodos más importantes que integran el proyecto. De ellas se excluyen la jerarquía de interfaces implementada por *Objeto*, y sus elementos más importantes se resumen en dicha clase:

a. TerminalProcessor. Permite realizar la separación entre los terminales y la gramática propuesta. Provee los métodos necesarios para leer un dialecto de terminales y construir los diccionarios de género y número.

1) dom. Atributo generado como resultado del análisis del archivo de entrada del dialecto. Representa el DOM de un archivo XML.

2) **__init__.** Constructor. Recibe de parámetros el procesador de terminales que se iniciará y el nombre del archivo de dialecto con el cual se creará.

3) getTerminals. A partir de un recorrido al dom, construye los diccionarios de género, número, artículos y sustantivos.

b. Nodo. Representa un Nodo en un árbol sintáctico. Contiene la información necesaria para manejar eficientemente el análisis semántico (producción o terminal representado).

1) lexema. Valor de la hilera de caracteres leída e interpretada como token durante la fase de análisis léxico.

2) token. Nombre de la expresión regular con la cual el segmento de entrada obtuvo concordancia durante la fase de análisis léxico.

3) indice. Referencia al número de segmento correspondiente al token actual en la hilera de caracteres de entrada del analizador léxico.

4) __init__. Constructor. Para inicializar un Nodo es necesario especificar su nombre, lexema e índice de entrada.

5) __str__. Obtiene una representación de la forma <<Nodo (nombre, lexema, índice)>> para el Nodo actual.

c. LogicoABJ. Representante de la arquitectura a nivel controlador, se encarga de realizar el manejo del programa en su parte lógica. Contiene las funciones del intérprete y posee acceso directo al estado de un problema (escenario).

1) problema. Representación genérica (Escena) de un problema en memoria. Los comandos ejecutados por LogicoABJ afectan el estado de problema.

2) heroe. Referencia al héroe del problema, se utiliza para guiar la ejecución de comandos a nivel semántico (evaluación de cercanía, postura y capacidad).

3) keywords. Diccionario de datos que relaciona una expresión regular con el conjunto de palabras reservadas (nombres de no terminales en la gramática).

4) sinonimos. Diccionario de datos que relaciona un sustantivo, verbo o artículo con un conjunto de sustantivos, verbos o artículos (respectivamente) con los que comparten la misma función semántica.

5) Genero. Diccionario de datos que relaciona un artículo o un sustantivo con un género (masculino, femenino y neutro).

6) gramatica. Objeto procesado por la librería NLTK que representa la gramática libre de contexto a usar para interpretar el lenguaje ABJ.

7) rdParser. Objeto del NLTK para representar el parser descendente de la gramática generada.

8) __init__. Constructor. Recibe como parámetros la dirección en la que se encuentra el dialecto a utilizar y el nombre del héroe (para futuras referencias al usuario del software).

9) matchRegex. Regresa el nombre de la expresión regular con la que la hilera de caracteres, pasada como parámetro, coincide.

10) matchKw. Verifica si la hilera pasada como parámetros es una palabra reservada.

11) nodificar. Servicio interno que convierte un árbol sintáctico NLTK en un árbol sintáctico cuyos terminales son del tipo Nodo. Este proceso se hace con el fin de facilitar la ejecución semántica poblando el árbol sintáctico con la información necesaria durante el recorrido (identificador de token y valor de lexema).

12) comparaGenero. Dado un sustantivo y un artículo, verifica si ambos corresponden al mismo género. Se utiliza en la ejecución semántica para verificar la existencia de fórmulas del tipo “corro hacia el mesa”, donde el artículo “el” y el sustantivo “mesa” tienen género distinto.

13) ejeucionSemantica. Toma de entrada el árbol sintáctico generado en la fase anterior y lo recorre para verificar compatibilidad de las oraciones tanto en género y número. Además verifica la aplicabilidad del comando según el estado del escenario. Si todas las condiciones se cumplen, se cambia el estado de problema.

14) runCommand. Interfaz de comunicación con el intérprete ABJ. Ejecuta todas las fases de la traducción interpretada.

d. Acción (verbo). Esta clase representa un verbo el cual se puede ser aplicado a un objeto o al héroe para cambiar su estado. Por lo tanto éstas van a regir el comportamiento de todos los objetos del escenario de un problema.

Cualquier acción (abrir, verter, beber, etc.) deben heredar de 1la clase IAccion, para tener compatibilidad con el escenario de ABJ (Véase Figura X3).

1) en mochila. Indica si la acción se puede realizar solamente cuando se tiene dentro de la mochila.

2) es heredable. Indica si la acción de un objeto contenido en otro objeto pasa a ser parte del contenedor. Por ejemplo el agua se puede revolver, si se coloca dentro de un pichel este heredara la propiedad de revolver el agua.

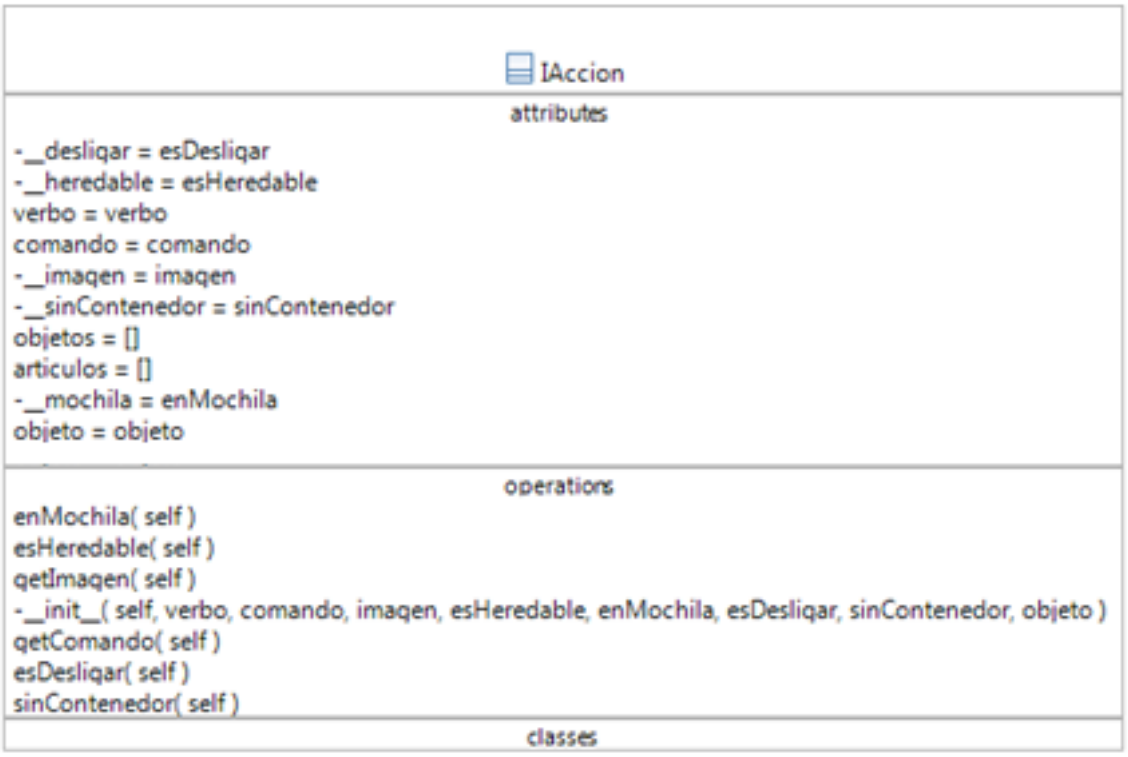
3) comando. Es el comando creado a partir de la lista de artículos y objetos que se hayan proveído a la acción. Dicho en otras palabras, es la oración que se enviará al intérprete.

Este comando puede contener un verbo y uno o más artículos y objetos. Un ejemplo puede ser "abrir la puerta", el comando se vería de la siguiente forma "{verbo} {articulo} {objeto}".

4) imagen. Representa la dirección relativa del archivo de imagen utilizado para mostrar en la bitácora de comando ejecutados.

5) sincontenedor. Indica si la acción se puede realizar sin que el objeto este contenido en otro.

Figura 24. Diagrama de clase para el controlador de acciones.



e. Objeto. Esta clase representa a cada sustantivo en un escenario de un problema. Tiene relación con el módulo gráfico y con el módulo lógico, por ello debe llevar el control de ambas categorías de datos.

1) posicion. Variable de control que indica las coordenadas en dos dimensiones en las que el objeto se encuentra y las cuales deben ser utilizadas para renderizar el escenario.

2) peso. Variable utilizada para saber si la mochila está capacitada para contener al objeto.

3) imagen e imagenMochila. Representan la dirección relativa de los archivos de imagen utilizados para representar al objeto tanto en la ventana del problema como en la mochila.

4) nombre. Identificador del objeto.

f. Heroe. Esta clase representa al protagonista en la solución de un problema en el paradigma ABJ. Puede cambiar de postura, tiene una representación gráfica y puede contener (guardar en mochila) algunos objetos del problema.

1) posicion. Variable de control que indica las coordenadas en dos dimensiones en las que el héroe se encuentra y las cuales deben ser utilizadas para renderizar el escenario.

2) postura. Variable utilizada para saber si el héroe se encuentra parado, sentado o acostado en un momento determinado de la vida del objeto héroe que lo posee.

3) imagen. Representa la dirección relativa del archivo de imagen utilizado para representar al héroe en la ventana del problema.

4) Elementos. Debido a que el concepto de héroe está ligado al de mochila, consideraremos a un héroe como un contenedor (siendo éste la mochila). El diccionario de elementos representa todos los objetos que el héroe contiene en su mochila en un momento dado de la ejecución del programa.

5) nombre. Identificador del héroe. Utilizado para mostrar mensajes según el usuario del sistema.

g. Escenario. Representación de cada problema, contiene un árbol de sustantivos espacialmente posicionados para representar la situación a resolver. Posee también las

funcionalidades necesarias para guardar una bitácora de acciones, evaluar el progreso de un problema y condiciones de éxito del mismo. Siempre que se desee crear un nuevo problema, se tiene que heredar de la clase Escena, la cual proveerá acciones útiles para el problema y esto permitirá mantener un estándar o contrato entre los problemas creados.

1) nombre. Hilera de caracteres asignada para identificar el problema representado.

2) descripcion. Texto introductorio al problema. Este texto se muestra al iniciar la aplicación para dar una guía general de los resultados esperados al implementar un algoritmo para su solución.

3) objetos. Diccionario de los sustantivos contenidos en el escenario.

4) heroe. Referencia al objeto héroe de la escena. Al iniciarse un problema, se cargan los datos de instancia del héroe cristalizado en la definición del problema.

5) background. Dirección relativa a la imagen que representará visualmente al héroe.

6) muestraFlechas. Variable que indica si la interfaz de usuario debe mostrar las flechas de desplazamiento básicas del héroe (arriba, abajo, izquierda y derecha).

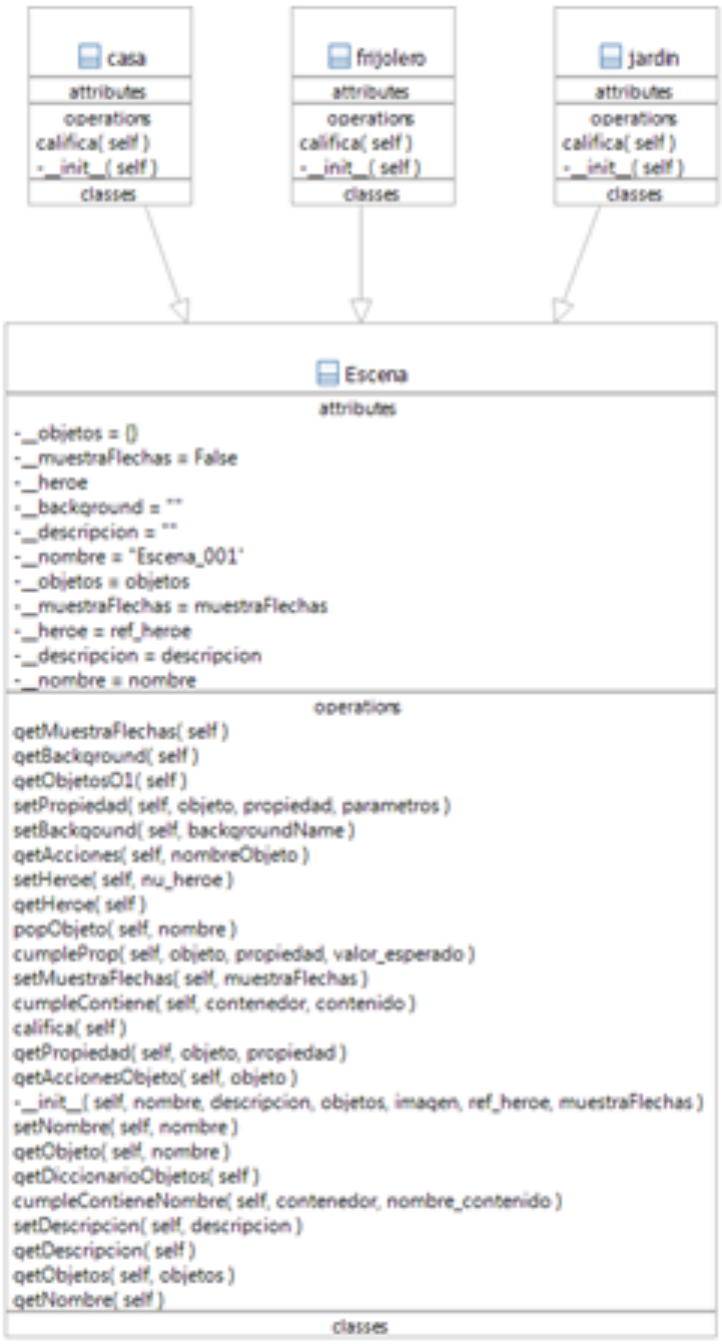
7) cumpleContine y cumpleContieneNombre. Dado un contenedor y un sustantivo (o su identificador), verifica si éste último se encuentra adentro del primero (ya sea de forma directa o a través de la relación “contenido en” aplicada más de una vez).

8) getAccionesObjeto. Recorre la jerarquía de implementación de interfaces de un objeto dado y devuelve la lista de acciones aplicables al mismo.

9) getPropiedad y setPropiedad. Dado un sustantivo y una hilera de caracteres representando una propiedad del objeto, la accede (es decir, la lee o la asigna).

10) cumplePropiedad. Verifica si una propiedad dada, de un objeto dado, posee el valor especificado.

Figura 25. Diagrama de clase para el diseño de los escenarios.



h. ABJLabel. Esta clase representa una de las uniones entre el hemisferiológico y el hemisferio gráfico, se encarga de mostrar en pantalla el estado de un sustantivo en el escenario, y de reportar si un evento ha sido ejecutado sobre dicho sustantivo.

1) seleccionado. Lista que almacena referencias a todos los objetos seleccionados en el escenario.

2) objeto. Sustantivo (objeto lógico) representado por el ABJLabel.

3) enMochila. Valor binario que indica si el sustantivo contenido se encuentra en la mochila del héroe.

4) Eventos del mouse. ABJLabel responde a eventos de menú contextual (clic derecho) y a los eventos de liberación de los botones del ratón.

i. ABJFrame. Clase representante de la ventana de problema y el panel de la mochila. En ella se despliegan los ABJLabel necesarios correspondientes al estado actual del problema.

1) labels. Referencia a la lista de labels desplegados en el escenario.

2) Eventos del mouse. Responde a los eventos de inicio de arrastrado y fin de arrastrado.

j. VentanaPrincipal. Interfaz gráfica (vista) que contiene los complementos gráficos, entiéndase la bitácora de acciones, la interfaz de la mochila y los botones estándar en una aplicación de ventana.

1) sele y seleMochila. Almacenan los objetos seleccionados en la ventana del problema y en la mochila respectivamente.

2) lógico. Referencia a un objeto LogicoABJ, representa el controlador de la aplicación.

3) prob. Problema a resolver. Representa el estado actual del escenario.

4) canvas_lbls y mochila_lbls. Objetos gráficos a desplegar en la ventana del problema y en la mochila respectivamente.

5) vacia_mochila y vacia_canvas. Eliminan todos los elementos gráficos de la mochila y la ventana del problema respectivamente.

6) refresca_mochila y refresca_canvas. Vuelven a dibujar el estado actual de la ventana del problema y de la mochila respectivamente.

7) mover, ligar, buscar y desligar. Con la información de los objetos seleccionados, estos métodos construyen un comando en lenguaje ABJ para que el intérprete lo ejecute y cambie el estado actual del problema.

8) cargaProblema. Lee el estado inicial de un problema en disco. Actualiza la tabla de sustantivos y se asegura de refrescar la pantalla.

9) main. Método llamado por el sistema al iniciar a correr el programa. Acá se inicializan los objetos necesarios para desplegar la ventana principal y utilizar al intérprete.

Figura 26. Primera parte del diagrama de clases propuesto para la solución de software del megaproyecto ABJ.

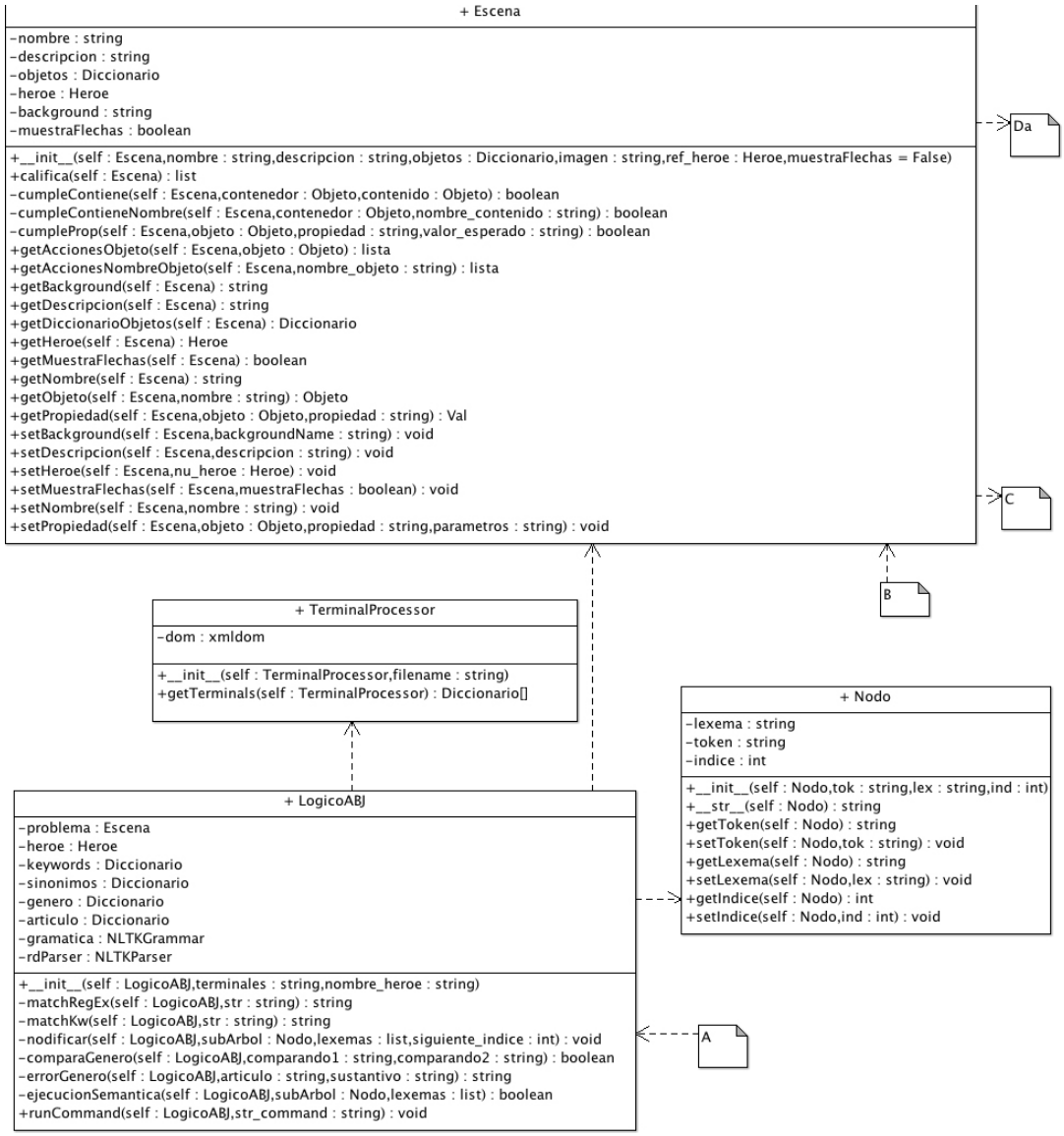

+ Escena
-nombre : string
-descripcion : string
-objetos : Diccionario
-heroe : Heroe
-background : string
-muestraFlechas : boolean
+__init__(self : Escena,nombre : string,descripcion : string,objetos : Diccionario,imagen : string,ref_heroe : Heroe,muestraFlechas = False)
+califica(self : Escena) : list
-cumpleContiene(self : Escena,contenedor : Objeto,contenido : Objeto) : boolean
-cumpleContieneNombre(self : Escena,contenedor : Objeto,nombre_contenido : string) : boolean
-cumpleProp(self : Escena,objeto : Objeto,propiedad : string,valor_esperado : string) : boolean
+getAccionesObjeto(self : Escena,objeto : Objeto) : lista
+getAccionesNombreObjeto(self : Escena,nombre_objeto : string) : lista
+getBackground(self : Escena) : string
+getDescripcion(self : Escena) : string
+getDiccionarioObjetos(self : Escena) : Diccionario
+getHeroe(self : Escena) : Heroe
+getMuestraFlechas(self : Escena) : boolean
+getNombre(self : Escena) : string
+getObjeto(self : Escena,nombre : string) : Objeto
+getPropiedad(self : Escena,objeto : Objeto,propiedad : string) : Val
+setBackground(self : Escena,backgroundName : string) : void
+setDescripcion(self : Escena,descripcion : string) : void
+setHeroe(self : Escena,nu_heroe : Heroe) : void
+setMuestraFlechas(self : Escena,muestraFlechas : boolean) : void
+setNombre(self : Escena,nombre : string) : void
+setPropiedad(self : Escena,objeto : Objeto,propiedad : string,parametros : string) : void
Da
C
B
+ TerminalProcessor
-dom : xmldom
+__init__(self : TerminalProcessor,filename : string)
+getTerminals(self : TerminalProcessor) : Diccionario[]
+ Nodo
-lexema : string
-token : string
-indice : int
+__init__(self : Nodo,tok : string,lex : string,ind : int)
+__str__(self : Nodo) : string
+getToken(self : Nodo) : string
+setToken(self : Nodo,tok : string) : void
+getLexema(self : Nodo) : string
+setLexema(self : Nodo,lex : string) : void
+getIndice(self : Nodo) : int
+setIndice(self : Nodo,ind : int) : void
+ LogicoABJ
-problema : Escena
-heroe : Heroe
-keywords : Diccionario
-sinonimos : Diccionario
-genero : Diccionario
-articulo : Diccionario
-gramatica : NLTKGrammar
-rdParser : NLTKParser
+__init__(self : LogicoABJ,terminales : string,nombre_heroe : string)
-matchRegEx(self : LogicoABJ,str : string) : string
-matchKw(self : LogicoABJ,str : string) : string
-nodificar(self : LogicoABJ,subArbol : Nodo,lexemas : list,siguiente_indice : int) : void
-comparaGenero(self : LogicoABJ,comparando1 : string,comparando2 : string) : boolean
-errorGenero(self : LogicoABJ,articulo : string,sustantivo : string) : string
-ejecucionSemantica(self : LogicoABJ,subArbol : Nodo,lexemas : list) : boolean
+runCommand(self : LogicoABJ,str_command : string) : void
A

Figura 27. Segunda parte del diagrama de clases propuesto para la solución de software del megaproyecto ABJ.

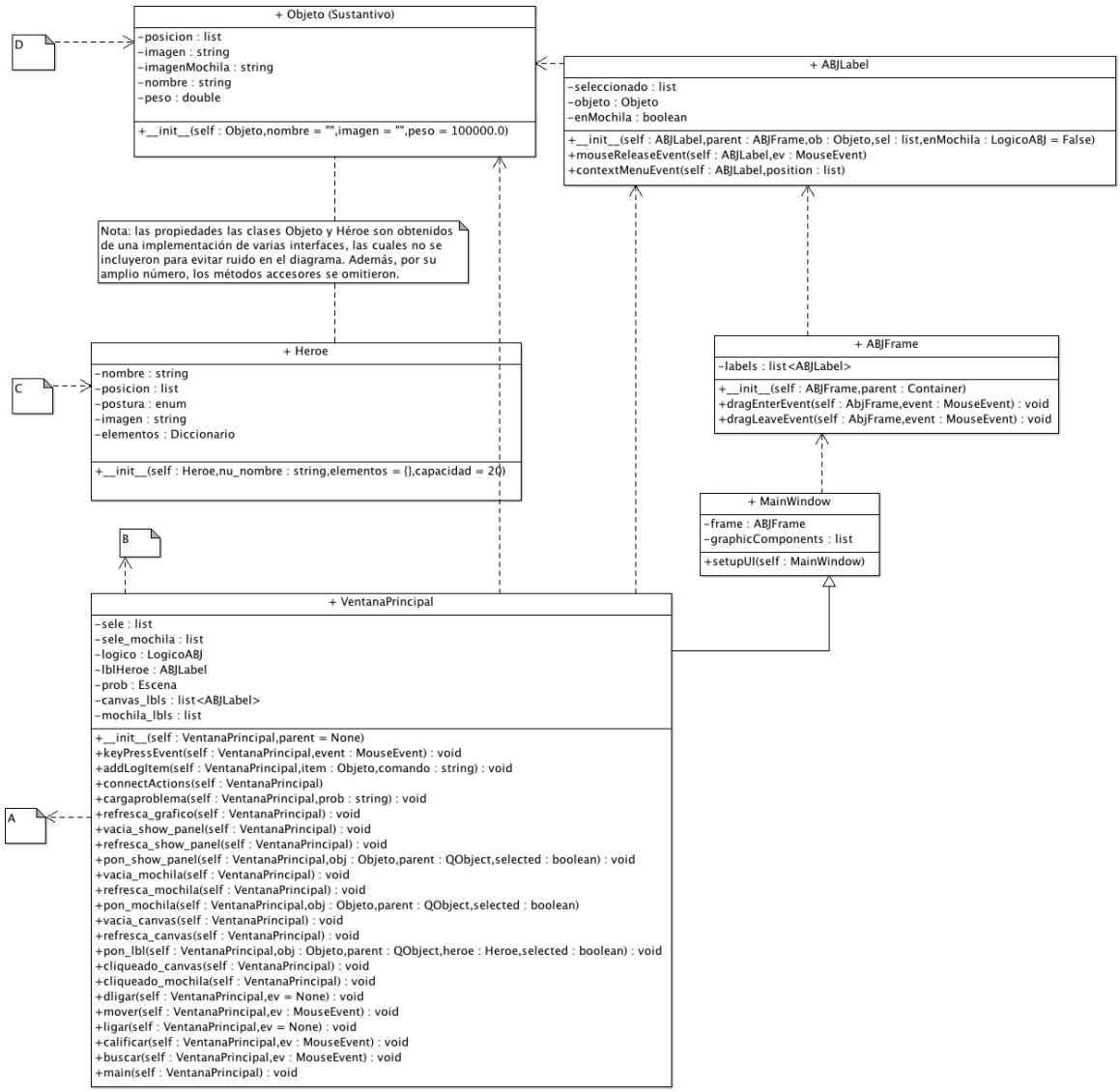
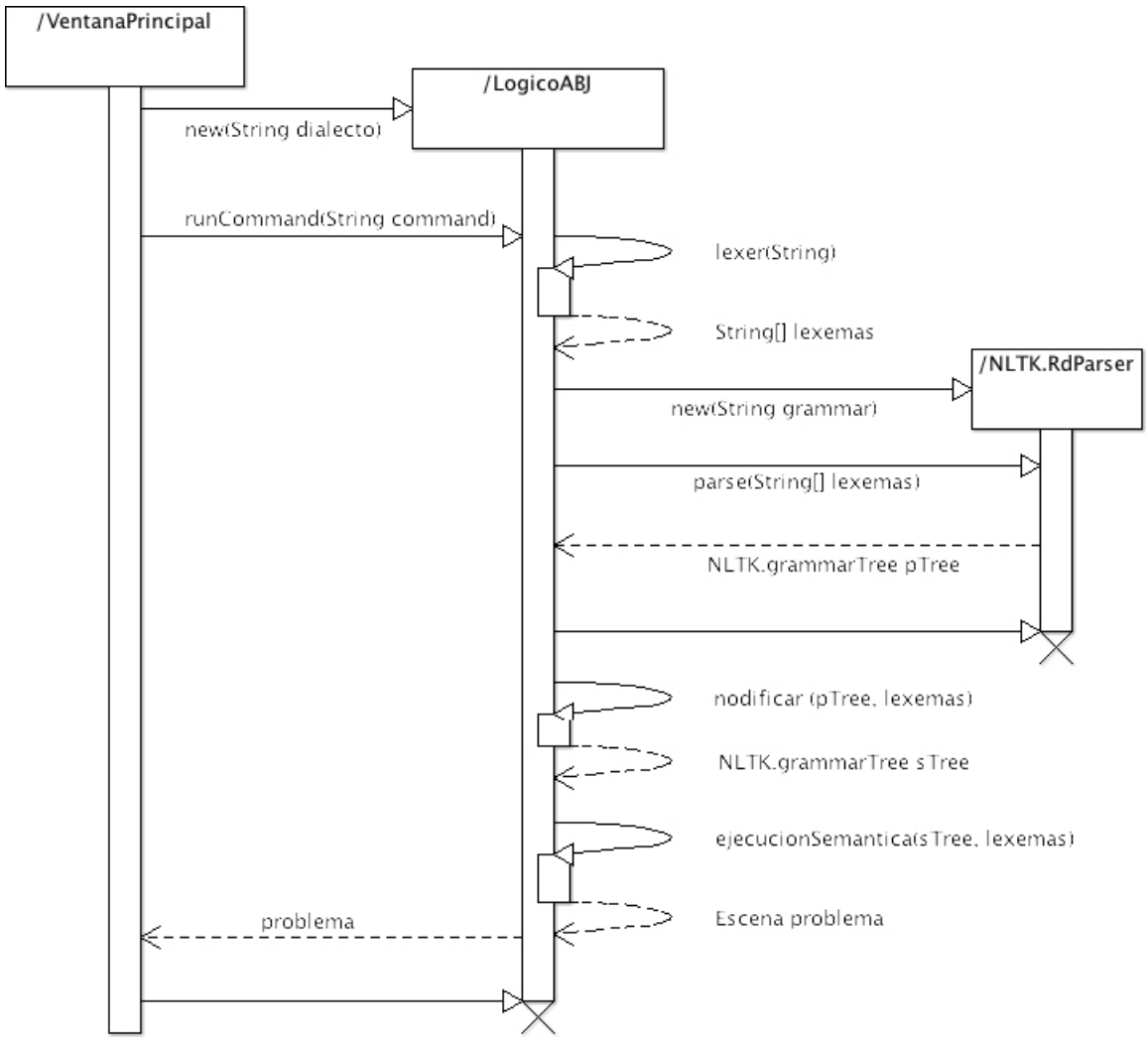


Figura 28. Diagrama de secuencia representando lo sucedido durante la ejecución de la instrucción runCommand.



D. Gramática.

El diseño de la gramática tomó en cuenta algunos experimentos realizados con el equipo de educación para determinar el formato general de los algoritmos que se producirían a partir de nuestra herramienta. Uno de dichos ejemplos se esboza en el Cuadro 10.

Con lo anterior en mente, dividimos los tipos de oraciones en las siguientes categorías:

1. Búsqueda. Estas acciones permiten al usuario interactuar con su entorno y encontrar elementos que probablemente en un principio no estaban a la vista. Algunos ejemplos:

- Buscar algo que brille
- Buscar el carro

- Localizar la licuadora

2. Ligadura. Estas acciones permiten al usuario obtener herramientas del escenario, para luego usarlas en el desarrollo de su solución al problema. Algunos ejemplos son:

- Recoger la pala
- Agarrar la pelota
- Sostener el carrito

3. Desligadura. Estas acciones permiten al usuario soltar los objetos que ha ligado, y ligarlos con otros objetos (sobre los cuales los deja). Algunos ejemplos son:

- Soltar la pala
- Dejar la pelota sobre la mesa

4. Movimiento. Estas acciones le dan la facultad al usuario de moverse en el escenario. Algunos ejemplos son:

- Ir hacia la puerta
- Ir hacia adelante
- Moverse hacia atrás tres pasos
- Correr hacia el carro

5. Uso. Toda acción que no se puede encasillar en alguna de las categorías anteriores forma parte de esta categoría (acciones específicas de los objetos con los que se interactúa). Por ejemplo:

- Encender el fósforo
- Abrir el grifo
- Cerrar la puerta

6. Otros.

a. Cambiar de postura. Como funcionalidad adicional para futuros problemas, se creó la interfaz 'IPosturaCambiable', la cual posee las acciones e información para consultas y cambios sobre la postura de un personaje. Dentro del alcance del proyecto los problemas propuestos no incluyen interacción del héroe con otros personajes, por lo que la interfaz fue aplicada únicamente al personaje principal. No se programó una correspondencia en la interfaz gráfica sobre los cambios de postura, mas sí se sentaron las restricciones pertinentes en cuanto a acciones realizables. Las posturas permitidas al momento para el héroe son: parado, sentado y acostado; y el héroe puede únicamente realizar acciones y desplazarse cuando está parado.

Cuadro 11. Experimento de algoritmo propuesto para resolver el problema "¿Cómo hacer limonada?".

Oración propuesta	Acción categorizada
1. Voy hacia la refrigeradora	1. Movimiento
2. Abro la refrigeradora	2. Búsqueda
3. Tomo el pichel	3. Ligadura
4. Voy hacia la mesa	4. Movimiento
5. Dejo el pichel en la mesa	5. Desligadura
6. Abro la alacena	6. Búsqueda
7. Tomo el toki	7. Ligadura
8. Tomo el azúcar	8. Ligadura
9. Voy hacia la mesa	9. Movimiento
10. Vierto el toki en el pichel	10. Uso
11. Vierto el azúcar en el pichel	11. Uso
12. Revuelvo el agua	12. Uso

E. Diseño del sitio de comunicación (Wiki).

Se empezó diseñando dos propuestas de diseños gráficos de la wiki del proyecto. Estos fueron sometidos a discusión en las reuniones de Megaproyecto para escoger el diseño que más agradara a todos los miembros del Megaproyecto. La diferencia entre ambas propuestas es que una era con colores llamativos y la otra con colores opacos.

Figura 29. Arte propuesta para el diseño gráfico de la Wiki.

Se utilizó el logo para extraer la paleta de colores a utilizar en el sitio. El objetivo principal del diseño era reflejar el motivo de las abejas, ya que el nombre coloquial del Megaproyecto es ABJ.

1. Implementación de la forma de contacto. Para facilitar el contacto con las personas que visitan nuestro sitio web de apoyo, se implementó una forma de contacto que envía un correo al grupo del Megaproyecto ABJ con los comentarios del interesado. El sitio se desarrolló a partir de una instalación de MediaWiki como su manejador de contenidos, y la forma de contacto se programó como una extensión programada en los lenguajes PHP y HTML, basada en la herramienta 'phpmailer'. PHPMailer es una clase hecha por Andy Prevost de CodeWorxTech, gratuita para descarga en su sitio http://sourceforge.net/projects/phpmailer/.

2. Estructura de categorías y usuarios. El diseño de la estructura en categorías del sitio se realizó buscando aprovechar esta característica particular de MediaWiki. Considerando la posibilidad de asignar diferentes categorías a un mismo artículo, se ideó la estructura ideal con base a los grupos de trabajo, tal y como se distribuyó el trabajo en la planificación del El Proyecto. De este modo, se podrían crear subcategorías interdisciplinarias para relacionar un

mismo tema con diferentes temas académicos, grupos, facultades y usuarios. A manera de ejemplo, se podría inventar una categoría llamada “Pruebas y experimentos”, cuyas subcategorías separarían el conjunto de pruebas realizadas (e.g., usabilidad, pruebas estandarizadas, etc.) y cuyas categorías padre se dividirían en las facultades involucradas, a saber: Facultad de Ingeniería, Facultad de Educación.

3. Implementación de editor WYSIWYG. Debido a que la plataforma MediaWiki utiliza su propio lenguaje (llamado ‘wikitext’) para la manipulación de artículos, se acordó incluir un editor de texto especializado en el sitio en lugar de impartir una capacitación sobre el wikitext. La razón principal fue la falta de intuitividad que el wikitext presenta para usuarios no relacionados con los aspectos básicos del desarrollo web, y la falta de relevancia que dicho aprendizaje tendría para los fines del Proyecto.

El editor implementado es una herramienta existente como extensión a la plataforma MediaWiki, llamada FCKEditor. Dicho editor se caracteriza por poseer las herramientas fundamentales que la mayoría de procesadores de texto poseen, facilitando la tarea de creación de tablas, alineamiento de texto e inclusión de enlaces e imágenes (por mencionar algunos ejemplos) por medio de herramientas visuales como botones y listas seleccionables. La extensión se encuentra gratuita para descarga en el siguiente enlace: http://www.mediawiki.org/wiki/Extension:FCKeditor_(Official).

F. Selección de Herramientas de implementación.

1. Herramientas para el procesamiento de lenguajes. Es de suma importancia la exploración de posibilidades y herramientas que ayuden al desarrollo exitoso del megaproyecto ABJ. Por ello, es necesario profundizar en el tema de lenguajes formales y las herramientas que actualmente se usan para acercarse a su procesamiento. Antes de mostrar las herramientas investigadas, se presenta a continuación una lista de características deseables:

a. Características deseables de la herramienta de procesamiento de lenguajes.

1) Flexibilidad en la definición de gramáticas. La idea del proyecto es que en un futuro se pueda extender para que reconozca gramáticas distintas al español (idiomas mayas, por ejemplo), por tanto, necesitamos que el análisis semántico no dependa enteramente de la gramática. Desde este punto descartamos herramientas que preprocesan gramáticas para generar una tabla de acción. Por ello preferimos generadores de analizadores sintácticos que permitan la definición de la gramática en tiempo de ejecución. Para los límites de este proyecto se decidió no

implementar un sistema de reglas semánticas dinámico. Es decir, la definición de un dialecto no soporta la definición de sus propias reglas semánticas.

2) Soporte para análisis semántico. Es necesario que el árbol sintáctico generado por una entrada provea métodos para recorrerlo e interpretarlo.

3) Requerimientos mínimos de hardware. Se requiere que la herramienta pueda correr en un ambiente con las mínimas capacidades de cómputo, debido a que se toma en cuenta la realidad de la mayoría de escuelas y colegios guatemaltecos, y así obtener una herramienta con un alcance e impacto mayor. A continuación se esbozan los mínimos requerimientos para correr el proyecto:

- Procesador Intel Pentium 3 o equivalente.
- Windows XP Service Pack 2 / Ubuntu Intrepid Ibex (8.10)
- 0.6 GB de espacio en disco duro.
- 128 MB de memoria RAM.

4) Multiplataforma. Es preferible que la herramienta corra tanto en ambientes Windows como Linux, debido a que esos son los dos sistemas operativos que se instalan normalmente en las computadoras de escuelas y colegios guatemaltecos.

b. Herramientas consideradas.

1) Gold Parser. Es un sistema de análisis que provee un editor y una interfaz gráfica para definir gramáticas libres de contexto a través de expresiones regulares y producciones. Entre sus características más atractivas están la detección de errores gramaticales, generación de código de escáner y parser en distintos lenguajes de programación (lo más destacado Java y Python), entre otros. Además la mayoría de integrantes del megaproyecto ABJ ya han tenido experiencia con Gold Parser, haciéndolo, hasta este punto, una propuesta interesante. Entre las desventajas principales se encuentra su poca flexibilidad debido a la tabla de acción precompilada, y la poca compatibilidad entre generadores de programas esqueleto y editores (DevinCook).

2) Yet Another Python Parser System (YAPPS). Es una librería de poco peso para Python que ofrece una interfaz simple para la creación de gramáticas libres de contexto. Genera los analizadores sintácticos y la estructura de análisis semántico (parser) basados en una

descripción de producciones en tiempo de ejecución. Tiene como desventaja que los analizadores que generan tienen muchas limitantes en cuanto a la gramática soportada, y el proceso de instalación de YAPPS es manual (Stanford).

3) NLTK. Es una enorme librería para Python que provee varios servicios para análisis de entradas basadas tanto en gramáticas libres de contexto como en una versión de la gramática del idioma Inglés. La herramienta provee las facilidades de generación de analizadores sintácticos y semánticos, y viene equipada de varias opciones de diagramación de resultados y análisis de significados según el contexto de aparición de un terminal. También tiene la ventaja que permite la definición de gramáticas en tiempo de ejecución, y el árbol sintáctico generado es bastante sencillo de utilizar. Entre sus desventajas se puede mencionar que su sistema de definición de expresiones regulares recae en el sistema de Python, su amplio peso, no tiene utilidades que lo integran con su propio sistema; también cabe decir que el proceso de instalación es manual (Natural Language Toolkit).

Dicho lo anterior se concluyó que la mejor herramienta a usar sería la librería NLTK (Natural LanguageToolkit) para Python. Esta herramienta corre en sistemas básicos con Python instalado y permite la generación de potentes analizadores basados en gramáticas definidas en tiempo de ejecución.

2. Herramientas para el despliegue gráfico. La implementación de la interfaz grafica fue realizada con la herramienta Diseñador QT la cual permitía por medio de un IDE sencillo hacer ventanas y manejar las acciones sobre estas.

Sobre el archivo generado por QT con las ventanas hechas se utilizo la herramienta PYQT para transformar el archivo de XML con las funciones de C++ a Python. Con esto se procedió a modificar el formato desde Python.

VI. RESULTADOS Y DISCUSIÓN

A. Resultados cuantitativos.

Al inicio del Megaproyecto se tomó la decisión de aplicar la prueba Otis Lennon para determinar un cambio en la adquisición de nuevas habilidades como resultado del uso de algoritmos.

Los resultados obtenidos en la evaluación cuantitativa mostraron que la elección de la prueba Otis Lennon, como instrumento para medir la adquisición de nuevas habilidades en los alumnos, no fue la adecuada.

Expertos consultados indicaron que:

- El test Otis Lennon mide la Inteligencia G, que como constructo general es bastante estable, y por lo tanto, no se esperaría que se dieran variaciones significativas.

- Tres meses es un tiempo muy corto para esperar que una intervención tenga un resultado significativo que pueda ser medible, especialmente en el desarrollo de habilidades y destrezas.

B. Resultados cualitativos.

Los resultados y observaciones que se detallan en esta sección son el resultado del acompañamiento semanal que se hizo a los docentes de los grupos experimentales de las cinco instituciones educativas, así como de visitas hechas al aula. Para ello, se utilizó el formulario que se encuentra anexo al presente documento. Esta sección incluye también los comentarios que hicieron los docentes durante una reunión conjunta que se convocó al finalizar el período de tres meses. En dicha reunión se llevó a cabo un intercambio de experiencias y recomendaciones, y se aprovechó para agradecer y entregar reconocimientos a los docentes colaboradores.

En cuanto a la capacitación y al manual del docente se puede afirmar que, en general, en las visitas semanales realizadas a los docentes participantes se hizo evidente que la capacitación individual no surtió el efecto esperado, estos tenían dudas sobre lo que era un algoritmo y sobre la metodología del **Modelo ABJ**, no habían comprendido completamente lo que la herramienta era y menos aún, lo que con ésta se podía lograr. Al respecto se recibió el siguiente comentario:

«Le agradezco que haya venido a visitar mi clase porque uno de maestro no sabe si lo está haciendo bien.» (Docente de grupo experimental).

Además, en el manual se solicita que los algoritmos se formulen en modo indicativo. En el intercambio realizado en la reunión de clausura, algunos de los docentes coincidieron en la necesidad de explicar el concepto, y lo expresaron de la siguiente manera:

«Al principio no entendí qué era eso del 'modo indicativo', lo deberían explicar en el manual, yo no soy maestra de idioma y eso no lo tengo muy fresco…» (Docente de grupo experimental)

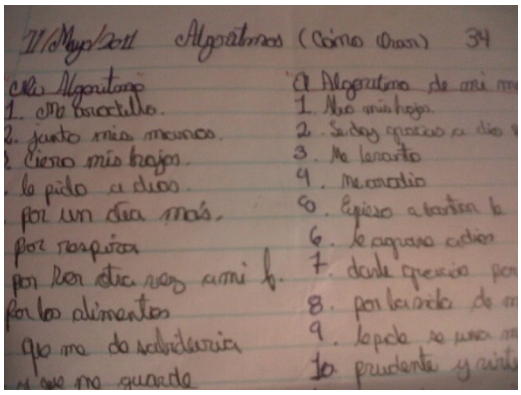
Sin embargo, la herramienta en sí tuvo muy buena aceptación entre los docentes y el alumnado de los grupos experimentales. Esto se hizo más evidente cuando los docentes comentaron que muchos alumnos también aplicaron los algoritmos con miembros de sus familias, de su propia iniciativa. Un docente expresó:

«…los niños disfrutan mucho el tiempo que dedicamos a los algoritmos, especialmente porque sienten que están jugando. Cuando no lo hacemos, los niños me lo recuerdan e insisten en hacerlo…» (Docente)

Figura 30. Muñecos de tusa.

Fuente: Margarita Suhr M.

Figura 31. Algoritmo con dos procesos distintos.



Fuente: Margarita Suhr M.

Para la comprensión del concepto en sí, uno de ellos expresó que inició la aplicación de la herramienta solicitando que "hicieran una sandalia de foamy" y que luego elaboraran el algoritmo de la actividad. Esto le ayudó a que sus alumnos comprendieran lo que es un algoritmo. Los demás coincidieron con la idea, expresando que la aplicación de la herramienta sería más funcional si se divide en dos fases; una para la comprensión de lo que es un algoritmo, y la segunda, su aplicación.

Adicionalmente, los docentes comentaron que los algoritmos han contribuido a su crecimiento personal y profesional, pues ahora son más ordenados y aplican lo aprendido en su vida diaria. Todos los docentes que participaron manifestaron su intención de continuar el uso de algoritmos en sus aulas, ya que observaron cambios positivos en los estudiantes:

«La palabra algoritmo es ya común en mi aula.»(Docente de grupo experimental).

«También lo estoy usando en mis clases de refuerzo y tutorías. Me está funcionando bien.»(Docente de grupo experimental)

Los docentes indicaron que uno de los cambios más evidentes que observaron fue en los hábitos personales y de trabajo de los estudiantes. Una docente explicó que el aula luce ahora más ordenada, ya no hay libros tirados ni mochilas regadas. Así mismo, algunas de las personas que aplicaron el postest indicaron que les fue más fácil organizar la sección del grupo experimental que la del control. Además, en una institución, los primeros terminaron antes, y tuvieron oportunidad de chequear sus respuestas. Así mismo, los docentes hicieron la misma observación,

que, en general, los alumnos de los grupos experimentales se tornaron más ordenados e independientes, lo que confirma la impresión de los que aplicaron el pre y postest. Con relación a esto uno de los docentes afirmó:

«Esta sección era especialmente desordenada… libros tirados por aquí y allá, mochilas, loncheras, papeles. He notado que ahora son más ordenados. El señor que vino a hacer la evaluación me comentó cuando se fue que le había sido más fácil poner en orden a los alumnos de esta sección (experimental) que la otra (control) antes del examen. » (Docente de grupo experimental)

Otro de los efectos positivos que tuvo la herramienta en uno de los grupos experimentales fue observado en una competencia de matemáticas que se llevó a cabo dentro de la institución. La sección experimental fue la ganadora sobre las otras tres secciones participantes. Según informó el coordinador del área de matemáticas del plantel, se esperaba que ganara otra sección, pues era la que tenía un mejor rendimiento en esa materia. Por esa razón, la directora de esta institución tomó la decisión de incorporar la herramienta en todos los niveles y materias. Los docentes sugirieron que los alumnos de los primeros grados pueden aprender la herramienta de forma verbal:

«Le estamos pidiendo a la miss que capacite a todos los docentes de la institución. Yo también estoy asistiendo a las capacitaciones y hace poco le pude ayudar a mi hija a comprender un concepto usando un proceso de algoritmos No pude hacer el algoritmo del avioncito de papel.»(Coordinador área de matemática de institución educativa)

Los maestros trasladaron la utilización de la herramienta en las áreas curriculares de su práctica docente. Observaron que muchos alumnos comprendían mejor un proceso matemático cuando lo convertían en algoritmos. Los alumnos de una de las secciones experimentales se habituaron a elaborar un algoritmo después de aprender un nuevo tema. Un docente comentó que en su aula había empezado a utilizar los algoritmos para evaluar el aprendizaje o medir la comprensión de los alumnos sobre un tema:

«…ahora, después de enseñar un tema, les pido a mis chicos que elaboren un algoritmo del tema. También lo estoy utilizando en mis exámenes. Incluyo preguntas pidiendo que hagan un algoritmo de algún problema. »

Los docentes también indicaron que varios alumnos habían mejorado su rendimiento en varias áreas curriculares donde habían aplicado los algoritmos. Observaron también que un número de ellos desarrolló un orden lógico en sus acciones.

Los docentes comentaron que la parte que más les gusta a los alumnos de la enseñanza y práctica de los algoritmos es la dramatización (o mímica) que hacen. Observaron que esta forma fue más efectiva para lograr que los alumnos comprendieran mejor que un algoritmo puede ser distinto, sin que eso afecte el resultado final del ejercicio. Esto fue más efectivo que volver a escribir otra forma de cómo lo haría, como se pedía en el cuadernillo. Esto último los aburría y muchos de ellos se resistían. Sin embargo, dramatizarlo les gustó mucho. Todos los niños participaban, estaban atentos a señalar los errores que se cometían y ellos mismos buscaban otras formas de hacerlo. Se divertían y fue una buena forma para comprobar si lo hicieron bien:

«…mire, eso de aprender sobre algoritmos pareciera sencillo, así creí yo al principio, pero en esta primera semana no he logrado que lo hagan bien. El concepto ya lo entendieron, pero cuando se ponen a hacerlo, no incluyen todos los pasos. Lo vimos bien cuando lo trataron de dramatizar. » (Docente grupo experimental).

También comentaron que los alumnos trabajaron mejor los algoritmos cuando estos estaban relacionados con su contexto: algún tema que estuviera viendo en ese momento, alguna actividad que se estuviera realizando dentro de la institución o medio donde viven: la celebración del día de la madre, por ejemplo

Un docente hizo la observación que la escritura de los alumnos de ese nivel es aún muy grande, y que el espacio proporcionado para escribir los algoritmos en el cuadernillo es muy pequeño. Sugirió hacer más grande los espacios donde escriben los niños:

«La letra de los niños es grande y no les cabía bien porque el espacio era muy pequeño. »(Docente de grupo experimental)

Por lo tanto, para el mejoramiento del cuadernillo, se hizo lo siguiente:

- énfasis en la dramatización (mímica) del algoritmo, como una forma de que el alumno identifique similitudes y diferencias, encuentre errores y los corrija y analice otras opciones.

- proporcionó un espacio en el cuadernillo para que el alumno elabore una lista de lo algoritmos que les gustaría hacer. Esto también apoya el trabajo del docente.

- motivó al alumno a utilizar los algoritmos dentro de las distintas áreas curriculares
- eliminó el ejercicio de construir un avión de papel, pues se observó que les daba mucha dificultad y les causaba frustración.

Y, para el mejoramiento del manual del docente y la elaboración del instructivo de capacitación docente se puede deducir que es necesario que el capacitador pueda transmitir (esté convencido) de lo que la herramienta es capaz de desarrollar en los alumnos cuando ésta se aplica frecuentemente (tres o más veces por semana). Además, los docentes deben ser capacitados en grupos de cuatro o más, para que se pueda realizar la práctica de la aplicación de los algoritmos con el "comparar-analizar- decidir" como se detalla en el manual del docente para que experimenten los beneficios del proceso, lo que permitirá hacer mayor énfasis en la importancia de dicho ejercicio.

Para los docentes que no se haya podido capacitar en grupo, es necesario acompañarles a lo largo de algunas semanas (según sea necesario) al principio de la aplicación de la herramienta, para orientarlo en la práctica.

Otro aspecto importante es la dramatización o mímica de los algoritmos fue una de las prácticas que más benefició la aplicación de la herramienta. Los alumnos no sólo la disfrutaban, sino que fue apta para practicar el comparar (estaban prestos a las similitudes y diferencias de sus algoritmos con relación a las de los de sus compañeros), analizar y decidir qué pasos eran más prácticos o adecuados.

De la retroalimentación de los docentes se concluye que la comprensión y aplicación de la herramienta sería más fácil si se hace en dos fases: 1) comprender qué es un algoritmo; 2) poner en práctica la elaboración de algoritmos en la resolución de problemas. Además, para los docentes de áreas específicas, como la matemática, es necesario explicar en el manual del docente qué es el modo indicativo.

Por otro lado, las actividades a realizar deben sean propias del contexto de los educandos, sea a nivel de la institución educativa o familiar. Por ser una herramienta que debe utilizarse continuamente, los docentes, conociendo el contexto de sus alumnos, deben plantear otras actividades y/o invitar a los alumnos a que ellos las propongan.

A pesar que en todo proceso de enseñanza aprendizaje el centro de atención debe ser el alumno, es necesario enfatizar la importancia que tiene que el docente esté motivado y deseoso

de implementar esta nueva práctica en el aula, para que ese entusiasmo sea trasladado a los alumnos. Para ello, es indispensable el seguimiento y apoyo del docente, por lo que se debe insistir en las visitas periódicas dentro y fuera del aula. Las visitas que el evaluador hace al aula tienen un efecto bidireccional: el evaluador es observador directo del efecto de la herramienta en los alumnos y forma sus propias opiniones, eliminando el grado de subjetividad de las observaciones del docente. Al mismo tiempo, envía al docente un mensaje de que valora y aprecia lo que está haciendo.

Los resultados que se observaron en sólo tres meses de aplicación de la herramienta, así como el deseo e intención de los docentes que participaron de continuar la aplicación de la herramienta pueden ser el fundamento que dé origen a un nuevo megaproyecto que busque proyectarse a las comunidades rurales e indígenas y/o en grupos de estudiantes del primer ciclo de primaria. También puede dar origen a un estudio longitudinal que compruebe, con resultados tanto cualitativos como cuantitativos, la efectividad del uso de algoritmos en el desarrollo de habilidades cognitivas en los alumnos. Se espera también, interesar a autoridades del Ministerio de Educación, para que incorpore esta herramienta (el cuadernillo, el manual del docente, el instructivo de capacitación docente y la herramienta computarizada) en el Currículum Nacional Base, empezando desde en el primer ciclo de primaria.

C. Entregables de software.

El presente apartado expone los resultados obtenidos tanto en el área de desarrollo de software y afines.

1. Gramática del lenguaje ABJ. Basado en el diseño de la gramática, y después de realizar varios experimentos con el equipo de computación y con el equipo de educación, se determinó que las siguientes producciones compondrían la gramática del lenguaje ABJ. Antes de proceder con su descripción, es necesario decir que la notación *terms ['categoría']* es utilizada para denotar el conjunto de terminales leídos del dialecto, pertenecientes a la categoría 'categoría'. Para una documentación completa de los significados de los terminales y de la implementación completa de la gramática del lenguaje ABJ, referirse al apéndice:

a. Definición de un comando ABJ. Un comando es (en orden de aparición según la siguiente producción) cargar un problema usando un identificador, cargar un problema usando un sustantivo, o un paso de un algoritmo:

- P → PLA id | PLA SUS | PAS

b. Cargar un problema. Representa un comando especial utilizado para cargar un problema a la memoria.

- PLA →terms['vPlantear']

c. Sustantivos.

- SUS →terms['sus']

d. Paso en la solución de un problema. Un paso de un algoritmo se define de forma genérica como sigue:

- PAS → SCAN | LIG | DLIG | MOV | HACE | COND_P | CAMB_POSTURA

A continuación se detallan cada uno de los tipos de pasos:

1) SCAN (ligadura).

- SCAN → V_SCAN terms['iCaller'] V_PRES_SUBJ | V_SCAN ART SUS | V_SCAN PREP ART SUS
- V_SCAN →terms['vScan']
- V_PRES_SUBJ →terms['vPresSubj']
- ART →terms['arts']
- SUS →terms['sus']
- PREP →terms['preps']

2) LIG (búsqueda).

- LIG → V_LIG ART SUS | V_LIG ART SUS terms[‘sPos’] ART SUS | V_LIG ART SUS terms[‘scPos’] SUS
- V_LIG →terms['vLigar']

3) DLIG (desligadura).

- DLIG → V_DLIG ART SUS | V_DLIG ART SUS PREP ART SUS

- V_DLIG →terms['vDligar']

4) MOV (movimiento).

- MOV → V_MOV OBJETIVO
- V_MOV →terms['vMov']
- OBJETIVO → ART SUS | ART DIR LONG
- DIR →terms['dirs']
- LONG →terms['longs']

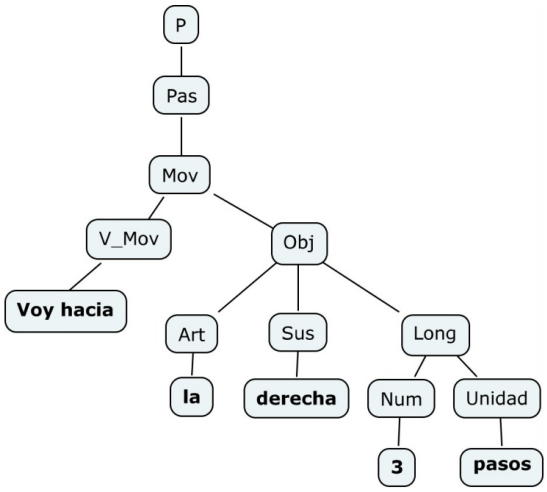
5) HACE (uso).

- HACE → VERBO ART SUS | VERBO ART SUS REP| VERBO
- VERBO →terms['hacer']
- REP →terms['reps']

6) CAMB_POSTURA (cambio de postura).

- CAMB_POSTURA → V_CAMB_POSTURA PREP ART SUS | V_CAMB_POSTURA
- V_CAMB_POSTURA →terms['vCambPostura']

Figura 32. Ejemplo de árbol sintáctico generado dada la oración d: “voy hacia la derecha 3 pasos”



Desde un principio se definió la gramática pensando en que sería lo suficientemente flexible para que su misma arquitectura, en un futuro, soportara la incorporación de nuevos idiomas y dialectos al sistema. Así pues, se observó que una manera bastante eficiente de realizar dicha tarea es construyendo la gramática en tiempo de ejecución. Es por ello que introducimos la notación *terms*, que tiene un significado en código literal, para la construcción de la gramática. Aún así, la definición de la gramática carece de una flexibilidad especialmente útil al momento de definir nuevos idiomas (no dialectos). La misma es la carencia de un mecanismo de definición de producciones, o reestructuración de las mismas.

De la gramática se decidió dejar fuera (en términos del alcance del proyecto) la definición de estructuras condicionales (por ejemplo: si, entonces) y estructuras cíclicas (por ejemplo: mientras, haz) debido a la amplia carga semántica que implica la definición de la cláusula de condición que utilizan ambas estructuras como parámetro. Por ejemplo, nos encontramos con problemas para la definición de mecanismos de acceso a propiedades en términos de sustantivos, y la especificación de los mismos.

2. Intérprete. El analizador semántico juega un papel crucial para el desarrollo de problemas, ya que permite la inclusión de nuevo vocabulario en la gramática del Proyecto sin necesidad de editar el código fuente. Es importante mencionar que la trascendencia del analizador semántico depende tanto de la gramática como del archivo diccionario. El núcleo del analizador semántico se encuentra en la función getTerminals, perteneciente a la clase terminalProcessor. Por su parte, las interfaces IPosturaCambiable e ITener se incluyeron en el conjunto de acciones utilizadas para

los problemas de prueba. IPosturaCambiable se implementó exclusivamente en el héroe, aunque no fue utilizada en ninguno de los problemas ejemplares. ITener, en cambio, se aplicó a los contenedores Despensa, Refrigerador, Mesa, Gaveta y Mochila. Obsérvese como los contenedores incluyen también superficies de reposo, en cuyo caso son los denominados "contenedores abiertos", por tener a la vista los elementos que contiene. El diccionario dinámico es una de las piezas claves del proyecto, este complementa la definición del lenguaje ABJ.

La creación de nuevos ejercicios manualmente es bastante simple para personas con un nivel de programación medio, gracias a estructuración de los componentes del ejercicio. El problema radica en la creación de nuevas acciones que se pueden realizar sobre los objetos, por ejemplo: abrir, cerrar, encender, apagar, etc. Esto porque el nivel de programación para crear nuevas acciones depende de la complejidad de éstas; por lo tanto, en un futuro se debe trabajar para aumentar el nivel de abstracción para la creación de nuevas acciones y objetos; y crear una interfaz de usuario que lo haga, de forma semiautomática.

La división por módulos (acciones, objetos y escenas) de un ejercicio fue una decisión crucial, porque esto permitió que un cambio en objeto, acción o escena no perjudique a los ejercicios ya creados.

3. Interfaz gráfica. La maquetación de la interfaz gráfica es el resultado principal de esta fase. Para ello se tomaron las siguientes decisiones:

a. PYQT. Se eligió PYQT para la implementación de la interfaz gráfica por las siguientes razones:

1) Soporta OpenGL. OpenGL es una librería que permite la creación de modelos y escenas, esto abre la posibilidad de crear animaciones con los gráficos ya existentes en un futuro. Incluso deja abierta la posibilidad de integrar gráficos en tres dimensiones al ambiente.

2) Genera XML y código Python.XML es importante por la portabilidad que provee a la información, es fácil poder luego exportar esta información a otro tipo de dispositivos, como PDAs. La utilidad del código Python permite una integración transparente con NLTK.

3) Permite utilizar Diseñador QT (*QT Designer*).Diseñador QT es una IDE que permite diseñar interfaces en QT fácilmente, por medio de "*drag'ndrop"* de elementos, de esta forma se ahorra código y tiempo.

4) **Multiplataforma**. Dado que PYQT está implementado en Python, esto lo convierte en un ambiente multiplataforma, por lo cual es independiente del sistema operativo, siempre y cuando tenga instalada Python.

f. Posición de la bitácora de acciones. Se tomó la decisión de que la bitácora de acciones no estuviese visible todo el tiempo, como lo está la mochila. Tiene como ventajas la libertad en el área de trabajo, y evitar confundir a los niños al momento de mover la ventana a los extremos ya que la bitácora puede encajarse automáticamente en el escenario. Sin embargo, haberla escondido, disminuye la accesibilidad de la misma, siendo necesario buscar la opción para desplegarla desde el menú principal.

Puede que sea necesario asignar un botón para hacer visible la bitácora de acciones como se realizó con la mochila. De esta forma es más intuitiva su ubicación.

Figura 33. Maquetación final de la interfaz gráfica.

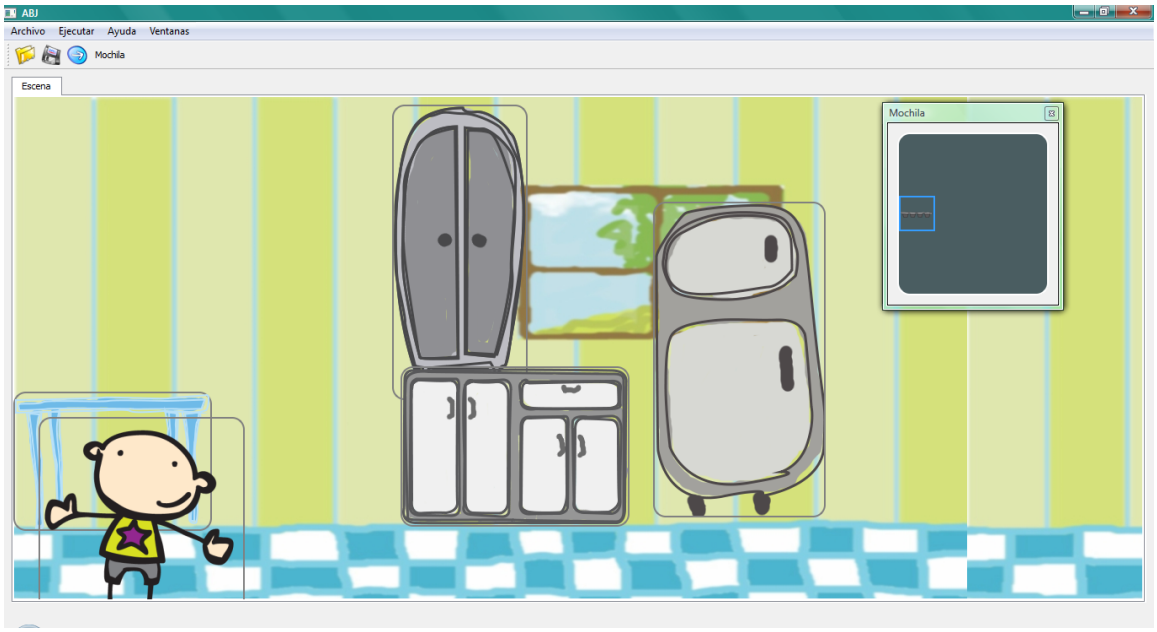
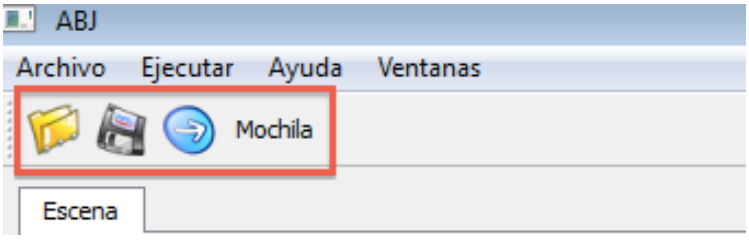


Figura 34. Botones



g. Botones de la barra superior. Se colocaron a disposición los botones que permiten un fácil y rápido acceso a las siguientes funciones:

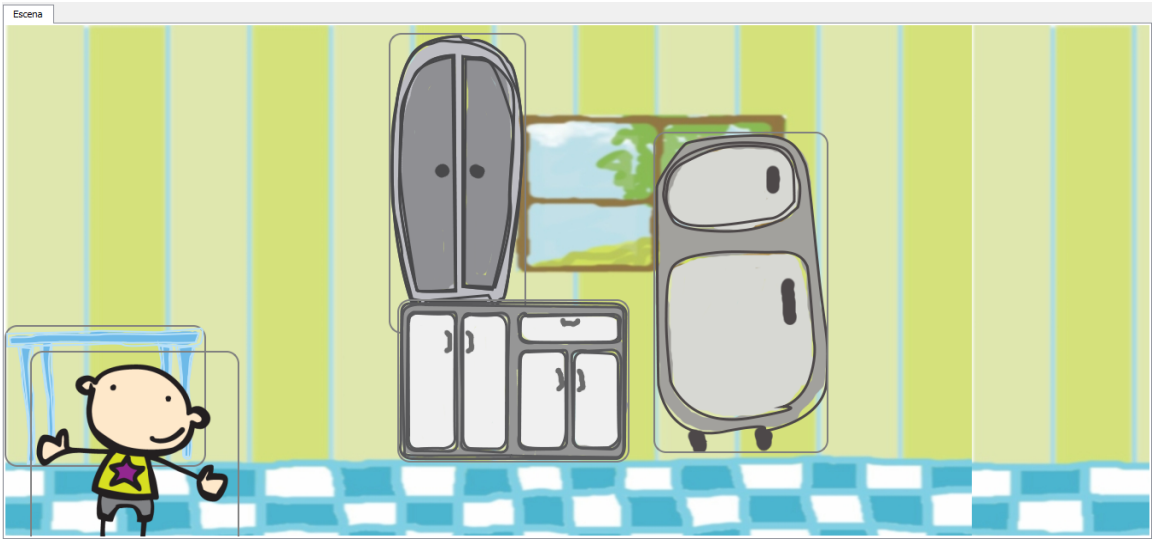
- Guardar
- Salir
- Comprobar el problema
- Mostrar la mochila

Las pruebas de usabilidad revelaron que a los niños les gusta que los botones tengan imágenes caricaturescas para identificar las funciones. Además, la barra superior es un área donde este tipo de funciones se encuentran comúnmente.

h. Menús superiores. Para evitar la saturación de la interfaz, las funciones planificadas para uso poco frecuentes se colocaron en menús. También se hizo una copia en los menús de todas las funciones representadas por botones, tomando en cuenta que las pruebas de usabilidad mostraron preferencia por los menús en algunos de los sujetos.

i. Ventanas emergentes. Son necesarias para retroalimentación al usuario sobre estado actual de la herramienta en momentos determinados; e.g., si ya ha terminado de resolver el problema, qué problema debe resolver.

Figura 35. Escenario.



j. Escenario. El escenario ocupa el área principal de la herramienta.

Para desarrollar la interfaz se decidió utilizar Diseñador QT (*QT Designer*, en inglés) debido a su capacidad de manejo de ventanas y *drag and drop*. Las principales razones por las que se trabajó con QT fueron las siguientes: QT provee documentación extensa que facilita la programación. El archivo generado por éste en lenguaje C++ puede ser fácilmente traducido a Python con la herramienta PyQT.

Diseñador QT permite manejar de forma dinámica las mallas lo cual facilitó el orden y la disposición de los elementos a través de los cambios de tamaño de ventana. También brinda la opción de modificar cada una de las propiedades de las ventanas y los widgets, lo cual facilita el manejo de las ventanas durante el diseño y la codificación.

Con base a las pruebas de usabilidad realizadas y a la teoría de Interacción Humano-Computador, es importante darle al usuario varias formas de realizar las acciones, donde cada forma de acceder a las acciones está orientada a un tipo de usuario. Por ejemplo, el usuario experto puede ser más ágil en el uso de atajos con el teclado, mientras que un usuario poco afín a las computadoras prefiere usar menús contextuales a través del mouse.

La función del marco gris que rodea las imágenes es mostrar a los usuarios los límites de la imagen, delimitando el área correspondiente a cada objeto que reaccionará ante los eventos del botón derecho o izquierdo del ratón.

Finalmente, se escogió Mediawiki como plataforma para el sistema de comunicación ya que es el manejador de contenidos colaborativo con mayor trayectoria y prestigio en el mercado. Además de ser gratuito, la documentación es extensa y hay disposición de un gran número de extensiones útiles para el Megaproyecto, e.g., la extensión que permite crear PDFs con el contenido del artículo en el wiki.

4. Software de comunicación (Wiki). Como resultado final del software de comunicación, se obtuvo un sitio que comprendía el historial de las actividades realizadas, la lista de entregables y las actas de las reuniones que se sostuvieron durante la realización del Megaproyecto. Lo más importante, es que el historial del Wiki nos permite ver el aporte de cada persona y poder regresar a versiones previas si esto fuera necesario.

Figura 36. Resultado de los módulos de diseño e implementación de la wiki del proyecto.

Una vez finalizada la fase de desarrollo de la herramienta computarizada, se trabajo en otro sitio en un ambiente wiki para poder utilizarlo como sistema de comunicación con el mundo. En él, cualquier visitante puede descargar los cuadernillos de trabajo para los niños, los manuales para el docente, el instructivo para el docente, así como el código de la herramienta computarizada. Por el momento, el código no está apegado a ninguna licencia, por lo cual se deben tomar medidas de precaución para evitar plagio. Adicionalmente, el visitante al sitio puede enterarse de qué se trata el Megaproyecto, visitar nuestro perfiles en las

redes sociales y visitar el wiki en donde se encuentra la documentación del proceso investigativo del Megaproyecto.

Figura 37. Wiki promocional.

Finalmente, se escogió Mediawiki como plataforma para el sistema de comunicación ya que es el manejador de contenidos colaborativo con mayor trayectoria y prestigio en el mercado. Además de ser gratuito, la documentación es extensa y hay disposición de un gran número de extensiones útiles para el Megaproyecto, e.g., la extensión que permite crear PDFs con el contenido del artículo en el wiki.

5. Documentación técnica. La documentación técnica del proyecto es un conjunto de páginas Web con la descripción completa de lo que cada variable, método y clase hace dentro de nuestro proyecto. Esta herramienta es de especial utilidad para consulta, sobre todo para individuos que deseen desarrollar sobre nuestra plataforma.

Figura 38. Página ejemplo sobre la documentación técnica generada para la clase ABJLabel.

Trees Indices Help

Module ABJLabel :: Class ABJLabel

[hide private]
[frames] | no frames]

Class ABJLabel

source code

```
      object --+
               |
??.simplewrapper --+
                   |
          ??.wrapper --+
                       |
    PyQt4.QtCore.QObject --+
                           |
          object --+       |
                   |       |
    ??.simplewrapper --+   |
                       |   |
PyQt4.QtGui.QPaintDevice --+
                           |
         PyQt4.QtGui.QWidget --+
                               |
              PyQt4.QtGui.QFrame --+
                                   |
                 PyQt4.QtGui.QLabel --+
                                      |
                                   ABJLabel
```

Esta clase representa una de las uniones entre el emisferio logico y el emisferio grafico, especificamente la referida a los objetos logicos (cuchara, carro, etc) y los objetos graficos (imagenes desplegadas)

Since: 4/14/2011

Version: 1.0

Nested Classes

[hide private]

Inherited from PyQt4.QtGui.QFrame: Shadow, Shape, StyleMask
Inherited from PyQt4.QtGui.QWidget: RenderFlag, RenderFlags
Inherited from PyQt4.QtGui.QPaintDevice: PaintDeviceMetric

Instance Methods

[hide private]

	`__init__(self, parent, ob, sel, logico, grafico=None, enMochila=False)` Constructor	source code
	`mouseReleaseEvent(self, ev)` Overload del manejador de evento de clic (release)	source code
	`contextMenuEvent(self, position)` Overload del manejador de evento de clic (release)	source code

D. Resultados logísticos y administrativos.

1. Presupuesto de gastos.

Cuadro 12. Presupuesto de gastos

Presupuesto Megaproyecto ABJ	
Gastos varios	Q 500.00
Gastos publicidad	Q 500.00
Gastos de Artículos de oficina	Q 200.00
Gastos por servidor	Q. 1,540.00
Pre y Post test	Q 6,000.00
Total	**Q 9,825.00**

2. Gantt de as actividades.

Figura 39. Actividades fase 1

Task	Status	Est. Time	Duration	Start Date	Due Date
Investigación sobre los lenguajes formales y su aplicación en la educación	Completed		21 days	30 Aug 2010	20 Sep 2010
Realizar el documento formal de descripción del proyecto	Completed		21 days	30 Aug 2010	20 Sep 2010
Aprendizaje en los niños	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Creación de matriz de marco lógico	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Estudio de Benchmark	Completed	3 hours	21 days	06 Sep 2010	27 Sep 2010
Incentivar instituciones para integración del proyecto	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Investigación tecnologías para las cuales se podría desarrollar	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Presupuestos	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Proveer al grupo de información sobre la educación en Guatemala que sea relevante para el proyecto	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Proveer de información teórica sobre aprendizaje en niños de 10 a 12 años	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Sitio del proyecto, Integrarlo con el logo y la paleta de colores	Completed	2 hours	14 days	06 Sep 2010	20 Sep 2010
Técnicas para enseñar con computadora	Completed	3 hours	14 days	06 Sep 2010	20 Sep 2010
Usabilidad en software para niños	Completed	2 hours	4 days	06 Sep 2010	10 Sep 2010
Presentación de resultados FASE 1	Completed	0.5 hours	3 days	28 Sep 2010	01 Oct 2010

Figura 40. Gantt Actividades fase 1

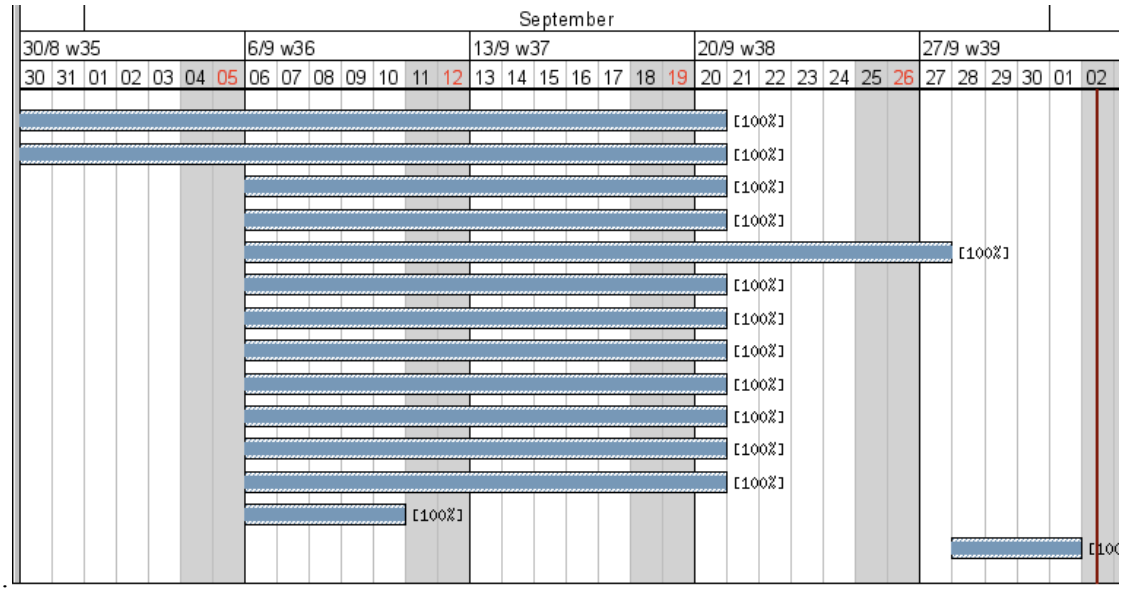


Figura 41. Actividades fase 2

Task	Status	Est. Time	Duration	Start Date	Due Date
Segundo Articulo	Open			17 Dec 2010	17 Dec 2010
Prototipo inicial en papel	Open	14 hours	14 days	15 Oct 2010	29 Oct 2010
Definición de problemas	Open	21 hours	20 days	16 Oct 2010	05 Nov 2010
Gramaticas en español	Done?	7 hours	4 days	18 Oct 2010	22 Oct 2010
Herramientas para programar en lenguaje natural	Done?	7 hours	4 days	18 Oct 2010	22 Oct 2010
Gramática (Producciones, terminales, etc)	Open	7 hours	13 days	23 Oct 2010	05 Nov 2010
Elementos de lenguaje	Done?	14 hours	6 days	23 Oct 2010	29 Oct 2010
Lenguaje de Programación para construir nuestro lenguaje	Done?	7 hours	6 days	23 Oct 2010	29 Oct 2010
Usability testing (papel)	Done?	7 hours	6 days	30 Oct 2010	05 Nov 2010
Diseño de tabla de símbolos, tipos estados, analizador semántico, etc.	Open	7 hours	6 days	06 Nov 2010	12 Nov 2010
Elaboración manual del docente	Open	21 hours	20 days	06 Nov 2010	26 Nov 2010
Modificaciones necesarias IDE	Open	7 hours	6 days	06 Nov 2010	12 Nov 2010
Diseño de ejemplos a manera de entregables, juego de mesa.	Open	14 hours	6 days	13 Nov 2010	19 Nov 2010
Testing: Diseño de ejemplos a manera de entregables juego de mesa.	Done?	7 hours	6 days	13 Nov 2010	19 Nov 2010
Traducción lenguaje	Open	7 hours	6 days	20 Nov 2010	26 Nov 2010

Figura 42. Gantt Actividades fase 2

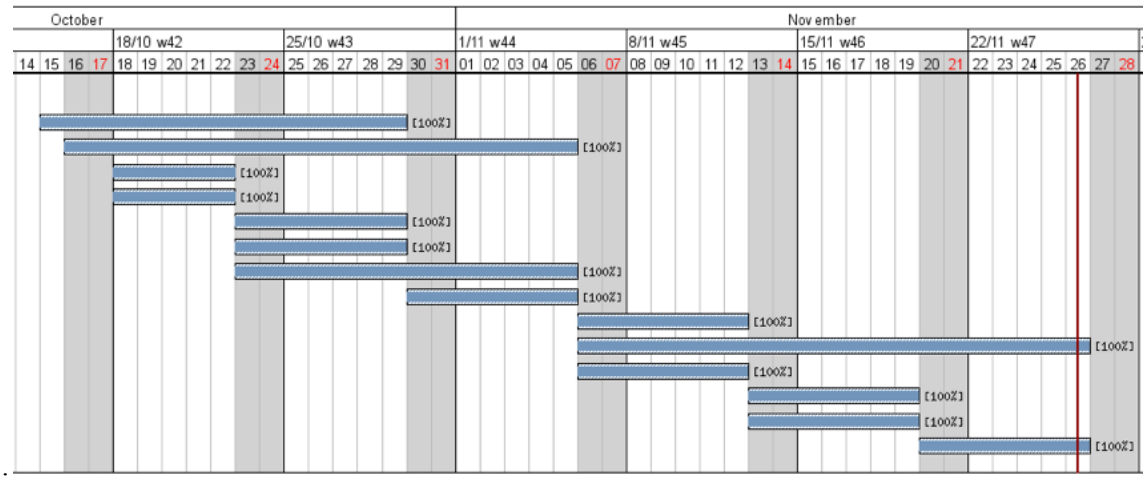


Figura 43. Actividades fase 3.

Task	Status	Est. Time	Duration	Start Date	Due Date
Acciones heroe	Completed		28 days	11 Jan 2011	08 Feb 2011
Cuadernillo de estudiante y docente terminado	Completed		31 days	11 Jan 2011	11 Feb 2011
Dialogo	Completed		28 days	11 Jan 2011	08 Feb 2011
Heore	Completed		28 days	11 Jan 2011	08 Feb 2011
Integración IDE y Compilador	Completed		53 days	11 Jan 2011	05 Mar 2011
Log	Completed		28 days	11 Jan 2011	08 Feb 2011
Mochila	Completed		28 days	11 Jan 2011	08 Feb 2011
Programación del análisis	Completed		31 days	11 Jan 2011	11 Feb 2011
Programación del heroe	Completed		24 days	11 Jan 2011	04 Feb 2011
XML/Diccionario	Completed		28 days	11 Jan 2011	08 Feb 2011
Plan de medición e instrumento de evaluación e Integración de manual con ejemplos de Software	Completed		4 days	28 Feb 2011	04 Mar 2011
Plan experimental	Completed		11 days	07 Mar 2011	18 Mar 2011
Corrección de errores en manual y/o software	Completed		11 days	21 Mar 2011	01 Apr 2011
Impresión de cuadernillos	Completed		11 days	04 Apr 2011	15 Apr 2011
Capacitación de maestros y Pre Test	Completed		11 days	25 Apr 2011	06 May 2011
Plan Piloto	Completed		84 days	09 May 2011	01 Aug 2011
Post Test	Completed		11 days	01 Aug 2011	12 Aug 2011

Figura 44. Gantt actividades fase 3

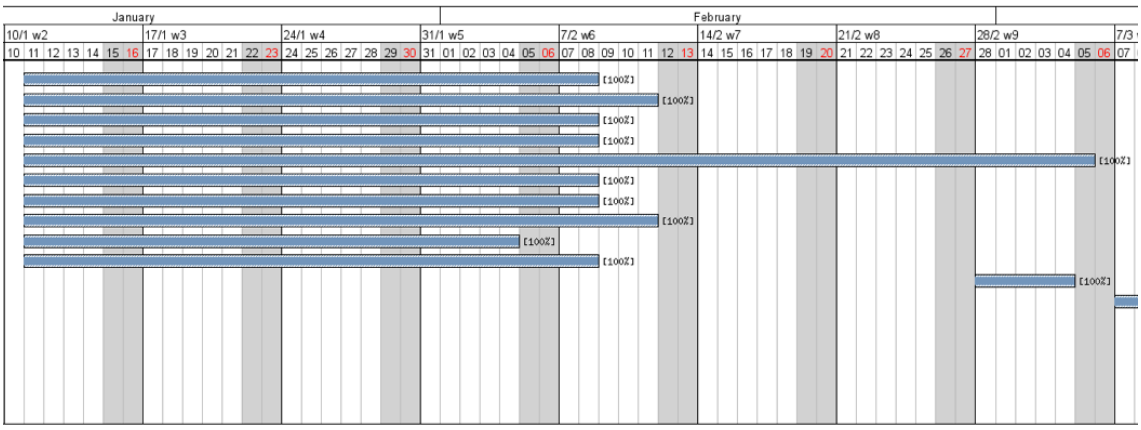


Figura 45. Gantt actividades fase 3.

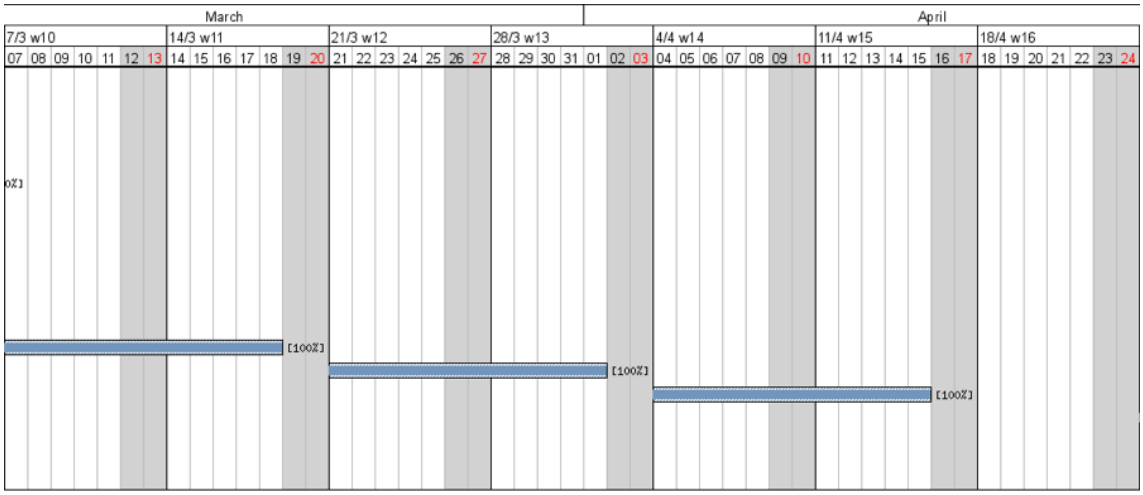


Figura 46. Gantt actividades fase 3.

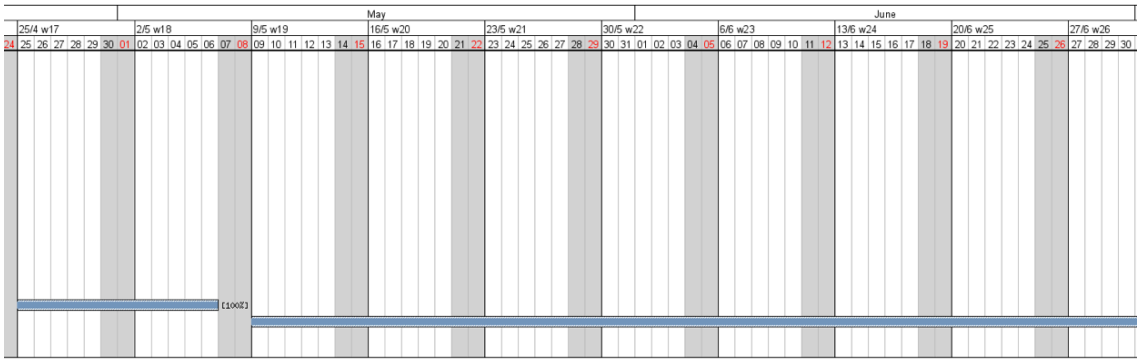
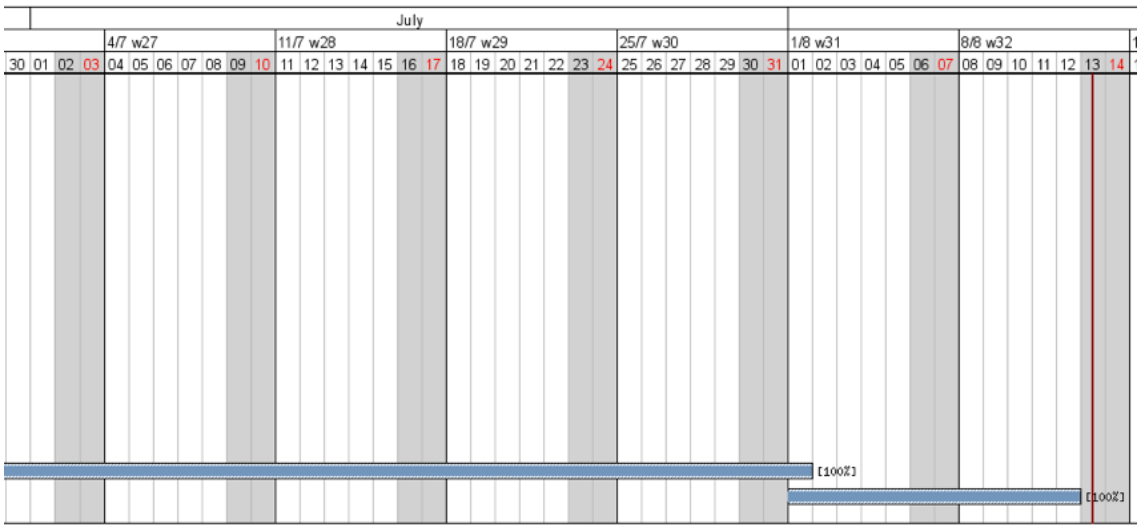


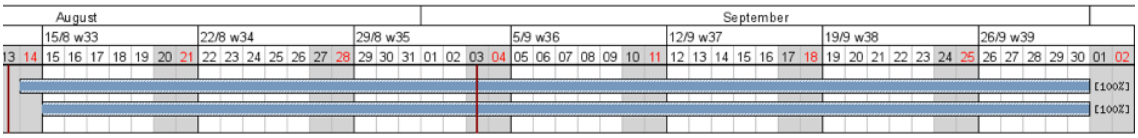
Figura 47. Gantt actividades fase 3.



Cuadro 13. Actividades fase 4.

Task	Status	Est. Time	Duration	Start Date	Due Date
Prupuesta MINEDUC	Completed		47 days	14 Aug 2011	30 Sep 2011
Análisis de los resultados	Completed		46 days	15 Aug 2011	30 Sep 2011

Figura 48. Gantt actividades fase 4.



3. Matriz de marco Lógico.

Cuadro 14. Matriz de marco lógico plan piloto.

Resumen Narrativo	**Indicadores**	**Medios de Verificación**	**Supuestos**
Reducir el índice de fracaso a nivel superior y alcanzar las competencias de aprendizaje autónomo a través de la implementación del curso de algoritmos y programación en el segundo ciclo del sistema de educación a nivel primaria.	Se aumentará en un 10% el número de alumnos de las escuelas participantes que están preparados para la Universidad.	Estudio con los profesores universitarios que atienden los primeros cursos en las diferentes Universidades. Las listas de los alumnos de las escuelas participantes que aprobaron los exámenes de admisión.	Las condiciones que se consideran importantes son los resultados que arrojará la prueba piloto, de esto dependerá la continuidad del proyecto.
Conseguir que el estudiante resuelva problemas de manera más fácil.	Se aumentará en un 20% de alumnos participantes, su capacidad para resolver problemas de una manera fácil y sencilla en sus cursos posteriores.	Análisis de varianzas. Dio como resultado que únicamente en una escuela había sido significativo el incremento de las medias.	Los resultados obtenidos luego de realizar el pre-test y el post test, no son significativos, es necesario incluir variables sociales para poder medir el verdadero impacto de las herramientas.
Software para resolver algoritmos, guías para el docente, Cuadernillos para los alumnos.	Los cuadernillos de los alumnos contendrán ejercicios y actividades que incentiven a los estudiantes a resolver problemas mediante una interfaz amigable. Las guías para el docente, con consejos para la enseñanza de	Se puede utilizar la metodología ágil para ver si se cumple en el tiempo la formación de cada uno de los manuales requeridos. Se realizaron entrevistas a los alumnos y maestros para saber su	La aceptación de los alumnos, según entrevistas realizadas, fue positiva. La aceptación de los maestros pudo haber sido buena, pero requiere más tiempo el poder cambiar una metodología de

4. Análisis de varianzas.

Cuadro 15. Análisis de varianzas por institución y general grupo experimental

Grupo Experimental						
Institución	**F**	**Probabilidad**	**Valor crítico para F**	**Diferencia F's**	**Diferencia 5%-Prob**	**Hipótesis nula**
Institución privada	6.74	6%	0.80	5.94	-0.01	Aceptada
Institución pública 1	7.07	1%	4.00	3.07	0.04	Rechazada
Institución pública 2	3.10	8%	4.01	-0.92	-0.03	Aceptada
Institución pública 3	4.38	4%	4.17	0.21	0.01	Rechazada
Institución pública 4	0.47	50%	4.23	-3.76	-0.45	Aceptada
General	7.50	0.01	3.88	3.61	0.04	Rechazada

95% de confianza

Cuadro 16. Análisis de varianzas por institución y general grupo control

Grupo Control						
Institución	**F**	**Probabilidad**	**Valor crítico para F**	**Diferencia F's**	**Diferencia 5%-Prob**	**Hipótesis nula**
Institución privada	0.63	43%	4.10	-3.47	-0.38	Aceptada
Institución pública 1	8.48	1%	4.05	4.43	0.04	Rechazada
Institución pública 2	7.23	1%	3.98	3.25	0.04	Rechazada
Institución pública 3	3.91	6%	4.26	-0.35	-0.01	Aceptada
Institución pública 4	3.91	6%	4.26	-0.35	-0.01	Aceptada
General	16.55	0.00	3.88	12.67	0.05	Rechazada

95% de confianza

Cuadro 17. Comparación de promedios grupo experimental.

Comparación promedios de Grupo Experimental				
Institución	**Pre-test**	**Post-test**	**Diferencia**	**Flag**
Institución privada	45.200	44.200	-1.000	
Institución pública 1	35.094	42.500	7.406	
Institución pública 2	26.933	31.933	5.000	
Institución pública 3	24.118	31.176	7.059	
Institución pública 4	24.857	27.857	3.000	
General	31.796	36.478	4.681	

Empeoró

Mejoró

Cuadro 18. Comparación de promedios grupo control.

Comparación promedios de Grupo Control				
Institución	**Pre-test**	**Post-test**	**Diferencia**	**Flag**
Institución privada	39.000	36.714	-2.286	
Institución pública 1	33.240	40.440	7.200	
Institución pública 2	26.919	33.297	6.378	
Institución pública 3	24.643	30.500	5.857	
Institución pública 4	22.200	32.933	10.733	
General	29.679	35.134	5.455	

Empeoró
Mejoró

A continuación se discute la secuencia de resultados generales que se obtuvieron y además validarán la metodología propuesta de Proyecto ABJ, para la ejecución de un plan a nivel micro y macro para mejorar la calidad de enseñanza a través del uso de algoritmos en el país.

En la etapa de planeación, en mi opinión, se empezó por lo más importante que se debe de realizar en cualquier empresa o proyecto, que es definir claramente los objetivos generales y específicos. Para esto se realizaron dos sesiones plenarias en donde todo el equipo participó en la creación de ellos. Además, mediante una lluvia de ideas, se propuso el titulo formal y el comercial que identificaría a nuestro proyecto y por ende el que se iba a utilizar.

Luego de haber definido los objetivos, se procedió a iniciar la etapa de planificación de actividades a realizar durante los tres semestres. La planificación en un inicio se realizó únicamente para los primeros dos meses y medio, situación que conllevo que al finalizar este periodo se realizará una nueva programación de actividades para la siguiente fase. En la segunda planificación se realizó toda el plan de actividades que faltaba del semestre y para el próximo año (un año y tres meses aproximadamente), lo cual generó que todas las actividades por hacer estuvieran calendarizadas y más importante aún, con tiempos estimados de duración, máximos y mínimos por actividad para asegurar el cumplimiento en tiempo del megaproyecto.

Además, para llevar a cabo buenas prácticas administrativas en el Megaproyecto, fue importante estructurar debidamente el equipo, con el objetivo de satisfacer los requerimientos del mismo y cumplir los objetivos estratégicos con las personas adecuadas, según la carrera y características específicas. Para esto el equipo se estructuró con dos estudiantes de Ingeniería Industrial, dos de Licenciatura en Educación y siete de Ingeniería en Ciencias de la Computación.

Así mismo, con el objetivo de administrar eficaz y eficientemente el proyecto se utilizó un software en línea llamado “Project QDPM”, éste fue de gran utilidad por las siguientes razones:

En él se construyó el diagrama Gantt de actividades por cada una de las fases planificadas.

Al asignarle alguna actividad a una persona, o grupo de personas, el software enviaba automáticamente un correo, en el cual notificada todo lo relacionado con la actividad.

Se utilizaron tickets que inmediatamente eran enviados al correo de la persona a la que se le quería notificar alguna situación o bien para notificar sobre el tiempo que le quedaba para su actividad, así mismo para realizar consultas de la misma, o bien, para exigirle a algún encargado que actualizará el estado de alguna actividad en especifico.

Todos los miembros podían entrar al software desde cualquier computadora con internet y consultar sobre el estado de sus actividades o bien, del megaproyecto en general.

Es un programa flexible que permite realizar cambios a las fechas de entrega de la actividad y cambios de los recursos a utilizar; pero más importante aún, era fácil de utilizar y amigable con el usuario.

Agilizó la comunicación entre los miembros del equipo y en gran medida evitó malentendidos, ya que cada persona o equipo fueron los encargados de actualizar el estado a cada una de sus actividades.

Con respecto a administrar eficientemente el megaproyecto, el software fue de gran utilidad debido a se utilizó para administrar de forma eficiente las actividades y todo el megaproyecto en general, ya que toda la información se encontraba centralizada. Además de ser útil para realizar revisiones sobre el cumplimiento de cada persona o equipo, en la entrega de las actividades asignadas en tiempo y según el cronograma. Para la administración de las actividades, si bien es cierto el software trajo beneficios, siempre es complicado trabajar con personas y esto va muy de la mano con la responsabilidad de cada una de ellas. Como administrador del proyecto, hubiese sido ideal no andar como “capataz” detrás de ellas, pidiéndoles que actualizaran los estados de sus actividades o bien que utilizaran el software. Dado lo anterior, en muchas ocasiones, se tuvo que utilizar otro medio de comunicación para poder informarme sobre el avance de las actividades para poder actualizar el estado de ellas y así llevar el control necesario de las mismas.

Dentro del proceso administrativo, se hizo un presupuesto de gastos y costos a incurrir dentro de las fases del megaproyecto. Para realizar el mismo se llevó a cabo una investigación extensiva

de todos los posibles costos en los que se iban a incurrir. Posteriormente se realizó el primer presupuesto tomando en cuenta los rubros mencionados en la fase de diseño. Al hacer el primer presupuesto, se tenía pensado pasar la prueba a alrededor de 2,000 estudiantes en toda la republica de Guatemala, pero el costo era muy elevado. Al estudiar la situación, se procedió a delimitar el impacto del Megaproyecto únicamente a cinco instituciones educativas del área urbana en la ciudad capital, teniendo a 175 niños dentro de nuestra muestra. El hecho de tener dicha muestra, fue debido a que el presupuesto disponible en el departamento de computación para el proyecto era de Q 10,000.00 y por ende, fue necesario evaluar la cantidad de estudiantes que iban a ser sometidos a la prueba para lograr acomodar nuestros costos a dicho presupuesto.

Los rubros más importantes en dicho presupuesto fueron el de la impresión de cuadernillos y el de las pruebas estandarizadas. El primer costo, era directamente proporcional a la cantidad de la muestra, es decir, a mayor cantidad de niños, mayor cantidad de cuadernillos a imprimir y por ende, mayor costo. De igual forma era el de las pruebas estandarizadas, el costo de cada una de las ellas era de Q 25.00 independientemente de la cantidad de las pruebas a realizar, pero por ser un proyecto de la universidad, por la razón de que el CIE tiene relación directa con la Universidad del Valle de Guatemala, nos brindaron el apoyo necesario y nos proporcionaron las pruebas a Q 10.00 cada una, algo que nos benefició a que nuestra muestra no tuviera que ser menor. Además, el hecho de tener dos grupos, uno control y uno experimental, y de tener que llevar a cabo un pre-test y un post-test, llevó a que el costo de las pruebas fuera el doble y que acaparará buena parte del presupuesto disponible, alrededor del 60%.

La Matriz de Marco Lógico se realizó luego de haber planificado y tener el presupuesto del Megaproyecto. Sintetizar todo lo que fue el proyecto en una página, que contiene cuatro filas y cuatro columnas en donde detalla y estudia el estado actual que en este caso son los problemas e involucrados, y además detalla la situación deseada que en este caso serían los objetivos y las posibles alternativas. La misma fue creada para que personas ajenas al Megaproyecto que se interesaran o desearan invertir, se enteraran de todo lo que se quería alcanzar, cómo se iba a alcanzar, con cuánto se iba a lograr y de dónde se estaba partiendo para lograrlo.

La fase uno y dos del megaproyecto fueron puramente de investigación y planeación. Durante éstas, cada persona era encargada de presentar el día de la reunión acordada lo que había realizado. Las siguientes dos fases ya fueron de campo. La fase tres, fue la fase critica y más importante del Megaproyecto, durante la misma se llevó a cabo la producción del Cuadernillo del Docente y del Alumno, además se diseño y programó el Software a utilizar. Al tener terminadas

las tres herramientas mencionadas previamente, se procedió a realizar pruebas con niños de diversos establecimientos educativos para poner a prueba y verificar si el contenido de las mismas era entendible y acorde a la edad de los estudiantes.

Para esto se coordinó la visita de dos establecimientos, uno público y uno privado. Dentro del público se llevó a cabo la prueba de manual del alumno y del docente, mientras en el privado se realizaron las pruebas del software. Se necesitó de logística para poder coordinar con los establecimientos las pruebas y así mismo con el equipo, para poder tener el personal necesario para ejecutarlas. La primera visita se realizó a la institución privada, en donde se realizó la prueba del Software y no se tuvieron inconvenientes. La visita en el establecimiento público fue igual de simple, lo único que se necesitó del equipo de educación para que ellas fueran las encargadas de realizar dicha prueba, por lo que se tuvo que coordinar la visita la visita al tiempo de ellas y a la disponibilidad de la escuela.

Al tener terminadas las pruebas de usabilidad, tanto para el software como para los cuadernillos, se procedió a corregir los errores que se encontraron en los mismos. Luego, al tener los cuadernillos listos se enviaron a la imprenta para que se encargaran del proceso de impresión. Durante el tiempo de impresión, una semana, se procedió a planificar las capacitaciones para los maestros y a coordinar las fechas con el CIE para realizar el pre-test utilizando las pruebas estandarizadas. Inmediatamente terminado el descanso de Semana Santa, es decir del 25 de abril del 2011 al 29 del mismo, se procedió a capacitar a los maestros con la ayuda del equipo de Educación de la Universidad del Valle de Guatemala en cada uno de los establecimientos. Por cuestiones de tiempo se decidió capacitar, en uno de los casos, a dos profesores a la vez y en todos los demás la capacitación fue personal y durante el tiempo que los profesores se encontrarán libres.

La principal actividad de la fase dos, fue la del plan piloto. En dicho plan las herramientas diseñadas se pusieron en práctica para analizar si en realidad generaban esa habilidad para resolver problemas en los alumnos.

Durante el plan piloto, se asignó a cinco personas del equipo para que fueran los encargados de la supervisión semanal en cada uno de los establecimientos educativos. Para esto, el equipo de Educación, realizó una hoja de actividades a realizar por visita (ver anexos). Dicha hoja abarcaba todo lo necesario para poder determinar si el docente estaba impartiendo de forma adecuada el curso, o si necesitaba algún tipo de colaboración de parte del equipo. Además, también tenía

como objetivo la retroalimentación sobre la reacción de los niños hacía la herramienta, desde el punto de vista del docente.

Cuando el plan piloto estaba por finalizar, dos semanas antes de su fin, se procedió a realizar la logística para llevar a cabo los post-test utilizando la segunda versión de las pruebas estandarizadas. El principal inconveniente con el que se topó en este momento de la planificación, fue que todo el magisterio estaba realizando paros en las carreteras y los maestros estaban ausentándose de sus labores. Se logró planificar con las escuelas y el CIE, el día de las pruebas y se realizaron sin inconvenientes.

Al finalizar el plan piloto, las actividades pendientes a realizar eran mínimas. Se estuvo al pendiente de los resultados de las pruebas estandarizadas durante dos semanas. Luego de que fueran entregados por el CIE, se procedió a tabular los resultados. La tabulación se llevó a cabo por escuela, con base a cada sección y en base a cada grupo, ya sea experimental o control.

Los resultados obtenidos, luego de aplicarles un análisis de varianzas con un alfa de 0.05, nos dicen que en general, tanto el grupo control como el grupo experimental, no tuvieron una diferencia entre sus medias para el pre-test y post-test; lo cual si se saca una conclusión precipitada, nos dice que la herramienta no tuvo impacto en la muestra.

Al llevar a cabo un análisis de varianza por instituciones educativas con un alfa de 0.05, se pudo determinar que únicamente en Instituto público 3 el grupo experimental sufrió un crecimiento en sus medias significativo y el grupo control mantuvo sus medias iguales, lo que quiere decir que en esta escuela hubo un crecimiento debido a la aplicación de la herramienta.

En otros casos, el grupo experimental tuvo un crecimiento con respecto el pre-test y el pos-test significativo, mientras el grupo control mantuvo sus medias iguales. Este es el caso para la institución pública 2, lo que quiere decir que dentro de ésta escuela la herramienta no generó un impacto.

Los otros tres establecimientos presentaron, un crecimiento en sus medias, tanto para el grupo experimental como para el control, lo que no nos garantiza si la aplicación de la herramienta logró el objetivo.

En general al analizar los resultados, se podría decir que la herramienta no tuvo un impacto y no causó nada. Pero la razón de estos resultados, puede estar influida como cualquier examen, no sólo por la capacidad del alumno, sino también por razones psicológicas. Seguramente, dentro de

la aplicación de la prueba piloto se debió haber medido otras variables para lograr determinar si en realidad la herramienta funcionó o no. Razones de comportamiento también se evaluaron en la hoja que los supervisores tenían que pasar semanalmente, en ella todos los maestros indicaron que los alumnos habían logrado desarrollar problemas matemáticos, de ciencias naturales y otras áreas, aplicando los algoritmos, por lo que dar conclusiones en base únicamente al análisis estadístico podría ser equivocado.

Cuadro 19. Matriz de marco lógico a nivel nacional.

Resumen narrativo	**Indicadores**	**Medios de verificación**	**Supuestos**
Reducir el índice de fracaso a nivel superior y alcanzar las competencias de aprendizaje autónomo a través de la implementación del curso de algoritmos y programación en el segundo ciclo del sistema de educación a nivel primaria.	Se aumentará en un 5% el número de alumnos de las escuelas participantes que están preparados para la Universidad.	Estudio con los profesores universitarios que atienden los primeros cursos en las diferentes Universidades. Las listas de los alumnos de las escuelas participantes que aprobaron los exámenes de admisión.	Las condiciones que se consideran importantes son los resultados que arrojará la prueba piloto, de esto dependerá la continuidad del proyecto. En este caso para ser aplicado a 4to, 5to y 6to primaria a nivel nacional.
Conseguir que el estudiante resuelva problemas de manera más fácil.	Se aumentará en un 10% de alumnos participantes, su capacidad para resolver problemas de una manera fácil y sencilla en sus cursos posteriores.	Se realizará una prueba piloto que nos proporcionara que tanto efecto, tiene el uso de algoritmos para las personas que fueron sometidas a cada una de las capacitaciones y pruebas.	La comprobación de los resultados del plan a nivel nacional, será fundamental para aplicarlo a más establecimientos y abarcar todo el segundo ciclo de primaria.

Continuación Cuadro 19

Resumen narrativo	Indicadores	Medios de verificación	Supuestos
Conseguir que el estudiante resuelva problemas de manera más fácil.	Se aumentará en un 10% de alumnos participantes, su capacidad para resolver problemas de una manera fácil y sencilla en sus cursos posteriores.	Se realizará una prueba para comprobar el efecto, esto se hará por medio de un Pre y Post Test, el cual nos dirá el impacto que tiene el uso de algoritmos para las personas que fueron sometidas a cada una de las capacitaciones y pruebas.	Los resultados del proyecto darán la pauta para proseguir con el proyecto y poder ser aplicado en más grados de primaria y en escuelas rurales.
Guía para la capacitación del docente. Cuadernillo para los alumnos. Software.	El costo de cada manual es de Q13.125 ya que cuenta con 15 páginas hechas a dos colores, tamaño carta, papel bond 80 gramos, y la caratula en papel couché base 10.	Se tiene en la tabla de costos proyectados que la cantidad a invertir es de Q6,044,773.75 en donde incluye cuadernillos de trabajo, capacitación, visitas para revisión, y los exámenes pre y post test.	La entrega en el tiempo establecido de los folletos que se van a imprimir, para ejecutar las pruebas necesarias. La ejecución correcta del proyecto propuesto.

Cuadro 20. Resultados del costeo

Escuelas a Nivel Guatemala del Área Urbana		
	Costo U.	**Costo Total**
Manual de docente (3000)	Q 13.13	Q 39,375.00
Cuadernillos (62386)	Q 13.13	Q 818,816.25
Examen Pre-Test (62386)	Q 40.00	Q 2,495,440.00
Examen Post-Test (62386)	Q 40.00	Q 2,495,440.00
Capacitación Docentes		Q 12,185.00
Revisión de Uso de Herramienta		Q 97,480.00
Gastos de Servidor		
Usar QDPM	Q 130.00	Q 1,040.00
Total de Costos		**Q 5,959,776.25**

El detalle de los costos se encuentra en el Cuadro 22 del apéndice, en el cual se debe de tomar en cuenta que las visitas para revisiones son 8 y la capacitación es única.

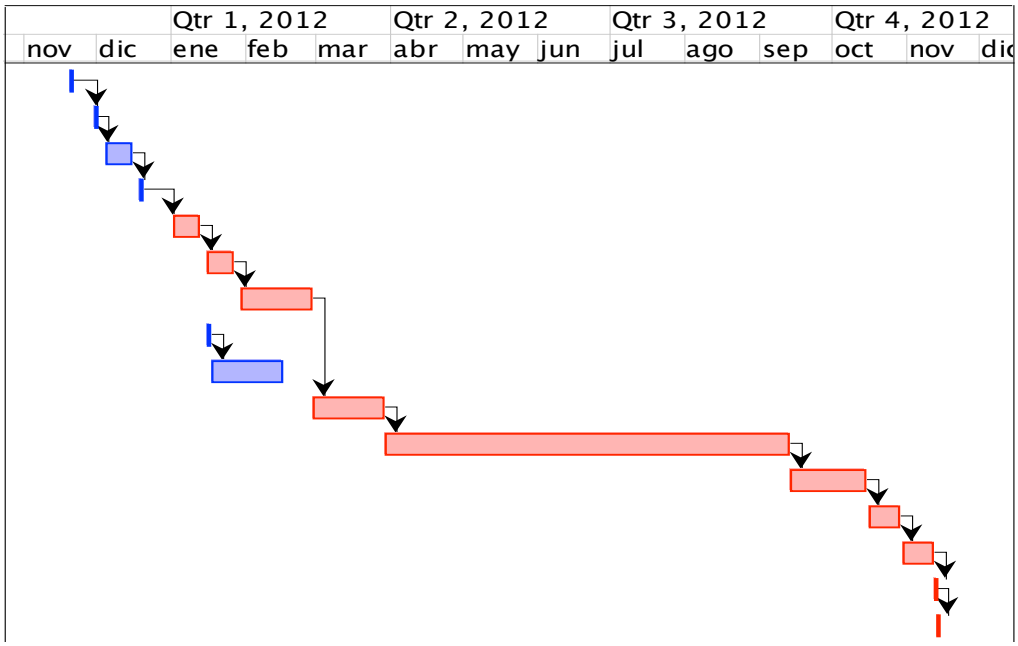
Cuadro 21. Análisis beneficio / costo.

Beneficios	
Alumnos en Primaria	62386
Alumnos que llegan a la Universidad	7486.32
Cantidad de Cursos en primer año	10
Cantidad de Cursos en beneficio	1
Costo de cada curso	Q 3,300.00
Beneficio de alumnos	**Q 24,704,856.00**
Análisis Beneficio/Costo	
Beneficios	Q 24,704,856.00
B/C	**4.145265688**

Figura 49. Actividades proyecto a nivel nacional.

	Nombre	Duración	Inicio	Terminado	Predecesores	Nombres del Recurso
1	Exposición de Proyecto	1 day?	21/11/11 8:00	21/11/11 17:00		
2	Aprobación del Proyecto	1 day?	1/12/11 8:00	1/12/11 17:00	1	
3	Planificación del Pre Test	10 days	5/12/11 8:00	16/12/11 17:00	2	
4	Planificación de la Impresión de Cuadernillos	2 days	19/12/11 8:00	20/12/11 17:00	3	
5	Planificación del Post Test	10 days	2/01/12 8:00	13/01/12 17:00	4	
6	Planificación de Capacitación Docentes	10 days	16/01/12 8:00	27/01/12 17:00	5	
7	Aplicación de Pre Test	22 days	30/01/12 8:00	28/02/12 17:00	6	
8	Impresión de Cuadernillos	2 days	16/01/12 8:00	17/01/12 17:00		
9	Distribución de Cuadernillos	22 days	18/01/12 8:00	16/02/12 17:00	8	
10	Capacitación del Docente	22 days	29/02/12 8:00	29/03/12 17:00	7	
11	Revisión de la Herramienta	120 days	30/03/12 8:00	13/09/12 17:00	10	
12	Post Test	22 days	14/09/12 8:00	15/10/12 17:00	11	
13	Entrega de Resultados	10 days	16/10/12 8:00	29/10/12 17:00	12	
14	Ananlisis de resultados	10 days	30/10/12 8:00	12/11/12 17:00	13	
15	Exposición de Resultados	1 day?	13/11/12 8:00	13/11/12 17:00	14	
16	Planeación para continuación de Proyecto	1 day?	14/11/12 8:00	14/11/12 17:00	15	

Figura 50. Diagrama Gantt.



A continuación se presenta la discusión de los resultados anteriores:

El objetivo principal de este proyecto es dar a conocer el uso de algoritmos, para el aprender a resolver problemas en el segundo ciclo de primaria específicamente en el grado de quinto primaria. Específicamente se esta abarcando las escuelas del área urbana. Al tomar dicha área estamos dedicándonos y teniendo la certeza de que todos hablan español, ya que los cuadernillos de trabajo están hechos en dicho idioma.

Si tomamos en cuenta las escuelas del área rural, nos tendríamos que contactar con gente traductora, y como el país es multilingüe, estaríamos hablando de un costo extra para la aplicación del proyecto, y como se está en una etapa de aceptación, se contemplo únicamente el área urbana.

Se deben de tomar en cuenta todos los costos y que el proyecto es a nivel de Guatemala, en la etapa previa se tomaron 4 escuelas y un colegio, y ahora se tomarán casi 1300 establecimientos, es importante tener en cuenta todas las actividades correspondientes.

Todas las actividades propuestas para el proyecto que se está proponiendo con forme a un plan establecido. Existen actividades que son consideradas de vital importancia, ya que no se pueden atrasar y no tienen holgura porque si se atrasan dichas actividades también se atrasara todo el proyecto.

La aplicación del pre test es una de estas, ya que sin estos no se puede iniciar a utilizar la herramienta, al igual que la capacitación de los docentes es de las actividades que se consideran de vital importancia, si los maestros no tienen conocimiento de dicho material, aunque lo tenga en sus manos no sabrá como guiar a sus alumnos para que estos puedan aprender de una forma diferente y solo confundirá a los alumnos, ya que cuando se le de la información y capacitación vera en el error que tuvo.

La impresión de cuadernillos tanto de maestros como de alumnos, es importante establecer buena relación con los proveedores para que la entrega sea en el tiempo correcto y en el lugar especificado. Estos tienen que estar con días de antelación, para que cuando se de la capacitación, ya al siguiente día estén listos para poder aplicar la herramienta con sus alumnos.

Por el contrario las revisiones no se tienen que tener fechas específicas, ya que en éstas solo se evalúa si se está utilizando bien la herramienta. Los exámenes posteriores a la aplicación de los cuadernillos, si es de suma importancia ya que esta nos dará el efecto que tuvo el proyecto durante el año escolar.

La matriz de marco lógico es considerada como la herramienta que se aplica para la presentación de resultados a una entidad publica, esta consta de una matriz cuadrada de cuatro filas por cuatro columnas, en las cuales se colocan los objetivos que se quieren alcanzar durante el proyecto, es importante reconocer que dicha matriz habla de algo a corto, mediano y largo plazo, la cual nos da la pauta a donde queremos llegar en este momento y en el futuro, pasando por todos los procesos.

Existen casillas de dar a conocer el impacto que se espera que tenga dicho proyecto, pero se especifica que son supuestos que se tienen para cumplirlos. Se tomara en cuenta que trabajar con el gobierno no es como trabajar en el sector privado, ya que en el gobierno se tienen manejo de influencias, se busca el interés propio y no del país, se maneja mucha corrupción, así como muchas cosas más.

El considerar a maestros que trabajan doble jornada por lo bajo de los salarios, dan una pauta, que muchas veces no dan el rendimiento que se esperaría por el mismo cansancio.

Las condiciones de pobreza de muchos departamentos de Guatemala también forman parte de algunos problemas que se pueden tener, ya que está comprobado que los alumnos que no desayunan, tienen algunos déficits de atención y de entendimiento durante el tiempo de estudio.

Considerando que al inicio de todo el proyecto de prueba se tendían muchas expectativas a cerca de los resultados de las pruebas. Se consideró como medio de presentación al ministerio de educación la matriz de marco lógico la cual nos da a conocer todos los alcances que se pretenden en el proyecto, además de costos y algunas cosas relacionadas con la gente que pretende ver algunas cosas de estas.

En la presentación se pretende dar a conocer todo para que el plan lo realicen durante el año 2012 o 2013, teniendo en cuenta todos, los costos y acciones necesarias para que el proyecto sea un éxito.

El éxito del proyecto se baso en las capacitaciones a docentes, revisiones de cómo se estaba pasando cada una de las etapas, donde los niños iban aprendiendo día con día todas las técnicas para ser mejores en su resolución de problemas.

Con respecto a los costos que se tienen contemplados, es importante considerar, que los datos obtenidos para el número de cuadernillos en esté caso aproximadamente 64,000, se obtuvieron de las tablas de datos proporcionadas por el MINEDUC en el 2008, cabe recalcar que estos datos son

los últimos que se tienen para este reporte que hace dicho ministerio. Los datos de los manuales para docentes se obtuvieron de la cantidad de establecimientos en el área urbana, pero se hizo una proyección de crecimiento de 2.5%, lo cual nos da un dato final de 68,862 cuadernillos, para los manuales de docentes se tienen un aproximado de 3000 cuadernillos, ya que estamos tomando que existirán 1,500 establecimientos con 2 grados de quinto primaria, lo que nos da 3,000 manuales de docente. Se estimo según los costos que tuvimos, que cada manual y cuadernillos tendrán un costo estimado de Q13.13.

Para los exámenes de pre test y post test, se tienen costos mayores a lo que se trabajo durante la aplicación del pre proyecto, ya que, por ser parte de la Universidad, se recibió un descuento, pero el precio normal, es el que se encuentra en la tabla de costos (Q40), pero esto incluye el paso de los exámenes en cualquier lugar del país, sin tener que pagar viáticos y ellos organizan la logística para el paso de dichos exámenes, además dan a conocer los resultados de dichos exámenes.

La parte que se considera de mayor importancia son las capacitaciones a docentes y las revisión de que se este utilizando correctamente la herramienta.

Para el costeo de las capacitaciones y visitas a cada departamento, se tomaron como referencias, hoteles de aproximadamente Q300, que es un precio considerable, también se tomaron en cuenta para la cantidad de establecimientos de cada departamento. Las comidas de cada persona se tomaron como un promedio de Q30, tanto desayunos, almuerzos, como cenas, balanceando que no todos los tiempos son del mismo valor.

Como una persona será la encargada de capacitación y de revisar, tomando en cuenta que no se puede hacer en días seguidos, es de suma importancia establecer que ellos irán y vendrán, día a día, solo con las excepciones de los departamentos que tienen establecidos dos días para quedarse una noche en dicho departamento.

Para el transporte se tomaron en cuenta que el galón de gasolina estará a Q35 promedio, y se tomaron datos de la distancia de Guatemala a cada departamento. Ya que muchas veces se irá cada persona a un departamento. Todas las distancias se encuentran en el Apéndice.

Por otra parte para que cada quien vea y actualice todas sus actividades, se usara un servidor, en el cual se encontraran todas las actividades asignadas para cada persona involucrada dentro del proyecto. Es importante que todas estas actividades sean actualizadas, para ver como va el avance

del proyecto en sí. Tomando en cuenta que todos tienen acceso a internet en estos, días es de suma importancia este tipo de herramientas virtuales.

Al final los costos estimados de inversión para el proyecto está estimado en Q 5,959,776.25 lo cual nos dice lo que el gobierno debería de dar para que esto se lleve a cabo durante los años propuestos con anterioridad.

Cabe señalar que se están tomando en cuenta 8 revisiones por departamento que se establecerán según se vaya necesitando cada una de las mismas, quedará a discreción de la gente que ejecute el proyecto el cual dará a conocer la cantidad de visitas. Lo que se recomienda son las establecidas dentro del plan que se está proponiendo.

En la propuesta de logística se tomaron en cuenta sede central, regional y de sector, lo cual nos da una idea de que dichas sedes también pueden funcionar como guías para las demás escuelas, lo cual ayudaría también a tener menos revisiones, pero siempre los ejecutores del proyecto deben de estar atentos para poder dar a conocer como lo llevaran a cabo y de que forma, si tiene alguna modificación al mismo.

Los cuadernillos y manuales se estarán trasladando de la sede central, a la regional, y de la regional a las sedes de sector, esto para que las escuelas tengan una apoyo más con sus sedes regionales, en donde tendrán todo el material a utilizar durante el proyecto y si necesitan algo solicitarlo mediante ellos, esto facilitará la comunicación con los mismos, ya que, en este tipo de proyectos la comunicación es lo más importante para el éxito del mismo.

En cuanto al análisis Beneficio/Costo se tomaron como todos los costos, lo discutido con anterioridad, para los beneficios se calculó con base a los alumnos que se tienen y que solo el 12% de los mismos llegará a la universidad, con esto podemos decir que alrededor de 7486 alumnos ingresaran al primer año de la Universidad.

Con esto decimos que el costo de un curso es de Q3,300.00, esto basado en costos de la Universidad del Valle de Guatemala, se planteo así debido a que en otras universidades el costo es menor por cada clase recibida, pero la cantidad y el volumen de las mismas es mayor, esto conlleva a que es rentable con costos más bajos, por el volumen de estudiantes que tiene cada una de ellas. En el caso de la Universidad de San Carlos, es importante mencionar que los alumnos costean una pequeña parte del valor del curso, pero el resto es cargado al gobierno, lo cual tomamos como dato el costo de la Universidad del Valle por clase.

Al tener estos datos es importante reconocer que el impacto es que ganen el curso de Programación Básica el cual, al perderlo les genera una perdida del total del curso, por lo tanto nuestro beneficio económico se calcula con base a los alumnos que llegarán a la Universidad, en conjunto con el valor de un curso de Programación.

Teniendo ya nuestros beneficios que son de Q 24,704,856.00 y teniendo nuestros costos, tenemos que nuestro beneficio/ costo es de 4.145 lo cual es mayor de 1, en el análisis se toma como factible y que se puede realizar el proyecto si este sobre pasa el 1, lo que significa que los beneficios serán mayores que los costos. En realidad este tipo de análisis solo se realizan para proyectos que son públicos y que son financiados por el gobierno, ya que los beneficios se deben obtener de alguna forma.

VII. CONCLUSIONES Y RECOMENDACIONES

La experiencia del Megaproyecto dejó claro que es necesario contar con un grupo de cuatro o más docentes para capacitación para asegurar la interacción entre los mismos a fin de que puedan experimentar tanto la elaboración de los algoritmos, pero más importante aún, la práctica de comparar, analizar y decidir. De no ser posible la capacitación de varios docentes se recomienda dar seguimiento al docente capacitado a través de visitas al salón de clases para corregir u orientar adecuadamente cualquier aspecto en el que sea necesario. La herramienta ha sido efectiva siempre y cuando se practique como mínimo tres veces por semana; la frecuencia fomenta la creación del hábito necesario para la autonomía personal y la práctica de comparar-analizar-decidir, el pensamiento lógico.

Los logros más importantes del uso de la herramienta en las cinco instituciones educativas fueron la buena aceptación que la misma tuvo entre docentes y alumnos, así como el deseo de los maestros de continuar utilizando los algoritmos dentro del salón de clase, motivados por los cambios positivos que pudieron observar en sus alumnos. Dentro de los factores que determinaron esto se pueden mencionar: (a) la motivación al docente a través de la capacitación previa que se hizo, (b) el apoyo al docente a través de reuniones semanales y visitas periódicas al aula (c) el enfoque lúdico del cuadernillo y (d) el contenido del cuadernillo que responde al contexto del alumno.

En vista de lo anterior, se recomienda realizar las capacitaciones de docentes en grupos de por lo menos cuatro miembros para simular una clase en donde ellos son los alumnos y tienen la oportunidad de hacer las comparaciones y análisis que la herramienta requiere. Se recomienda también la organización de sesiones de docentes cada dos o tres meses para que puedan compartir sus aciertos, desaciertos, experiencias y dudas sobre el programa. Durante las capacitaciones y seguimientos, se debe hacer énfasis, también, en que los docentes trabajen los algoritmos de forma dinámica, a través de la dramatización u otra práctica lúdica y que los practiquen por lo menos tres veces a la semana, incluyéndolos dentro los contenidos de sus áreas curriculares.

Estudios internacionales sobre capacitaciones cortas y sus resultados o aplicación en el aula a largo plazo indican que si no se crea conciencia o se logra que los docentes capacitados se sientan corresponsables de la importancia de lo aprendido en la capacitación esto se quedará en buenas intenciones. Las capacitaciones “magistrales” darán como resultado aprendices “pasivos”. La enseñanza debe promover la participación del capacitado, y al mismo tiempo debe valorar tanto sus conocimientos como su experiencia docente, lo que implica analizar las fortalezas,

limitaciones y el contexto de cada uno. Según dichos estudios, atender lo anterior aumenta la posibilidad de que el tema capacitado sea aplicado a largo plazo en el aula.

El Megaproyecto que originó este trabajo de graduación puede ser el fundamento para desarrollar otra herramienta que sea validada entre grupos indígenas del área rural. También se recomienda evaluar los efectos de la herramienta a largo plazo (mínimo dos o tres años). A pesar de que no exista información de esto último, es importante que autoridades del Ministerio de Educación consideren incorporar esta herramienta (el cuadernillo, el manual del docente y la herramienta computarizada) en el Currículum Nacional Base, empezando desde en el primer ciclo de primaria, a la mayor brevedad posible.

Se concluye que una mayor cantidad de iteraciones en un diseño de pruebas de usabilidad iterativo mejoró significativamente la usabilidad del software desarrollado.

Se concluye que la estructura de separación de módulos definida otorgó la abstracción necesaria para agilizar el trabajo de los grupos gráfico y lógico del megaproyecto.

Se concluye que el desarrollo de un lenguaje formal que defina oraciones simples en lenguaje español para representar cada uno de los pasos de un algoritmo simple es factible. La planificación de actividades, puede cambiar, pero es de suma importancia que se planifiquen actividades para todo el proyecto desde el inicio, con esto se pueden ir modificando o eliminando en el camino, pero deben de estar para tener una idea de lo que se realizará.

La división de las actividades en 4 fases, fue un factor elemental que benefició a todo el equipo para poder cumplir con el objetivo de cada módulo.

Al tener una buena logística y una metodología durante las visitas a los establecimientos educativos hizo que se eliminaran tiempos perdidos y generó que las visitas fueran eficientes, dinámicas, rápidas y enriquecedoras para ambas partes.

Un proyecto debe tener claro el alcance que puede llegar a tener con los recursos disponibles, es decir el presupuesto de gastos debe ir de la mano con el capital disponible.

La matriz de marco lógico es, sin duda, una buena carta de presentación para llevar a cabo propuestas de proyectos públicos, ya que en una hoja sintetiza todo lo que conlleva el proyecto.

Utilizar una herramienta en línea como Project QDPM, trae consigo beneficios como el tener centralizada la información, pero requiere de una cultura de responsabilidad de todos los miembros para que se le de el uso adecuado.

La única institución en donde, según el análisis de varianzas, se produjo un impacto positivo fue en la Institución pública 3.

En tres instituciones no se pudo determinar si hubo un impacto debido a la aplicación de la herramienta, o por algún crecimiento de sus conocimientos ajenos a ella.

En la institución pública 2 el grupo experimental mantuvo sus conocimientos a pesar del aprendizaje de la herramienta y el grupo control aumento sin estar expuesto a la herramienta.

A pesar de que en el instituto Público 2 fue el único en utilizar la herramienta computarizada, no se ve reflejado en los resultados la utilización de la herramienta.

Es erróneo calificar el plan piloto únicamente por los resultados estadísticos, si bien es cierto son parte fundamental, existen otras variables sociales que son exógenas a las pruebas estandarizadas que no son medidas en estos.

Con los resultados del análisis Beneficio/Costo mayor a 1, decimos que el proyecto es factible y de un impacto para la sociedad, ya que con esto, se está mejorando la calidad de educación en el país.

Los proyectos educativos, son de suma importancia, para el gobierno y para la comunidad, ya que ambos se benefician, con educación para la gente y con mejores profesionales que se encarguen del país.

La utilización de sedes centrales, regionales y de sector, ayudan a la mejor comunicación del proyecto, para llegar a manejar mejores resultados y una mejor aceptación de la gente al proyecto.

Los pre test y post test aplicados por la gente capacitada, nos dan el respaldo de que los resultados del impacto del proyecto serán confiables para el ente que ejecuta el proyecto.

Se recomienda continuar utilizando el Software de Project QDPM o alguno similar en línea para tener una eficaz administración del proyecto y centralizar la información.

Se recomienda conseguir un financista, dispuesto a invertir en el proyecto para poder aplicar el mismo con una muestra más significativa, y con esto lograr obtener resultados más significativos.

Las pruebas estandarizadas necesitan un mayor tiempo entre el pre-test y el pos-test, por lo que se recomienda realizar las mismas con un tiempo en medio de seis meses por lo menos.

Como recomendación se tiene al ser satisfactorios los resultados, buscar primero ampliar hacia las escuelas rurales, ya que muchas veces se deja fuera y es la gente que más necesitada de educación esta. Teniendo en cuenta que se tiene que traducir el manual de docente como el cuadernillo de trabajo a la lengua materna para que ellos tengan un mejor entendimiento de lo que se les está explicando.

La aprobación para todo el segundo ciclo de primaria, nos referimos a aplicar el proyecto en los grados de cuarto, quinto y sexto primaria de todas las escuelas, tanto urbanas como rurales.

Estudiar la posibilidad de crear nuevas herramientas para grados mas avanzados, como lo es educación media, ya que la preparación en educación nunca termina y si se busca un mejor país, que mejor que prepararlo para su futuro.

Como recomendación, desarrollar un editor de problemas que utilice el mismo lenguaje ABJ (con algunas palabras reservadas) para definir un escenario y una lista de objetivos.

Se recomienda desarrollar un mecanismo de definición reglas semánticas en tiempo de ejecución, con la finalidad de que en el proyecto no solo sea posible la definición de dialectos, sino también de idiomas.

Se recomienda considerar la migración de la vista del sistema a una aplicación Web con el fin de aprovechar las ventajas que proveen los navegadores Web en términos de despliegue de elementos e interacción con el usuario.

Se recomienda agregar las producciones y terminales necesarios para soportar estructuras de control condicionales y cíclicas.

Se recomienda realizar una forma de analizar los problemas ya generados e integrar los sustantivos que se utilizan en este al diccionario del intérprete.

Se recomienda integrar métodos de inteligencia artificial para realizar data-mining de los algoritmos utilizados por varios niños. Así lograrán determinar factores ocultos en la forma en que cada niño resuelve distintos problemas.

Se recomienda agregar animaciones y sonido al ambiente gráfico de la herramienta para un aspecto más profesional que atraiga tanto a los estudiantes y maestros que lo utilizan como a inversionistas, científicos de la computación y educadores nacionales e internacionales.

Se recomienda implementar la opción para restablecer una escena utilizando las acciones guardadas en la bitácora.

Se concluye que es muy importante documentar todo el proceso de desarrollo del Megaproyecto en la Wiki. Por esto mismo se recomienda seguir documentando todas las reuniones, investigaciones y toda actividad en la Wiki del Megaproyecto, con el mayor detalle posible para que futuras generaciones puedan entender el por qué de las decisiones tomadas.

VIII. BIBLIOGRAFÍA

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2007). Compilers principles, techniques & tools. En J. D. Ullman, *Compilers principles, techniques & tools* (págs. 39-47). Boston, United States of America: Pearson Addison Wesley.

Alice Carnegie Mellon. (s.f.). *About Alice.* Recuperado el 25 de septiembre de 2011, de Alice: An Educational Software that teaches students computer programming in a 3D environment: http://www.alice.org/index.php?page=what_is_alice/what_is_alice

Association for the Advancement of Artificial Intelligence. (10 de septiembre de 2011). *Natural Language: Understanding & Generating Text & Speech*. (B. Webber, Productor, & AITopics Editorial Board) Recuperado el 25 de septiembre de 2011, de Association for the Advancement of Artificial Intelligence Web site: http://aaai.org/AITopics/NaturalLanguage

Ausubel, D. P. (s.f.). (Trillás, Ed.) Recuperado el 05 de febrero de 2011, de http://cadel2.uvmnet.edu/portalPLE/asignaturas/mesxxi/contenido/unidad5/psico.pdf

Ausubel, D. P. (2002). (Paidós, Ed.) Recuperado el 5 de febrero de 2011, de http://books.google.com.gt/books?hl=es&lr=&id=VufcU8hc5sYC&oi=fnd&pg=PA13&dq=aprendizaje+significativo+de+ausubel&ots=-5u8Fxx5dz&sig=woInSOuKNflZuHdojT3ryYIklxU#v=onepage&q=aprendizaje%20significativo%20de%20ausubel&f=false

Atorresi, A., Macedo, B., & Leymonié, J. (2009). *Habilidades para la vida.* Chile: UNESCO.

Beas Franco, J., Santa Cruz V., J., Thomsen, P., & Utreras, S. (2005). *Enseñar a pensar para aprender mejor.* México: Alfaomega Grupo Editor, S. A. de C. V.

Beltrán, J. (2002). *Procesos, estrategias y técnicas de aprendizaje.* Madrid: Ed. Síntesis.

Bossidy, L. y. (2008). *El arte de la ejecución en los negocios.* México: Punto de Lectura.

Blank, Leland (2006). Ingeniería Económica. Análisis beneficio / costo y economía del sector público. 6ta Edición. México. 816 págs.

Buendía Eisman, L., Colás Bravo, M. P., Colás Bravo, P., & Hernández Pina, F. (1998). *Métodos de Investigación en psicopedagogía.* Madrid, España: McGraw Hill.

Bronzina, L., Chemello, G., & Agrazar, M. (2009). *Aportes para la enseñanza de la Matemática.* Santiago de Chile: Oficina Regional de Educación de la Organización de la Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) y Laboratorio Latinoamericano de Evaluación de la Calidad Educativa (LLECE).

Burgos, M. (s.f.). *Formalismo y Funcionalismo: Concepciones teóricas y metodológicas divergentes dentro de la lingüística*. Recuperado el 25 de septiembre de 2011, de Distrito Federal 24: http://distritofederal24.com/manuelburgos/Manuel_Burgos/INTERESTS_files/FormalismoFuncionalismo.pdf

Bruner, J. S. (1972). *Hacia una Teoría de la Instrucción.* Mexico: Unión Tipográfica Editorial Hispanoamericana.

Caldeiro, G. P. (2005). *Lev Semenovich Vigotsky* . Recuperado el 18 de septiembre de 2011, de http://vigotsky.idoneos.com/index.php/293538

Cattaneo, M. (s.f.). *Universidad de Palermo.* Recuperado el 18 de septiembre de 2011, de http://www.palermo.edu/ingenieria/downloads/Investigacion/211105MCattaneo.pdf

Center for Natural Language Processing. (s.f.). *Natural Language Processing.*Recuperado el 25 de septiembre de 2011, de Center for Natural Language Processing Research Web site: http://www.cnlp.org/publications/03nlp.lis.encyclopedia.pdf

CIEN y PREAL. (2008). *Educación: un desafío de urgencia nacional. Informe de progreso educativo, Guatemala.* GUATEMALA: CIEN y PREAL.

Comellas, M. J. (2002). *Los hábitos de autonomía. Proceso de adquisición.* Barcelona: Grupo Editorial Ceac, S. A.

Comisión Paritaria de Reforma Educativa. (1998). *Diseño de Reforma Educativa.* Guatemala.

Constitución Política de la República de Guatemala. (1985). Guatemala.

Coulter, R. (2010). *Administración* (10 ed., Vol. 10). Mexico: Pearson Educación.

DevinCook. (s.f.). *GOLD Parsing System*. Recuperado el 25 de septiembre de 2011, de Devin Cook Web site: http://www.devincook.com/goldparser/

Diaz-Barriga Arceo, F., & Hernández Rojas, G. (2002). *Estrategias docentes para un aprendizaje significativo. Una interpretación constructivista.* México: Mc Graw Hill.

Dirección de Calidad y Desarrollo Educativo (DICADE). (2006). *Curriculum Nacional Base para la formacion inicial de docentes del nivel primario.* Guatemala: Ministerio de Educación (MINEDUC).

Eduteka. (s.f.). Recuperado el 15 de septiembre de 2011, de HYPERLINK "http://www.eduteka.org/TaxonomiaBloomCuadro.php3" http://www.eduteka.org/TaxonomiaBloomCuadro.php3

*Eduteka (s.*f.) Recuperado el 15 de septiembre de 2011, de HYPERLINK "http://www.eduteka.org/modulos/9" http://www.eduteka.org/modulos/9

Estándares de Guatemala. (s.f.). *Investigación Nacional*. Recuperado el 25 de septiembre de 2011, de Estándares de Guatemala: http://www.estandaresdeguatemala.org/images/noticias/Bolet%C3%ADn-espa%C3%B1ol.pdf

Estándares de Guatemala. (s.f.). *Investigación Nacional Competencias Básicas para la Vida*. Recuperado el 25 de septiembre de 2011, de Estándares de Guatemala: http://www.estandaresdeguatemala.org/images/noticias/Investigacion%20Nacional%20final%20web-LR.pdf

Eytam Computer Science. (s.f.). *Educational Programming Language.*Recuperado el 25 de septiembre de 2011, de Eytam Computer Science Web site: http://www.eytam.com/mama/educational_programming_language

Fieldman, R. S. (2007). *Desarrollo psicológico a través de la vida.* México: Pearson Education.

Functional Grammar. (29 de enero de 2009). *Welcome*. Recuperado el 25 de septiembre de 2011, de Functional Grammar Web site: http://www.functionalgrammar.com/

Gálvez-Sobral, L. A. (2010). *La influencia de las características docentes sobre el rendimiento de los alumnos en primaria. Comparación en muestras de dos años y exploración multinivel.* Guatemala: DIGEDUCA, Mineduc.

García, C. (04 de septiembre de 2007). Claro y Conciso: Mi vieja Osborne y la bella Antigua. *Prensa Libre* .

Gillespie, C. (1966). *Contabilidad y control de costos* (6 ed.). Mexico: Editorial Diana.

Gutiérrez, R. (s.f.). Recuperado el 05 de febrero de 2011, de http://www.raco.cat/index.php/ensenanza/article/viewFile/50960/92902

Heng, C. (2009). *Educational Programming Languages.* Recuperado el 25 de septiembre de 2011, de The free country Web site: http://www.thefreecountry.com/compilers/educational-programming-languages.shtml

Hernández Pina, F. (2001). Métodos de Investigación Psicopedagógica. En E. P. Buendía. México: McGraw Hill.

Junauza. (10 de febrero de 2009). *Teaching kids programming with free.*Recuperado el 25 de septiembre de 2011, de Junauza Web site: http://www.junauza.com/2009/10/teaching-kids-programming-with-free.html

Laza, S. (24 de Abril de 2006). *Zona Económica*. Recuperado el 5 de Abril de 2011, de Adam Smith: http://www.zonaeconomica.com/adamsmith

López García, J. C. (2009). *Eduteka.org.* Recuperado el 8 de septiembre de 2011, de http://www.eduteka.org/GuiaAlgoritmos.php

Martínez, F. M. (19 de Diciembre de 2010). El modelo de formación caducó. *Prensa Libre* .

Mayer, R. E. (1982). The pshychology of mathematical problem solving. En F. J. Lester, *Mathematical Problem Solving* (págs. 1 - 13). Philadelphia: The Frankilin Institute Press.

McKinsey & Company. (09 de 2007). *Eduteka.org.* Recuperado el 15 de 07 de 2011, de Eduteka.org: http://www.eduteka.org/pdfdir/McKENSEY_InformeReformaEducativa.pdf

Ministerio de Educación de Guatemala. (s.f.). Recuperado el 15 de julio de 2010, de : http://www.mineduc.gob.gt/estadistica/2008/main.html

Ministerio de Educación de Chile. (1 de Enero de 2010). *Nota técnica: Matriz de Margo Lógico*. Recuperado el 12 de Enero de 2011, de Matriz de Marco Lógico: http://www.mineduc.cl/biblio/documento/Documento_Matriz_de_Marco_Logico.pdf

Ministerio de Educación de Chile. (1 de Enero de 2010). *Nota técnica: Matriz de Margo Lógico*. Recuperado el 12 de Enero de 2011, de Matriz de Marco Lógico: http://www.mineduc.cl/biblio/documento/Documento_Matriz_de_Marco_Logico.pdf

Ministerio de Educación de la Nación Argentina. (s.f.). *La gramática generativa (Chomsky)*. Recuperado el 25 de septiembre de 2011, de Educ Ar: Portal educativo del Estado argentino: http://aportes.educ.ar/lengua/nucleo-teorico/recorrido-historico/-la-lingueistica-del-siglo-xx/la_gramatica_generativa_chomsk.php

Montgomery, Douglas (2007). Probabilidad y estadística aplicadas a la Ingeniería. Población y muestras aleatorias. 2da Edición. México. 817 págs.

Natural Language Toolkit. (s.f.). *NLTK Home*. Recuperado el 25 de septiembre de 2011, de Natural Language Toolkit Web site: http://www.nltk.org/

Nerici, I. G. (1990). *Metodología de la enseñanza.* México: Kapeluz Mexicana.

Nieto Said, J. H. (26 de Julio de 2004). *Comité Estatal de Olimpiadas de Matemáticas.* Recuperado el 10 de Agosto de 2009, de http://ommcolima.ucol.mx/guias/TallerdeResolucionproblemas.pdf

Otis, A. y. (1969). *Manual Técnico Test Otis-Lennon de Habilidad Mental.*Harcourt, Brace & World, Inc.

Paiz, S. (Agosto de 2010). Continuidad de Políticas Educativas. (E. p. educación, Ed.) Guatemala, Guatemala.

Papalia, D. E. (1986). *Psicología del desarrollo.*Colombia: McGraw-Hill.

Polya, G. (1971). *How to solve it. A new aspects of mathematical method.*Princeton: Princeton University Press.

PREAL. (1998). *Organización de Estados Iberoamericanos.* Recuperado el 30 de septiembre de 2011, de http://www.oei.es/quipu/Informe_preal1998.pdf

Quiñónez, A. (2011). *Evaluar, un aporte para mejorar la calidad educativa.* Ministerio de Educación, Dirección General de Evaluación e Investigación Educativa, Guatemala.

Quiñonez, A., Del Valle, M. J., Castellanos, J., Aguilar, M., Johnson, J., Flores, M., y otros. (2010). *Serie de Cuadernillos Pedagógicos: De la Evaluación a la Acción. Matemáticas:*

Resolución de Problemas. Guatemala: Dirección General de Evaluación e Investigación Educativa, Ministerio de Educación.

Resnick, L. y. (1981). *The Psychology of Matematics for Instruction.*Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Riego Gaona, M. A. (1986). *Teoría de Jean Piaget acerca del desarrollo cognoscitivo del niño y su relación con el aprendizaje*. Recuperado el 25 de septiembre de 2011, de Sitio web de la Universidad Autónoma Nacional de México: http://www.bibliodgsca.unam.mx/tesis/tes15marg/sec_1.htm

Rodríguez, M. E. (15 de 12 de 2010). *Repositorio de las revistas científicas de la Universidad de Costa Rica.*

www.latindex.ucr.ac.cr. Recuperado el 25 de 08 de 2011, de latindex.ucr.ac.cr: http://www.latindex.ucr.ac.cr/aie-2010-03/actuainv-10-3-2000-17.pdf

Runger, G. (2007). *Probabilidad y estadística aplicada a la ingeniería* (2 ed.). Mexico: Limusa Wiley.

Sakharov, A. (s.f.). *Formal Language.* Recuperado el 25 de septiembre de 2011, de MathWorld - A Wolfram Web Resource: http://mathworld.wolfram.com/FormalLanguage.html

Scratch MIT. (s.f.). *About Scratch*. Recuperado el 25 de septiembre de 2011, de MIT Scratch Web site: http://info.scratch.mit.edu/About_Scratch

Stanford. (s.f.). *Parsing with YAPPS*. Recuperado el 25 de septiembre de 2011, de Theory Standford Web site: http://theory.stanford.edu/~amitp/yapps/

Taha, H. (2004). *Investigación de operaciones* (7 ed.). Mexico: Pearson Educación.

The George Washington University. (s.f.). *Programming Paradigms*. Recuperado el 25 de septiembre de 2011, de The George Washington University Web site: http://www.seas.gwu.edu/~bell/csci210/lectures/programming_paradigms.pdf

UNESCO. (2010). *Factores asociados al logro cognitivo de los estudiantes de América Latina y e Caribe.* Chile: UNESCO.

UNESCO. (s.f.). Recuperado el 6 de febrero de 2010, de http://www.unesco.org.uy/educacion/es/areas-de-trabajo/educacion/temas/educacion-diversidad-e-inclusion.html

Universidad Veracruzana. (s.f.). Recuperado el 15 de septiembre de 2011, de http://www.uv.mx/dgda/afbg/estudiantes/documents/C1.pdf

W3. (19 de enero de 2005). *DOM*. Recuperado el 25 de septiembre de 2011, de W3: http://www.w3.org/DOM/

W3 Schools. (s.f.). *What is XML?*Recuperado el 25 de septiembre de 2011, de W3 Schools Web site: http://www.w3schools.com/xml/xml_whatis.asp

Wikibin. (2007). *sEnglish.* Recuperado el 25 de septiembre de 2011, de Wikibin Web site: http://wikibin.org/articles/senglish.html

Wikipedia. (22 de septiembre de 2011). *Model-view-controller*. Recuperado el 25 de septiembre de 2011, de Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

Wikipedia. (23 de septiembre de 2011). *Sintagma*. Recuperado el 25 de septiembre de 2011, de Wikipedia, la enciclopedia libre: http://es.wikipedia.org/wiki/Sintagma

IX. APÉNDICE

A. Seguimiento de la aplicación de la herramienta

Docente: _____ Plantel: _____

Fecha:_____ Hecho por: _____

1. ¿Cuántos algoritmos aplicó durante la presente semana? _____

2. ¿Verificó que los pasos fueran individuales, es decir que no se pone más de una acción por cada paso?

_____

3. ¿Verificó que los pasos fueran en presente indicativo? _____

4. ¿Verificó que los pasos fueran específicos, es decir, no se debe utilizar palabras como "un poco, algunos, etc."? _____

5. ¿Cuánto tiempo (promedio) le lleva la aplicación de cada algoritmo? _____

6. ¿Cuál es la actitud de sus alumnos hacia la aplicación de algoritmos? (indique todas las opciones que sea aplicables

   a. Contentos
   b. Aburridos
   c. Entusiasmo
   d. Poco interés
   e. Otros _____

7. ¿Qué dificultades ha tenido en la aplicación de la herramienta no computarizada?

_____

¿y la computarizada?_____

8. ¿Ha aplicado un algoritmo en una materia o tema que esté enseñando? ¿Cuál?

_____¿Cómo? _____

¿Qué observó?_____

_____

9. ¿Ha ido incrementando la dificultad de los algoritmos _____? ¿Cómo han respondido los alumnos a esa dificultad?

_____

_____

10. ¿Han propuesto sus alumnos algún algoritmo distinto a los indicados en el cuadernillo y manual del docente? _____ ¿Cuáles?

_____

_____

_____

11. ¿Podemos tomar fotografías de algunos de los trabajos de los alumnos?

Cuadro 22. Costo de capacitaciones y revisiones.

Capacitación Docente y revisión de la Herramienta		
Presupuesto	**Guatemala**	4 días
Alimentación		Q 140.00
Hospedaje		Q -
Transporte		Q 160.00
Artículos de oficina		Q 50.00
Total		**Q 350.00**
Presupuesto	**Progreso**	1 día
Alimentación		Q 35.00
Hospedaje		Q -
Transporte		Q 140.00
Artículos de oficina		Q 50.00
Total		**Q 225.00**
Presupuesto	**Sacatepéquez**	1 día
Alimentación		Q 35.00
Hospedaje		Q -
Transporte		Q 70.00
Artículos de oficina		Q 50.00
Total		**Q 155.00**
Presupuesto	Chimaltenango	1 día
Alimentación		Q 35.00
Hospedaje		Q -
Transporte		Q 105.00
Artículos de oficina		Q 50.00
Total		**Q 190.00**
Presupuesto	Escuintla	1 día
Alimentación		Q 35.00
Hospedaje		Q -
Transporte		Q 105.00
Artículos de oficina		Q 50.00
Total		**Q 190.00**

Continuación Cuadro 22

Presupuesto	Santa Rosa	1 día
Alimentación		Q 35.00
Hospedaje		Q -
Transporte		Q 125.00
Artículos de oficina		Q 50.00
Total		**Q 210.00**
Presupuesto	Sololá	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 245.00
Artículos de oficina		Q 50.00
Total		**Q 365.00**
Presupuesto	Totonicapán	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 355.00
Artículos de oficina		Q 50.00
Total		**Q 475.00**
Presupuesto	Quetzaltenango	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 355.00
Artículos de oficina		Q 50.00
Total		**Q 915.00**
Presupuesto	Suchitepéquez	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 295.00
Artículos de oficina		Q 50.00
Total		**Q 855.00**

Continuación Cuadro 22

Presupuesto	Retalhuleu	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 340.00
Artículos de oficina		Q 50.00
Total		**Q 460.00**
Presupuesto	San Marcos	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 440.00
Artículos de oficina		Q 50.00
Total		**Q 1,000.00**
Presupuesto	Huehuetenango	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 470.00
Artículos de oficina		Q 50.00
Total		**Q 1,030.00**
Presupuesto	Quiché	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 290.00
Artículos de oficina		Q 50.00
Total		**Q 410.00**
Presupuesto	Baja Verapaz	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 270.00
Artículos de oficina		Q 50.00
Total		**Q 390.00**

Continuación Cuadro 22

Presupuesto	Alta Verapaz	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 390.00
Artículos de oficina		Q 50.00
Total		**Q 950.00**
Presupuesto	Petén	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 890.00
Artículos de oficina		Q 50.00
Total		**Q 1,450.00**
Presupuesto	Izabal	2 días
Alimentación		Q 210.00
Hospedaje		Q 300.00
Transporte		Q 540.00
Artículos de oficina		Q 50.00
Total		**Q 1,100.00**
Presupuesto	Zacapa	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 275.00
Artículos de oficina		Q 50.00
Total		**Q 395.00**
Presupuesto	Chiquimula	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 310.00
Artículos de oficina		Q 50.00
Total		**Q 430.00**

Continuación Cuadro 22

Presupuesto	Jalapa	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 180.00
Artículos de oficina		Q 50.00
Total		**Q 300.00**
Presupuesto	Jutiapa	1 día
Alimentación		Q 70.00
Hospedaje		Q -
Transporte		Q 220.00
Artículos de oficina		Q 50.00
Total		**Q 340.00**

X. GLOSARIO

- **Algoritmo:** conjunto de pasos ordenados planteados para resolver un problema.
- **Analizador léxico / scanner:** fase del proceso de traducción de lenguajes en la que se toma una hilera de caracteres escrita en el lenguaje fuente y se divide en bloques pequeños (tokens) a partir de una definición de expresiones regulares.
- **Analizador semántico / ejecución semántica:** fase del proceso de traducción de lenguajes en la que se evalúa la congruencia (sentido) del comando a traducir o ejecutar.
- **Analizador sintáctico / parser:** fase del proceso de traducción de lenguajes en la que se trata de asignar (derivar) una estructura gramatical a la hilera de tokens devuelta por el scanner.
- **Árbol sintáctico:** estructura definida por un conjunto de producciones que es asignada a una hilera de tokens durante la fase de análisis sintáctico.
- **Clase:** definición abstracta del dominio del estado y el comportamiento de un objeto (tipo).
- **Competencia:** conjunto de habilidades humanas útiles para el desempeño adecuado de un humano en sociedad, en la academia y en el trabajo.
- **Compilación (compilador):** proceso de traducción en la que se considera en un solo paso el conjunto completo de instrucciones en el código fuente.
- **Comportamiento:** conjunto de métodos provistos por un objeto.
- **Desligar:** separar la unión entre cualesquiera dos sustantivos (referente al paradigma Dirigido por la Historia).
- **Dialecto:** conjunto de terminales alternativos (o complementarios) que aplican sobre una gramática libre de contexto del lenguaje ABJ.
- **DOM:** DocumentObjectModel, definición de cómo acceder y utilizar la definición de documentos escritos en lenguaje de etiquetas.
- **Drag & drop (Arrastrar y soltar):** es una expresión informática que se refiere a la acción de mover con el ratón objetos de una ventana a otra o entre partes de una misma ventana.
- **Epsilon:** hilera de caracteres nula (vacía).
- **Escenario:** representación inicial de un problema en el paradigma Dirigido por la Historia. Es el análogo a una tabla de símbolos en paradigmas más comunes de programación.
- **Estado:** conjunto de valores de las variables de instancia de un objeto en un momento dado de la ejecución de un programa.
- **Estudio de benchmark:** análisis de la "competencia". Para fines del presente proyecto, se refiere más bien al análisis de proyectos similares al megaproyecto propuesto.

- **Evento:** suceso en el sistema operativo reportado al entorno de ejecución para ser manejado y producir una consecuencia en código.
- **Expresión regular:** patrón que define un conjunto de hileras de caracteres basado en su estructura a nivel caracter.
- **Gramática libre de contexto:** conjunto de reglas que definen sin lugar a ambigüedad la estructura de un lenguaje formal.
- **Herencia:** relación entre dos objetos en la programación orientada a objetos, en la que A hereda de B, si A comparte el estado y comportamiento de B.
- **Héroe:** representante "solucionador" de un problema en el contexto del paradigma Dirigido por la Historia.
- **IDE:** Ambiente de Desarrollo Integrado. Ambiente de software utilizado para desarrollar soluciones de software.
- **Instancia:** ejemplo de una clase representado por un objeto.
- **Inteligencia Artificial:** rama de la computación dedicada a la creación de agentes racionales.
- **Interfaz (OO):** conjunto de métodos públicos puestos a disposición por un objeto o por una jerarquía de implementación.
- **Interfaz gráfica (GUI):** parte visible del software desarrollado. Se utiliza para comunicarse con el usuario del sistema.
- **Interpretación (intérprete):** proceso de traducción en el que se aplican todas las fases instrucción por instrucción.
- **Keyword:** palabra reservada en un lenguaje. Propone una limitante en el esquema de nombramiento de métodos y variables.
- **Lenguaje formal:** conjunto bien definido de palabras.
- **Lexema:** valor actual de un token.
- **Ligar:** unión entre cualesquiera dos sustantivos (referente al paradigma Dirigido por la Historia).
- **Log:** bitácora de acciones realizadas registradas como pasos en la resolución de un problema de forma algorítmica.
- **Megaproyecto:** modelo de graduación propuesto por la Universidad del Valle de Guatemala en el que se propone un proyecto a gran escala interdisciplinario.
- **Método:** función que recibe un conjunto de parámetros con los cuales realiza un proceso. Los métodos son un tipo especial de funciones en los que se entiende que está relacionado con el comportamiento de un objeto.

- **Mochila:** contenedor perteneciente al héroe (en el contexto del paradigma Dirigido por la Historia) en el que se pueden colocar objetos necesarios para ejecutar cada paso de un problema.
- **Mockup:** modelo o diseño utilizado para enseñar, demostrar, evaluar diseño, promocionar y más. Un mockup toma el nombre "prototipo" si provee por lo menos, parte de la funcionalidad del sistema y permite probar el diseño.
- **MVC:** arquitectura o patrón de diseño Modelo Vista Controlador, en el que se propone una separación de tareas entre distintos módulos de un sistema de software (lógica y visual).
- **NLTK:** Natural LanguageToolkit, librería de Python que facilita el procesamiento de lenguajes formales. Provee bastantes facilidades para definir scanners y parsers en tiempo de ejecución.
- **Objeto:** estructura que posee estado y comportamiento. Son los bloques básicos de un programa escrito bajo el paradigma de la programación orientada a objetos.
- **Paradigma de programación:** un marco de referencia ampliamente adoptado que trata de definir la forma de programar estándar, para resolver varios problemas bien definidos.
- **Producción:** regla que define una serie de hileras de caracteres (comandos u oraciones) pertenecientes a un lenguaje formal.
- **Prueba de caja negra:** Black box testing. Pruebas llevadas a cabo sobre la interfaz de software actuando sobre ella como una caja negra, proporcionando entradas y estudiando las salidas para ver si son las esperadas.
- **Prueba de usabilidad:** proceso de evaluación la interacción de un usuario con un producto (en el ámbito del presente trabajo, se refiere al software desarrollado).
- **Prueba estandarizada:** examen estadísticamente corroborado. Son desarrolladas por equipos de estadistas, psicólogos y expertos en el tema de evaluación.
- **Prueba Otis Lennon:** prueba estandarizada de medición de habilidades generales.
- **Python:** lenguaje de programación multiparadigma y multiplataforma.
- **Recursión:** técnica de programación en la que la estructura de una función se define en términos de la misma función.
- **SVG:** lenguaje de definición de imágenes vectoriales basado en el estándar XML.
- **Tabla de símbolos / sustantivos:** estructura de control en el proceso de traducción utilizada para la resolución de ámbito y nombre de variables, objetos y métodos.
- **Teoría del desarrollo intelectual de Piaget:** teoría en la que se propone un modelo que describe la forma en que el humano desarrolla las habilidades mentales según su edad.

- **Token:** estructura que contiene un identificador, un lexema y una regla de asignación que determina por qué se relaciona un lexema con dicho identificador. Generalmente estas reglas son definidas a través de expresiones regulares.
- **XHTML:** lenguaje estándar de definición de documentos de hipertexto, definido en base al estándar XML.
- **XML:** Extensible MarkupLanguage. Metalenguaje que permite la definición de lenguajes basados en un sistema de etiquetado.