

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



Análisis y desarrollo del Sistema de Rocas

Trabajo de graduación presentado por Karen Andrea Tojín

para optar al grado académico de

Licenciada en Ingeniería en Ciencias de la Computación

Guatemala

2013

Análisis y desarrollo del Sistema de Rocas

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



Análisis y desarrollo del Sistema de Rocas

Trabajo de graduación presentado por Karen Andrea Tojín
para optar al grado académico de
Licenciada en Ingeniería en Ciencias de la Computación

Guatemala

2013

Vo.Bo.

(f) 

Ing. Jenniffer Guzmán


Tribunal

(f) 

MSc. Douglas Barrios

(f) 

Ing. Lynette García

(f) 

Ing. Jenniffer Guzmán

Fecha de Aprobación: 17 de junio de 2013

PREFACIO

El desarrollo del sistema de rocas surge de la necesidad de mejorar el control sobre las metas de la empresa *EBclosion*. Dichas metas son llamadas rocas, y surgen de la metodología del libro "*Mastering the Rockefeller Habits*", publicado por Verne Harnish en el 2002. Con el desarrollo de este sistema se busca resolver el problema de la empresa *EBclosion* de mejorar un sistema actual para llevar el control de las rocas.

ÍNDICE

	Página
PREFACIO	v
ÍNDICE	vi
LISTA DE FIGURAS	x
LISTA DE TABLAS	xii
RESUMEN	xiii

Capítulos

I. INTRODUCCIÓN	1
II. OBJETIVOS	2
A. Objetivo general	2
B. Objetivos específicos	2
III. MARCO TEÓRICO	3
A. Rocas	3
1. Concepto	3
2. <i>Mastering The Rockefeller Habits de Verne Harnish</i>	3
3. Propósito	3
B. Metodología y desarrollo	4
1. Metodología de desarrollo de sistemas dinámicos (DSDM)	4
2. Elección de la metodología	5
3. Conceptos clave de la metodología	5
4. Proceso de desarrollo de DSDM	6

5. Roles de DSDM	8
6. Herramientas y técnicas	9
C. <i>Symfony Framework</i>	11
1. <i>Framework</i>	11
2. Arquitectura Modelo Vista Controlador	11
3. <i>Object Relational Mapping (ORM)</i>	13
IV. ANÁLISIS DEL SISTEMA (FASE 1)	14
A. Análisis del negocio	14
B. Análisis de factibilidad	16
1. Evaluación técnica	17
2. Evaluación financiera	17
C. Versiones	18
1. Versión 0 – Hoja de cálculo	18
2. Versión 1 – Herramienta en línea	19
D. Análisis de requerimientos	24
V. DISEÑO DEL SISTEMA (FASE 2)	27
A. Diseño de la aplicación	27
B. Diagramas UML	27
1. Diagrama de casos de uso	27
2. Diagrama de secuencia	28
3. Priorización de MoSCoW	30
4. Esquema de datos	32
5. Prototipos	33
VI. IMPLEMENTACIÓN (FASE 3 A 5)	37

A.	Iteración 1	37
	1. Diseño gráfico del sistema	37
	2. Creación del proyecto en <i>Symfony</i> y maquetado del sitio	38
	3. Creación de la base de datos del sistema	39
	4. Conclusiones de la iteración	41
B.	Iteración 2	42
	1. Ingreso con el correo y contraseña de la empresa	42
	2. Creación, edición y eliminación de rocas	42
	3. Conclusiones de la iteración	43
C.	Iteración 3	43
	1. Cálculo de avances	43
	2. Conclusiones de la iteración	45
D.	Iteración 4	45
	1. Módulo de usuarios	45
	2. Módulo de unidades y departamentos	46
	3. Módulo de trimestres	47
	4. Módulo de EBRocas	48
	5. Conclusiones de la iteración	49
E.	Iteración 5	49
	1. Sección de reportes	50
	2. Manuales del sistema	51
	3. Conclusiones de la iteración	51
VII.	DESPLIEGUE, PRUEBAS Y MANTENIMIENTO (FASE 6 Y 7)	52
A.	Despliegue	52

B.	Pruebas	52
1.	Pruebas unitarias	52
2.	Pruebas funcionales	53
C.	Mantenimiento	53
VIII.	IMPACTO EMPRESARIAL	54
A.	Desarrollo futuro	54
IX.	CONCLUSIONES	55
X.	RECOMENDACIONES	56
XI.	BIBLIOGRAFÍA	57
XII.	GLOSARIO	59
XIII.	ANEXOS	61

LISTA DE FIGURAS

	Página
Figura no. 1: Enfoques del desarrollo de proyectos	4
Figura no. 2: Fases de DSDM	7
Figura no. 3: Tipos de prototipos	10
Figura no. 4: Arquitectura modelo, vista, controlador	12
Figura no. 5: <i>Object Relational Mapper</i>	13
Figura no. 6: Sistema de rocas en hoja de cálculo de <i>Google Docs</i>	19
Figura no. 7: Esquema de la base de datos de la primera versión del sistema	20
Figura no. 8: Diseño inicial de la primera versión del sistema de rocas	21
Figura no. 9: Diseño modificado de la primera versión del sistema	22
Figura no. 10: Diseño final de la primera versión del sistema	23
Figura no. 11: Diagrama de casos de uso del sistema	28
Figura no. 12: Diagrama de secuencia de la creación de rocas para un nuevo trimestre	29
Figura no. 13: Diagrama de secuencia de la actualización de rocas	29
Figura no. 14: Diagrama de clases que representa el esquema de datos del proyecto	33
Figura no. 15: Pantalla de ingreso	34
Figura no. 16: Pantalla de inicio de la aplicación, visualización de rocas	34

Figura no. 17: Área de ingreso de rocas	35
Figura no. 18: Sección de administración de usuarios. Esta vista sería utilizada para la administración completa del sistema	35
Figura no. 19: Sección de reportes de avances	36
Figura no. 20: Sección de reportes de cierre semanal	36
Figura no. 21: Diseño gráfico aprobado para el sistema de rocas	38
Figura no. 22: Diagrama entidad relación de la base de datos	39
Figura no. 23: Ejemplo de unidades y departamentos dentro de la empresa	40
Figura no. 24: Diagrama de base de datos	41
Figura no. 25: Ingreso de rocas del sistema de <i>EBclosion</i>	43
Figura no. 26: Cálculo de avance de rocas	44
Figura no. 27: Módulo de usuarios	46
Figura no. 28: Módulo de unidades y trimestres	47
Figura no. 29: Módulo de trimestres	48
Figura no. 30: Módulo de EBRocas	49
Figura no. 31: Reportes de porcentaje de avance	50
Figura no. 32: Reportes de cierre semanal	51

LISTA DE TABLAS

	Página
Tabla no. 1: Priorización de MoSCoW en la primera iteración	30
Tabla no. 2: Priorización de MoSCoW en la segunda iteración	31
Tabla no. 3: Priorización de MoSCoW en la tercera iteración	31
Tabla no. 4: Priorización de MoSCoW en la cuarta iteración	32
Tabla no. 5: Priorización de MoSCoW en la quinta iteración	32
Tabla no. 6: Historia de usuario no. 1 – Creación de rocas de primer y segundo nivel	61
Tabla no. 7: Historia de usuario no. 2 – Creación de rocas de tercer y cuarto nivel	62
Tabla no. 8: Historia de usuario no. 3 – Cálculo de avances	62
Tabla no. 9: Historia de usuario no. 4 – CRUD de unidades	63
Tabla no. 10: Historia de usuario no. 5 – CRUD de usuarios	63
Tabla no. 11: Historia de usuario no. 6 – CRUD de trimestres	64
Tabla no. 12: Historia de usuario no. 7 – CRUD de EBRocas	64
Tabla no. 13: Historia de usuario no. 8 – Sección de reportes	65

RESUMEN

El Sistema de Rocas desarrollado para la empresa *EBclosion S.A.* busca proveer una solución que solvante el problema del registro de avance de las metas de la empresa, dichas metas se llaman rocas. El sistema también busca brindar reportes que sirvan como guías de evaluación de los empleados para los directivos de la empresa.

El concepto de roca se refiere a una meta o propósito que se plantea periódicamente para toda la empresa, para cada departamento y para cada persona que labora en ella. El propósito de una roca es tener registro de los avances en aspectos específicos que desean evaluarse en el rendimiento de la empresa. El concepto de rocas proviene del libro "*Mastering the Rockefeller Habits*", publicado por Verne Harnish en el 2002.

El sistema de rocas es una aplicación web desarrollada en *Symfony 1.4, framework* que utiliza la empresa para el desarrollo de sus aplicaciones. Fue creado para llevar el control de las rocas de la empresa, clasificándolas por rocas personales, rocas de unidad y rocas de toda la empresa (*EBRocas*). Cada persona que labora en la empresa está asociada a una unidad o departamento de trabajo, por lo que todas las rocas personales alimentan a dicha unidad, y a su vez, cada una de las rocas de las unidades está asociada a una roca de la empresa, haciendo un sistema piramidal.

El presente trabajo describirá el proceso de desarrollo del Sistema de Rocas, desde la recolección de los requerimientos del sistema, su planeación, diseño, desarrollo e implementación, hasta las pruebas de funcionalidad realizadas una vez que el sistema fue implementado y utilizado.

I. INTRODUCCIÓN

Como requisito de graduación se debe realizar una práctica profesional que conste de 400 horas, realizadas en una empresa. En esta práctica profesional debe realizarse un proyecto que solucione un problema planteado por la empresa en la que se realicen las prácticas. Durante las 400 horas se deben poner en práctica todos los conocimientos y buenas prácticas adquiridos durante la carrera de Ingeniería en Ciencias de la Computación.

Se tuvo la oportunidad de realizar la práctica profesional en la empresa *EBclosion*. Esta empresa se dedica al desarrollo de soluciones tecnológicas adaptadas a sus clientes. Entre su catálogo de productos se encuentran plataformas web, aplicaciones móviles y una especialización en servicios SMS, siendo uno de los mayores proveedores de contenido móvil en Centro América.

EBclosion es una empresa con valores inmersos en todas las personas que laboran en ella, y entre estos valores y buenas prácticas se encuentra la evaluación por metas de trabajo. Dichas metas son llamadas rocas. Utilizando la metodología de las rocas, *EBclosion* es capaz de evaluar el rendimiento de todos sus empleados. Sin embargo, esta metodología no estaba siendo implementada a su potencial debido a que no se contaba con la herramienta necesaria para lograrlo.

De esta evaluación surge la necesidad de un sistema de rocas, por lo que se planteó una reestructuración y mejora del sistema que se tenía actualmente. Entre los cambios que fueron planteados, lo más importante era la autonomía del nuevo sistema, ya que el actual se encontraba inmerso con el ERP de la empresa. Se propuso también una mejora en la forma de almacenamiento de los datos del sistema, ya que la versión actual carecía de optimizaciones, lo que hacía que el sistema respondiera de forma lenta.

El presente trabajo describe el proyecto propuesto y desarrollado con el fin de mitigar el problema del sistema de rocas ineficiente con el que contaba la empresa. El trabajo realizado para desarrollar el sistema propuesto consistió en un análisis, que incluyó un estudio de factibilidad y una toma de requerimientos del sistema. Luego del análisis, fue realizado el diseño del sistema, que incluyó la elaboración del diseño gráfico, el esquema de datos y la realización de prototipos. Por último fue realizada la implementación, en iteraciones, de lo que se había diseñado.

El trabajo fue realizado utilizando una metodología llamada DSDM, esta metodología se centra en entregar proyectos de calidad cuando se tiene un tiempo limitado o establecido, como fue el caso de este proyecto debido a las horas de práctica profesional que debían cumplirse.

II. OBJETIVOS

A. OBJETIVO GENERAL

Crear un sistema que le permita a la empresa *EBclosion* S.A. llevar el control de los avances en las metas propuestas trimestralmente para cada unidad que conforma a la empresa.

B. OBJETIVOS ESPECÍFICOS

1. Crear un sistema amigable y eficiente que permita el registro de las rocas de la empresa.
2. Visualizar únicamente lo pertinente a cada unidad de trabajo, incluyendo únicamente el avance de los miembros de la misma.
3. Permitir el ingreso y actualización de avance de rocas a cada miembro de la empresa.
4. Garantizar la estandarización en la forma de medir el progreso de cada unidad de trabajo.
5. Proveer un sistema de reportes que permita evaluar el progreso semanalmente.

III. MARCO TEÓRICO

A. ROCAS

1. Concepto. Según el diccionario de la Real Academia Española del 2013, una roca es una objeto firme y constante. Siguiendo esta idea, una roca es una meta o propósito que debe cumplirse dentro de la empresa en un tiempo estipulado de un trimestre. Dicha meta apoya al desarrollo de la empresa en diferentes aspectos, como automatización, evolución, expansión, cultura y aprendizaje.

En cada división de la empresa, llamada unidad, el líder del equipo define trimestralmente un conjunto de rocas para dicha unidad que deberán ser trabajadas durante el trimestre en curso. Cada una de la rocas definidas para la unidad debe apoyar a cada una de las rocas de la empresa.

2. *Mastering The Rockefeller Habits* de Verne Harnish. “*Mastering the Rockefeller Habits*” es un libro escrito por Verne Harnish en el 2002. Harnish es un autor, consultor, emprendedor y fundó la *Entrepreneurs’ Organization* (EO) en 1987. Harnish siempre ha sido un emprendedor, hasta llegar a escribir para revistas de renombre internacional como *Fortune* y *Money*.

En su libro, *Mastering The Rockefeller Habits*, Harnish describe una roca como un pilar para la organización; una meta que debe ser trabajada y alcanzada y que le da valor a la empresa. Esta meta no debe ser parte de la operación diaria, sino debe ser algo nuevo y de valor para el desarrollo de la empresa.

3. Propósito. El propósito de una roca es listar las metas planteadas para la empresa y para cada unidad de la misma periódicamente. Sin embargo, el sistema de rocas desarrollado va más allá. El propósito del sistema es llevar el control de los avances de las rocas de cada unidad de la empresa, desplegando los avances de una forma comprensible y amigable.

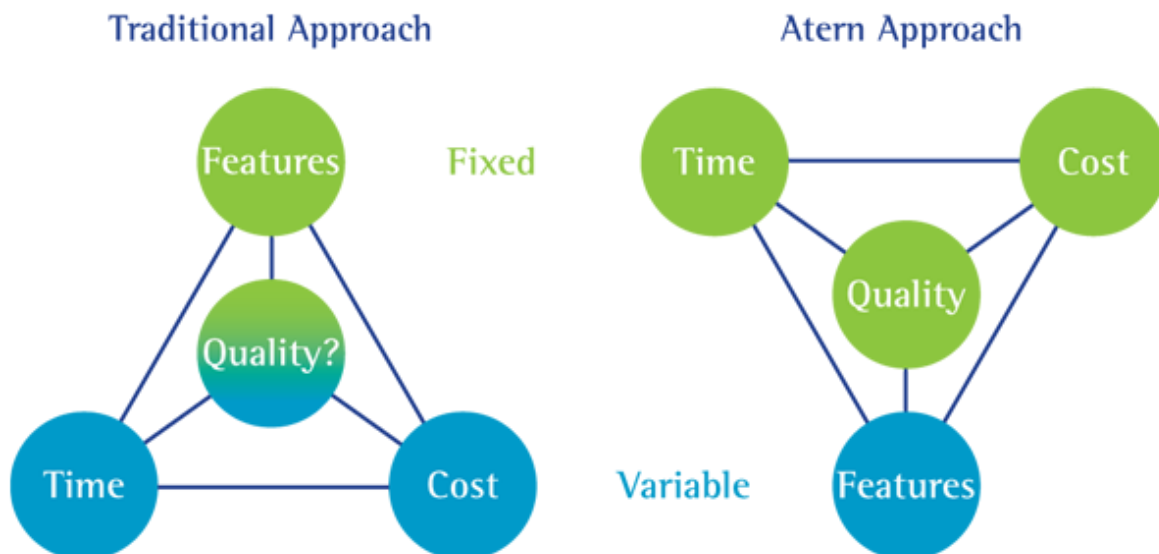
Este sistema permite además visualizar una sección de reportes, lo que le permite a los directivos de la empresa evaluar el rendimiento de cada unidad dentro de la misma. Las rocas de la empresa son alimentadas por cada persona que labora en la misma, apoyando al desarrollo y mejoras dentro de su unidad.

B. METODOLOGÍA Y DESARROLLO

1. **Metodología de Desarrollo de Sistemas Dinámicos (DSDM).** Por sus siglas en inglés. En esta metodología, contrario a las metodologías tradicionales, el tiempo es fijo y la funcionalidad depende del tiempo que se tenga disponible. DSDM es una metodología organizada y orientada en entregar soluciones de negocios de forma rápida y eficiente. Es similar a SCRUM y XP, pero se utiliza específicamente en proyectos en donde el tiempo es fijo.

DSDM es trabajada por pasos, para asegurar la factibilidad y el sentido que el proyecto tendrá para el negocio antes de su creación. DSDM se enfoca en la colaboración y cooperación de las partes interesadas en realizar el proyecto. Utiliza fuertemente la creación de prototipos para asegurarse que todas las partes interesadas tengan una visión clara de lo que llegará a ser el sistema.

Figura no. 1: Enfoques del desarrollo de proyectos



2. Elección de la metodología. DSDM fue elegida por las siguientes características:

- Los resultados del desarrollo son visibles rápida y directamente.
- Debido a que los usuarios finales forman parte del desarrollo del sistema, es más probable que lo utilicen y acepten.
- La funcionalidad básica es entregada rápidamente, con funcionalidades extras agregadas en intervalos regulares.
- Elimina la burocracia y hace más pequeña la línea de comunicación entre las partes interesadas, lo que es ideal para una empresa con una estructura horizontal.
- Ya que se cuenta con la retroalimentación de los usuarios, es más probable que el sistema desarrollado cumpla con las expectativas y necesidades de los clientes.
- El sistema es entregado a tiempo y utilizando el presupuesto asignado para el mismo.
- Los usuarios tienen la habilidad de interferir en la dirección del proyecto para que se adapte a sus necesidades.

3. Conceptos clave de la metodología. Algunos conceptos clave para la utilización de DSDM incluyen:

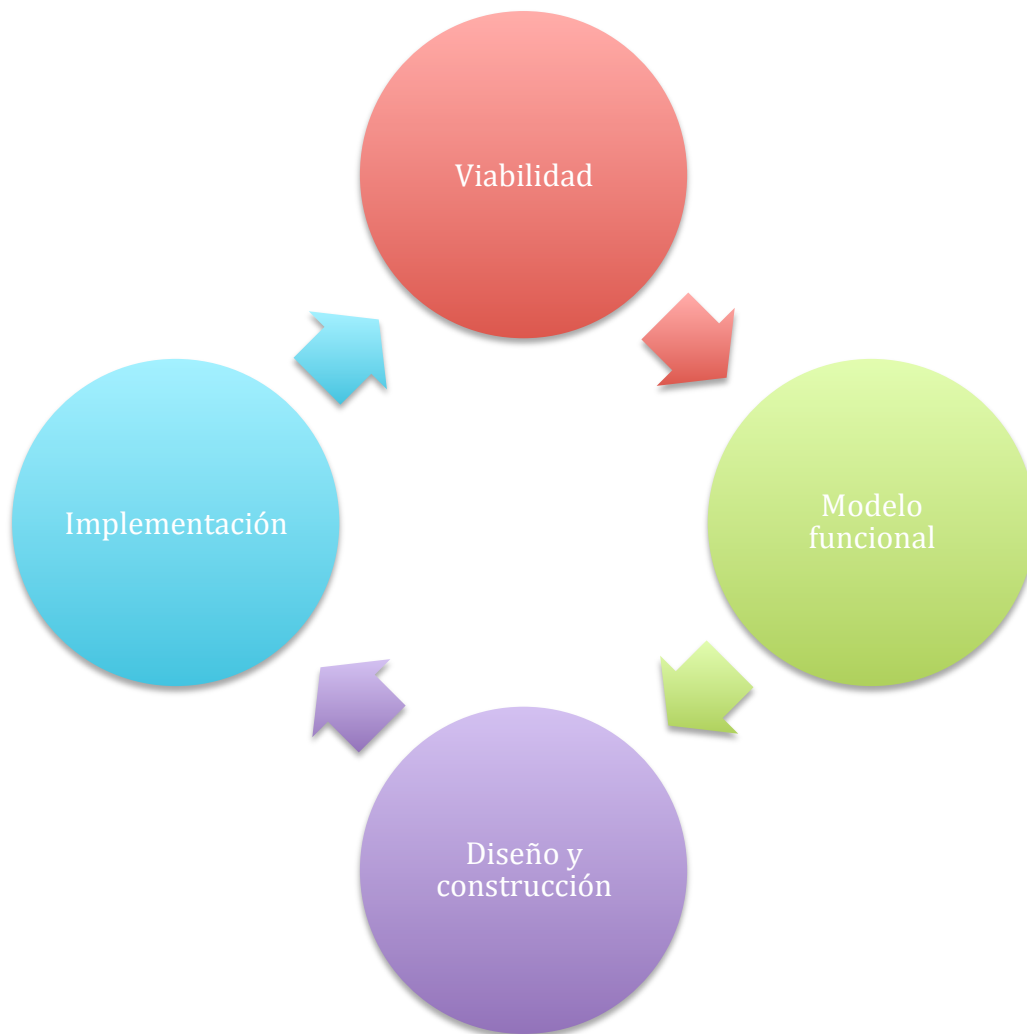
- **Involucramiento activo de usuarios.** Los usuarios finales del sistema deben estar involucrados activamente en su desarrollo. Esto es importante para que el producto sea utilizado y adoptado por ellos.
- **El equipo debe ser empoderado para tomar decisiones.** El equipo debe ser capaz de tomar decisiones de forma rápida e informada, sin tener que pasar por muchas personas para realizar acciones que afecten el proyecto.
- **Entregas frecuentes.** DSDM se enfoca en entregas frecuentes que permiten que los usuarios interactúen con el sistema e ingresen datos reales en etapas cruciales para el desarrollo del proyecto.
- **Desarrollo iterativo, impulsado por retroalimentación de los usuarios.** El desarrollo del sistema es realizado en iteraciones, lo que permite retroalimentación frecuente por parte de los usuarios y la creación de soluciones parciales, pero prontas a necesidades inmediatas, con más funcionalidad agregada en iteraciones

posteriores.

- **Los cambios deben ser reversibles.** Todos los productos deben estar en un estado totalmente conocido en todo momento, esto permite dar marcha atrás si cierto cambio o nuevo requerimiento no funciona como es debido.
- **Los requisitos se definen inicialmente en alto nivel.** Requisitos de alto nivel se elaboran al comienzo del proyecto, antes de que se inicie cualquier codificación, dejando los detalles para ser trabajados conforme el desarrollo del proyecto.
- **La meta es satisfacer las necesidades del negocio.** El propósito del desarrollo es satisfacer las necesidades del negocio más que la perfección técnica.
- **Pruebas integradas.** Las pruebas se realizan en cada paso del desarrollo para asegurar que el producto sea técnicamente correcto y no posea errores.
- **La colaboración y cooperación son esenciales.** La colaboración y cooperación entre todas las partes interesadas son esenciales para el éxito del proyecto. Todas las partes involucradas deben participar juntos para cumplir con los objetivos del negocio.
- **Regla del 20 % / 80 %.** DSDM supone que el 80 % de la solución puede ser desarrollado en el 20 % del tiempo que se necesitaría para producir la solución total. DSDM se centra en este 80 %, dejando el 20 % restante para revisiones posteriores. DSDM supone que no todos los requerimientos para la solución final se conocen desde el inicio del proyecto, por lo que es probable que el último 20 % de las características no esenciales son propensas a tener defectos por su falta de definición clara.

4. Proceso de desarrollo de DSDM. El proceso de desarrollo de DSDM consiste en siete fases. La primera fase es realizada antes de que el proyecto inicie, que es la definición inicial del proyecto. En esta fase son realizados los estudios del proyecto incluyendo el estudio de factibilidad y el estudio del negocio. La fase dos consiste en el desarrollo del modelo funcional. Las fases tres a la cinco consisten en el diseño y construcción del sistema. La fase cuatro consiste en la implementación, despliegue del proyecto. La última fase consiste en el mantenimiento del proyecto.

Figura no. 2: Fases de DSDM



a. Fase uno o pre-proyecto. Esta fase no es estrictamente definida. Ocurre antes que el proyecto inicie oficialmente. En esta fase el proyecto es conceptualizado y se toma la decisión si debe iniciarse o no.

En esta fase son realizados dos estudios importantes para el desarrollo del proyecto: un estudio de factibilidad para identificar si el proyecto funcionará y un estudio de los aspectos del negocio que serán afectados por el proyecto.

- **Análisis del negocio:** en esta fase el equipo investiga los aspectos comerciales sobre el proyecto, respondiendo a preguntas como ¿tiene un buen sentido para el negocio?, ¿quiénes son los participantes y las partes interesadas?, ¿cuál es el mejor plan de trabajo?, ¿qué tecnologías se utilizarán para construir y entregar el proyecto?, entre otras.
- **Análisis de factibilidad:** es realizado para delimitar si el proyecto puede ser realizado con las restricciones y recursos que le fueron asignados.

b. Fase dos, modelo funcional. En esta etapa, prototipos funcionales del proyecto son construidos y revisados. Un prototipo funcional es un prototipo de funciones que el sistema debe realizar y la forma en que las realiza.

La elaboración de prototipos incluye investigar, refinar y consolidar. También incluye muchos aspectos del sistema como la parte de negocios, usabilidad, rendimiento y capacidad y técnica para identificar la mejor manera de resolver el problema propuesto.

c. Fases tres a cinco, diseño y construcción. En esta fase el sistema es diseñado y desarrollado en iteraciones. En cada iteración se elabora el diseño del modelo que está siendo desarrollado, luego se codifica y revisa.

d. Fase seis, implementación y despliegue. En esta fase el producto es terminado y se elabora la documentación asociada al mismo. En esta documentación se incluye la comparación de los requisitos y sus cumplimientos con el desarrollo del producto. Debido a su involucramiento en fases anteriores, para esta fase la mayor parte de los usuarios ya están capacitados para utilizar el sistema.

e. Fase siete o post-proyecto y mantenimiento. Después de que el sistema es creado, el mantenimiento es necesario y debe llevarse a cabo. Este mantenimiento se realiza generalmente en un ciclo similar al que se utiliza para desarrollar el proyecto.

Entre el mantenimiento se incluye resolución de problemas reportados por usuarios y la revisión periódica de la base de datos.

5. Roles de DSDM. DSDM define varias funciones principales que

deben ser satisfechas por los miembros del equipo de desarrollo y las partes interesadas. Los roles definidos por DSDM son:

- **Embajador:** La persona que actúa como un intermediario entre el cliente, los usuarios y el equipo de desarrollo. Coordina el equipo de desarrollo, y debe tener una buena comprensión general de cómo el sistema va a funcionar.
- **Visionario:** La fuerza impulsora detrás del proyecto, mantiene el proyecto en curso dirigido hacia los objetivos de negocio. A menudo es la persona que inició el proyecto.
- **Consejeros:** Las personas que tienen conocimientos prácticos en las áreas del negocio que necesitan ser automatizadas, y en las tecnologías necesarias para automatizar estas áreas.
- **Coordinador técnico:** El coordinador técnico coordina los distintos aspectos técnicos del sistema y asegura que interactúan de manera correcta.
- **Patrocinador ejecutivo:** Un fuerte defensor del sistema, que tiene la capacidad de contribuir con fondos y recursos al proyecto.
- **Administrador de proyecto:** supervisa el desarrollo y creación de prototipos.

6. Herramientas y técnicas. Algunas herramientas y técnicas que deben conocerse para el desarrollo de DSDM son:

- **Priorización de MoSCoW.** La técnica de MoSCoW se acredita a Dai Clegg de Oracle Consulting y es un método simple para determinar las prioridades en proyectos de duración limitada. Cuando se trabaja en los proyectos, debe ser capaz de establecer sus prioridades y niveles de riesgo a medida que avanza el trabajo. Este enfoque también ayuda a mantenerse enfocado en las necesidades de los beneficiarios o interesados en el proyecto. Lo que distingue a MoSCoW de otras técnicas es que se debe considerar tanto lo que desea incluir como lo que no se incluirá.

La funcionalidad se clasifica de acuerdo a su importancia:

Tiene que tener - las cosas que son absolutamente esenciales, y son fundamentales para el sistema.

Debe tener - las cosas que son importantes para la solución de negocio.

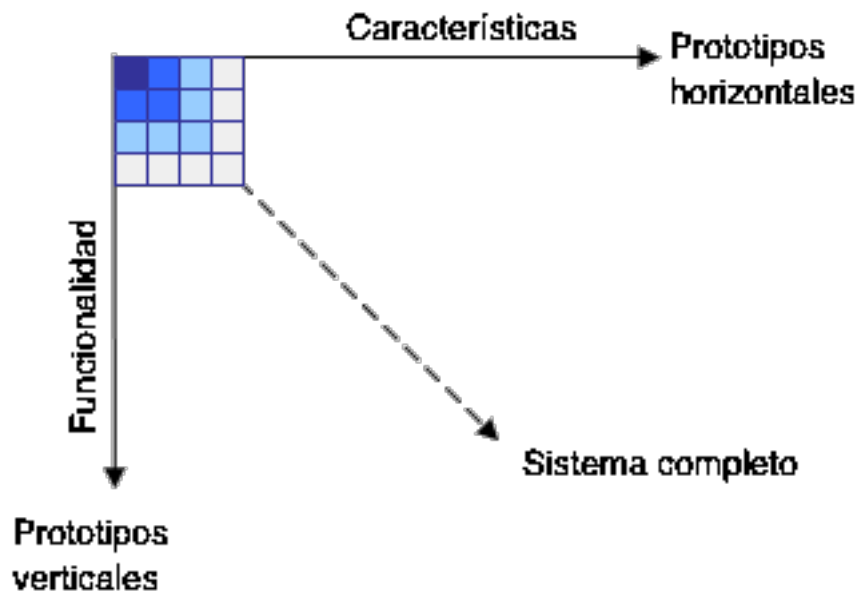
Podría tener - las cosas que son útiles, pero se puede hacer sin un tiempo corto.

No tendrá en esta iteración - las cosas que fácilmente pueden esperar hasta más tarde.

La priorización es importante porque no hay tiempo suficiente para hacer todo, y esas cosas que son esenciales deben ser realizadas antes de las cosas que no lo son.

- **Prototipos.** La elaboración de prototipos es un método utilizado para adquirir la retroalimentación de los usuarios acerca de los diseños futuros de un sistema. DSDM utiliza prototipos verticales, que se refieren a prototipos con pocas características del sistema, pero están completamente implementados.

Figura no. 3: Tipos de prototipos



- **Talleres facilitados.** Permiten un ambiente ideal para la formación de las ideas y de crecimiento rápido y equilibrado de esas ideas. Crean la iniciativa para que todas las partes interesadas estén informadas de las decisiones tomadas por otras partes interesadas. Las decisiones pueden ser hechas por un mayor número de interesados. Las decisiones se toman con rapidez y precisión.
- **Timeboxing.** Períodos fijos de tiempo aseguran que: (1) La entrega del producto se mantiene. (2) Permite la concentración en las prioridades. (3) Todos los miembros se conocen entre sí, en dónde están.

C. *SymfonyFramework*

1. **Framework.** *Symfony* es un *framework* de desarrollo web en PHP y una comunidad de desarrollo. *Symfony* fue desarrollado por Fabien Potencier. Un *framework* o marco de trabajo consiste de una caja de herramientas o *toolbox*, que es una serie de componentes de software prefabricados y rápidamente integrables; y una metodología, que es un “esquema de montaje” para las aplicaciones que vayan a desarrollarse.

Un enfoque estructurado para el desarrollo de *software* puede parecer restrictivo al principio, pero permite que el esfuerzo del desarrollo se centre en las tareas más complicadas y realizar el trabajo de una forma más rápida y eficiente. Además, el uso de modelos estandarizados y de buenas prácticas garantiza la estabilidad, facilidad de mantenimiento y actualización y la escalabilidad de las aplicaciones que se desarrollan.

2. **Arquitectura Modelo Vista Controlador.** El esquema Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software que separa la representación de la información con el uso de la misma. La arquitectura MVC fue implementada por primera vez por Trygve Reenskaug en 1979. Fue implementada en el lenguaje orientado a objetos Smalltalk en los laboratorios de Xerox.

Al pensar en una aplicación es común distinguir tres capas: presentación (UI), lógica de la aplicación y el manejo de recursos. En MVC la capa de presentación es dividida en el controlador y la vista. La separación más importante que debe hacerse es entre la lógica de la aplicación y el aspecto visual de la misma. Los elementos de la arquitectura MVC se describen a continuación:

- **Modelo:** es la representación específica del dominio de información en el que la aplicación opera. El modelo es otro nombre para la capa lógica de la aplicación, también conocida como la capa de dominio. La lógica de la aplicación añade significado a los datos. Muchas aplicaciones utilizan un mecanismo de almacenamiento, como base de datos, para almacenar los datos de forma organizada. La capa de manejo de recursos se encuentra inmersa en la capa del modelo.
- **Vista:** despliega el modelo de una forma adecuada para la interacción, usualmente se utiliza un elemento de interfaz de usuario. La arquitectura MVC es comúnmente utilizada para aplicaciones web, en donde la vista es la

página HTML.

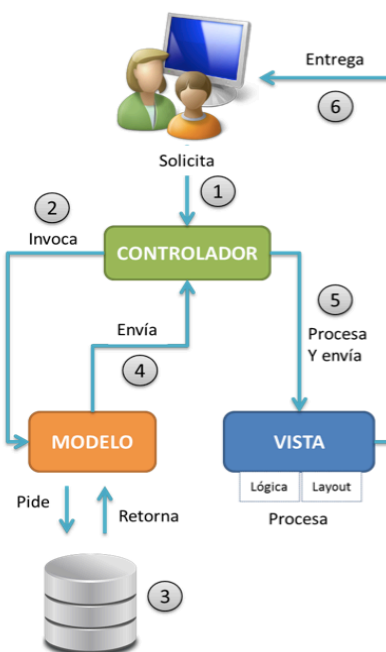
- **Controlador:** procesa y responde eventos que por lo general son iniciados por acciones de usuarios y puede invocar cambios en el modelo y la vista.

Aunque existen diferentes formas de implementar MVC, el flujo generalmente es similar al siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna manera (por ejemplo, el usuario pulsa un botón).
2. Un controlador escucha el evento de entrada de la interfaz de usuario, usualmente a través de un manejador o de una llamada a una función.
3. El controlador accede al modelo, actualizándolo, posiblemente, de una manera apropiada a la acción del usuario.
4. Una vista utiliza el modelo para generar una interfaz de usuario apropiada. La vista obtiene sus datos propios del modelo. El modelo no tiene conocimiento directo de la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, que se inicia un nuevo ciclo.

La figura número cuatro representa la relación entre el modelo, la vista y el controlador.

Figura no. 4: Arquitectura modelo, vista, controlador

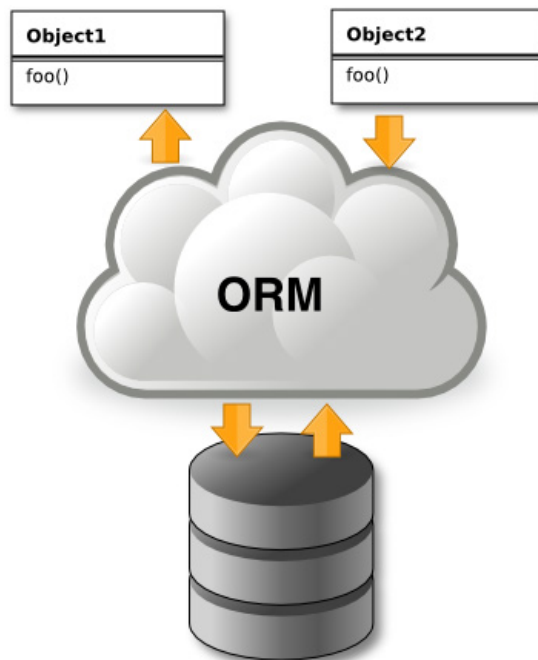


3. **Object Relational Mapping (ORM).** El mapeo de objeto-relación es una técnica de programación que convierte datos entre sistemas incompatibles a objetos de lenguajes orientados a objetos (POO). Con esta técnica se crea un objeto virtual de base de datos que puede ser utilizado dentro del lenguaje de programación.

ORM permite acceder y manipular objetos sin tener que considerar o preocuparse por cómo esos objetos se relacionan a sus fuentes. ORM permite a los desarrolladores mantener una vista consistente de objetos en el tiempo, incluso cuando se obtienen de las fuentes de datos.

Basado en la abstracción, ORM maneja el mapeo de detalles entre un conjunto de objetos y una base de datos relacional, XML, repositorios u otras fuentes de datos, mientras que se encarga de esconder detalles relaciones a las interfaces de conexión a los desarrolladores.

Figura no. 5: *Object Relational Mapper*



Symfony utiliza *Doctrine* como ORM. *Doctrine* es un conjunto de librerías PHP enfocadas en proveer servicios de persistencia y funcionalidades relacionadas. *Doctrine* contiene tanto la capa de abstracción de base de datos, como el manejador de objetos relacionales.

IV. ANÁLISIS DEL SISTEMA (FASE 1)

A. Análisis del negocio

En toda empresa es necesario para los directivos estar enterados y llevar el control de los avances de sus empleados en sus labores diarias. Más aún, si van relacionados con el desarrollo de los valores que se quieren inculcar en los trabajadores y están ligados al progreso empresarial.

El sistema de rocas surge de la necesidad de llevar control y representar estos avances a los directivos de la empresa en la planificación trimestral realizada. Surge también de la necesidad de representar dichos avances de una forma estándar para todas las unidades de negocio que conforman la empresa.

El sistema de rocas es utilizado por casi todas las personas que laboran en la empresa. Cada trimestre nuevas rocas son ingresadas al sistema y luego son alimentadas con los avances de todos los empleados. Al final de cada trimestre es presentado un reporte de cierre de rocas y la planificación para el trimestre siguiente.

Al inicio, se utilizaron diferentes soluciones, pero ninguna satisfacía las necesidades del negocio por llevar un control estandarizado de las metas a las que debía llegar cada equipo de trabajo dentro de la empresa.

Una de las soluciones que fue utilizada durante mucho tiempo fue una hoja de cálculo compartida en *Google Docs* o una hoja de cálculo de MS Excel enviada a cada miembro de la unidad de la empresa. Una hoja era creada y compartida para cada miembro de cada unidad, lo que generaba muchos documentos y era trabajo del líder de área unificarlos y consolidarlos.

La hoja de cálculo era actualizada por cada miembro y la consolidación y estandarización eran complicados de realizar. Es por esta razón que la necesidad de crear un sistema en el que se pudieran llevar los avances de las rocas trimestralmente se hace más evidente.

Para todos los empleados que laboran para la empresa, el presentar un avance en sus rocas y el hecho que se lleve el control sobre dichos avances es sumamente importante, ya que este reporte es uno de los aspectos de los que depende el cálculo del diez por ciento (10 %) del salario de cada persona que labora en la empresa. Este porcentaje es llamado variable, y es pagado dependiendo de los logros de la persona en el mes. Estos logros se deben ver reflejados en el avance de las rocas.

Para facilitar la labor directiva se necesitaba que el sistema que fuese a desarrollarse pudiera generar reportes gerenciales para optimizar la evaluación de los avances en las rocas cada trimestre.

Debido a que no se tenía un sistema preestablecido, fue posible proponer ideas sobre cómo debería realizarse el sistema, desde el proceso de estandarización, hasta la interfaz y la información que debería presentarse a todos los usuarios en pantalla.

La permisión de generar una lluvia de ideas y proponer formas de realizar el sistema, permitió que se lograra un producto de calidad, útil, visualmente agradable y que realiza las acciones que debe.

Antes de realizar la lluvia de ideas fue necesario conocer el proceso como se desarrollan las rocas:

1. Definición de rocas generales de toda la empresa (*EBRocas*). Estas rocas son definidas por gerencia y todas las rocas definidas deben alimentarlas. Desde el desarrollo inicial del sistema fueron definidas cinco rocas generales o *EBRocas*, estas son: automatización, expansión, evolución, aprendizaje y cultura. Cada una de estas rocas refuerzan los valores de la empresa y forman la base del sistema.
2. Luego de definir las rocas generales, cada líder de área debe definir rocas de primer nivel o rocas padre para su unidad. Estas de primer nivel están asociadas directamente a las rocas generales. Las rocas de primer nivel pueden ser definidas únicamente por el líder de área y deben estar enfocadas en apoyar a las rocas generales.
3. Al definir las rocas de primer nivel los líderes de área deben planificar el desglose de las mismas en tareas más pequeñas y planificar en trimestre con base en estas tareas.
4. Una vez que fueron definidas las rocas de cada unidad, al inicio de cada trimestre, reuniones de planificación son sostenidas. En este punto la planificación es presentada a gerencia, quién la aprueba o sugiere mejoras y cambios. En las presentaciones debe demostrarse cómo cada una de las rocas de primer nivel aporta a las rocas generales de la empresa.
5. Cuando la planificación fue aprobada, el líder de cada área o unidad de trabajo de la empresa explica y define la planificación a los miembros del equipo de trabajo y cada uno procede a ingresar sus rocas y empezar a agregar avances.

Un aspecto importante en la definición de las rocas es que deben asociarse a actividades que le den valor a la empresa, no a actividades operativas que no implican algo nuevo o alguna mejora medible en un proceso existente. Actividades como mantenimiento a sistemas y solución de problemas deben dejarse fuera del desarrollo de las rocas, ya que según la literatura, una roca debe ser un pilar y algo

que agregue nuevo valor a la empresa.

B. Análisis de factibilidad

Un análisis de factibilidad es un estudio realizado para determinar los objetivos de la organización y decidir si éstos pueden ser satisfechos por la creación del proyecto sobre el que se realiza el estudio. Es necesario analizar los objetivos de la empresa para determinar si es posible realizar un proyecto y si dicho proyecto va a minimizar los problemas de la empresa o agregar valor a alguna operación existente.

Un análisis de factibilidad permite determinar también los costos contra los beneficios del desarrollo del proyecto, así como también el grado de aceptación que se generará con la propuesta dentro de la institución como en este caso que el producto es interno.

Según el Diccionario de la Real Academia Española del 2013, la factibilidad es la cualidad o condición de que algo puede realizarse. El estudio de factibilidad es el análisis que realiza una empresa para determinar si el proyecto que se propone será exitoso o un fracaso, y cuáles serán las estrategias que se deben desarrollar para que sea exitoso.

Algunos de los objetivos del estudio de factibilidad se listan a continuación:

- Reducción de errores y mayor precisión en los procesos.
- Reducción de costos mediante la optimización o eliminación de los recursos no necesarios.
- Integración de todas las áreas y subsistemas.
- Actualización y mejoramiento de los servicios a clientes o usuarios.
- Hacer un plan de producción.
- Aceleración en la recopilación de los datos.
- Reducción en el tiempo de procesamiento y ejecución de las tareas.
- Automatización óptima de procedimientos manuales.
- Disponibilidad de los recursos necesarios para llevar a cabo los objetivos señalados.
- Saber si es posible producir con ganancias.

- Conocer si la gente utilizará el producto desarrollado¹.

Para que un proyecto pueda ser considerado como factible debe cumplir con cuatro evaluaciones básicas: evaluación ambiental, evaluación técnica, evaluación financiera y evaluación socio-económica. Para el presente proyecto se realizó únicamente la evaluación técnica y la evaluación financiera porque se consideró que las otras no eran necesarias, pues no se estaba incluyendo ningún producto masivo a algún mercado y se consideró que el concepto del proyecto no necesitaba los otros estudios.

1. Evaluación técnica. Para determinar si el proyecto es viable técnicamente se evaluaron las tecnologías a utilizar para desarrollarlo. Se quería que fuese una herramienta web para poder ser accesible desde cualquier lugar. No existían restricciones de explorador de internet y se quería una página amigable a la que se pudiera ingresar con el correo de la empresa.

Ya que en *EBclosion* se trabaja con el *framework Symfony*, el lenguaje de programación PHP y el manejador de bases de datos MySQL, se determinó el desarrollo del sistema debería de desarrollarse utilizando estas herramientas de tecnología.

Otro aspecto que facilitó la evaluación técnica fue que ya se tenían los servidores de desarrollo y producción para la elaboración del proyecto, por lo que no se tuvo que invertir en compra de equipo adicional.

2. Evaluación financiera. La evaluación financiera fue elaborada por gerencia. En esta evaluación se identificó si era posible y viable para la empresa que la elaboración de la segunda versión del sistema de rocas fuese realizada en un período de 400 horas.

En estas horas se incluía la toma de requerimientos, el trabajo del equipo de diseño gráfico, el desarrollo de la funcionalidad del proyecto, elaboración de manuales para líder de área, miembro de equipo y la sección administrativa, así como también la realización de pruebas en tiempo real y de aceptación del nuevo sistema.

Se llegó a la conclusión que sí era posible y rentable para la empresa realizar el desarrollo del proyecto en ese tiempo y con esos recursos, ya que los beneficios serían mayores que el costo que representaría el proyecto incluyendo a todos los involucrados.

C. Versiones

La necesidad de la elaboración del sistema de rocas no es un concepto nuevo y el sistema desarrollado para este trabajo es la tercera versión del mismo en un período dos años.

1. Versión 0 – Hoja de cálculo. Por años se utilizó un formato en una hoja de cálculo en *Google Docs* o MS Excel, dependiendo del líder de cada unidad. Esta opción no era práctica, ya que las rocas podían editarse por cualquier persona en cualquier momento, no había forma de granularlas y tampoco existía ningún estándar para la información.

Otros inconvenientes con el uso de la hoja de cálculo, si se utilizaba la versión de MS Excel, consistía en que era necesario enviar un archivo con los avances a cada persona del equipo de trabajo para que se actualizara la hoja y enviársela de vuelta al líder de área para que hiciera el cierre y consolidado de los avances, lo que tomaba mucho tiempo y ocasionaba errores.

El mayor inconveniente que ocasionaban las hojas de cálculo era que resultaba complicado generar cualquier tipo de reportes ya que la información no era completamente estándar.

Por mucho tiempo ocurrió también que las rocas en las que no se había avanzado eran borradas de la hoja de cálculo para que no afectara el porcentaje de la unidad al cierre del trimestre.

Por todos esos problemas ocasionados, se decidió que era necesario el desarrollo de una herramienta para llevar el control de las rocas de la empresa y se iniciaron los análisis para el desarrollo de dicho sistema.

Figura no. 6: Sistema de rocas en hoja de cálculo de *Google Docs*

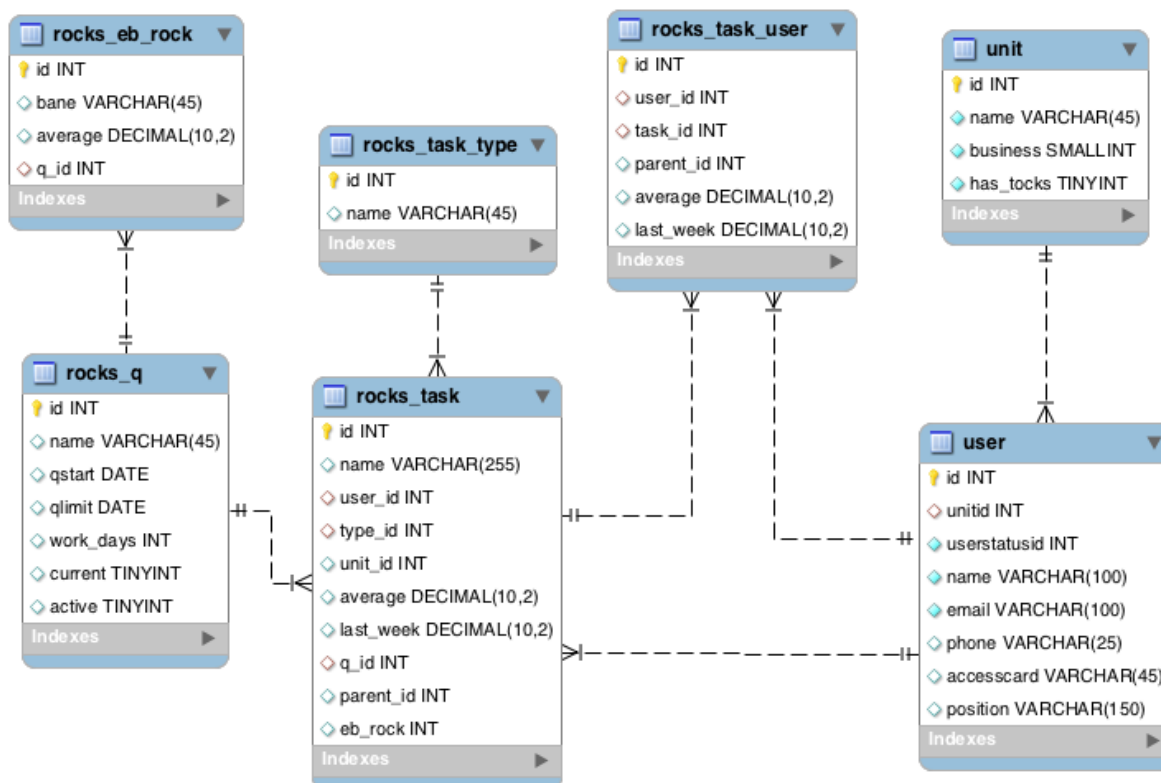
Project Development		Q DAYS		Q START	DATE	Q LIMIT	PRODUCTIVITY RANGE TO DATE		
Q4 - 2010		Left	Gone	07/01/2011	13/04/2013	31/03/2011	991.39%	996.39%	1001.39%
Department Rocks		-744	827						
WEEKLY ADVANCE		YOUR DAILY ADVANCE %		Q WORKABLE DAYS		DAILY ADVANCE	Watch Out	Good	Excellent
10%		0.12%		83		1.20	95.21%		
#	ITEM			LAST WEEK %	ACTUAL %	COMMENTS			
1	Desarrollo de productos			97%	100%				
1	1	Loyalty V1		98%	100%				
2	2	Modulo de cobranza V1		100%	100%				
3	3	Schools V1		100%	100%				
4	4	Globalogik V2		85%	100%				
5	5	Tragatrix		100%	100%	*** Cambio por prioridad			
2	Producción exclusiva			84%	99%				
6	1	Picapic		99%	99%				
7	2	Zootropic		77%	98%				
8	3	Ligas de CentroAmerica		100%	100%				
3	Foros virtuales			94%	100%				
11	1	Research and development		98%	100%				
12	2	SEO and analytics		88%	100%				
13	3	Piclo		96%	99%				
4	Mejora continua			63%	80%				
15	1	Desarrollo profesional (Capacitaciones)		90%	100%				
16	2	Desarrollo empresarial (KPIs, scrum, ebnet)		95%	100%				
17	3	Perfiles y pensum		5%	40%				
5	Personal			89%	97%				
18	1	Lecturas / Hobbies		91%	99%				
19	2	Certificaciones / Concursos		85%	92%				
20	3	Deporte / Ejercicio		91%	100%				
TOTALS				85.33%	95.21%				

2. Versión 1 – Herramienta en línea. El desarrollo de la segunda versión de la herramienta fue iniciada a principios del año 2011. El desarrollo fue planeado ya que el sistema de las hojas de cálculo no suplía las necesidades de la empresa, y para una empresa dedicada al desarrollo de tecnología, era necesario evolucionar y buscar la mejora continua de sus herramientas de trabajo.

El desarrollo de la primera versión en línea del sistema de rocas fue realizado con pocos estándares, sin tener cuidado de la seguridad del sistema y sin utilizar un las mejores prácticas de desarrollo. Mucha de la labor de la toma de requerimientos fue realizada con este sistema, ya que en forma en la que deberían de funcionar los cálculos, la manera en cómo se ingresan las rocas y la definición de permisos fue identificada y trabajada desde esta versión y mejorada para la versión posterior que es la que se presenta en este trabajo.

Algunos de los principales problemas que surgieron con esta versión fue la escalabilidad de la base de datos, ya que por el modelo que se había planteado, los datos almacenados del sistema comenzaron a crecer y la respuesta del sistema se tornó lenta al paso de un año.

Figura no. 7: Esquema de la base de datos de la primera versión del sistema



Otro problema con la primera versión en línea del sistema fue que el desarrollo nunca se terminó en realidad, ya que aunque se intentó estandarizar la forma de manejar las rocas para todos los equipos, cada líder de área solicitaba cambios en la forma de ingreso, en la visualización de los datos, en los permisos que debía tenerse, por lo que el desarrollo se extendía cada vez que algún líder de área solicitaba cambios.

Una deficiencia que tenía el sistema era la generación de reportes, ya que derivado del mismo problema del almacenamiento masivo en la base de datos, la exportación a Excel de la información contenida en el sistema era realizada lentamente, por lo que algunas veces el servidor se quedaba sin responder y sin entregar los reportes deseados.

Problemas de seguridad fueron identificados en el camino. Debido a que no se utilizó ningún *framework* para el desarrollo del sistema y no se le dio prioridad a la seguridad del mismo, los campos de texto eran vulnerables a ataques de inyección SQL, debido a que se podían ingresar sentencias de SQL y el sistema desplegaba el error de la base de datos en pantalla.

Esta fue una vulnerabilidad reportada cuando ya se tenían planes del desarrollo de la segunda versión del sistema por lo que, aunque era de suma importancia solucionarlo, su resolución fue omitida temporalmente.

A pesar de los inconvenientes generados por el desarrollo y la utilización de este nuevo sistema de rocas en la empresa, fue utilizado durante un año, pero con el tiempo se hizo evidente que era necesario modificarlo y mejorarlo.

Figura no. 8: Diseño inicial de la primera versión del sistema de rocas



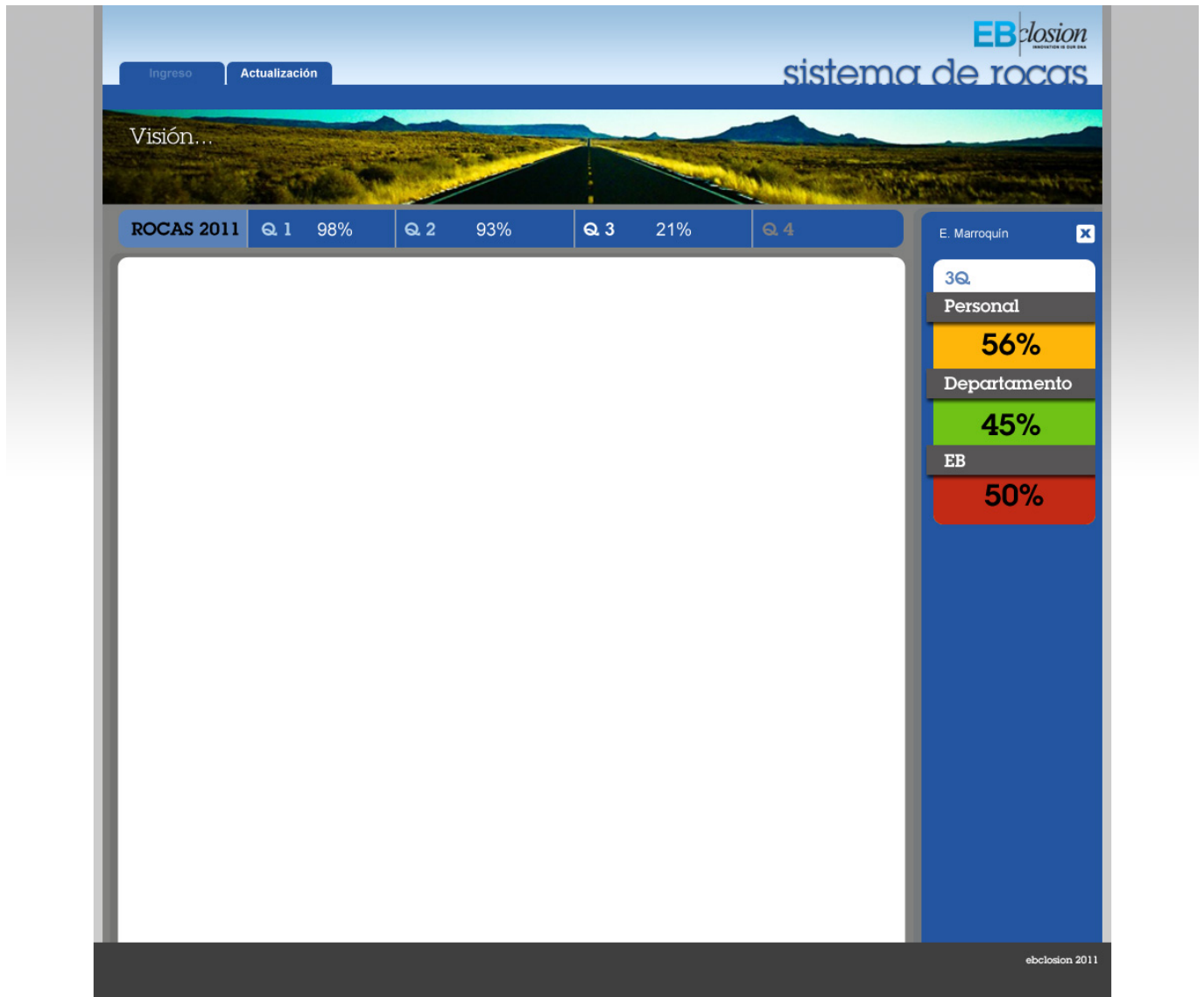
Figura no. 9: Diseño modificado de la primera versión del sistema

Figura no. 10: Diseño final de la primera versión del sistema

EB *clasion*

Visión Queremos y sabemos cómo contribuir al avance de la humanidad con nuestras tecnologías.

Ingreso Actualización sistema de rocas

Q1 2012 (Actual) exportar  log out 

Usuario	Silvia Paniagua	Esperado	78.54%
Área	Globalogikas	Esperado Diario	1.54%
Puesto		Semana Anterior	0.00%
Inicio 	01/01/2012	Avance Diario	0.00%
Límite 	30/03/2012	Avance Semanal	0.00%
Días Transcurridos	51	Personal	0.00%
Días Restantes	14	Departamento	54.27%
Días Hábiles	65	EB	37.22%

Globalogikas

Nombre	EB Rock	% Semana Anterior	% Actual
<input type="checkbox"/> Globalogik III	Evolución		
<input type="checkbox"/> Azucar	Automatización		
<input type="checkbox"/> E-Commerce	Expansión		
<input type="checkbox"/> RRPP	Expansión		
<input type="checkbox"/> Gestión Operativa	Evolución		
<input type="checkbox"/> Aprendizaje	Aprendizaje		

actualizar

D. Análisis de requerimientos

Parte del pre-proyecto o fase uno de la metodología DSDM es la reunión y definición de los requerimientos funcionales y gráficos del sistema. Debido a que ya se tiene conocimiento del negocio y de los objetivos que éste desea lograr con el desarrollo de este proyecto, se procedió a realizar la recopilación de requerimientos.

Para esta fase fue importante tomar en cuenta las necesidades de cada grupo de personas afectadas por la creación del nuevo sistema. Por suerte, la barrera de adaptación a un sistema web ya había sido rota con la creación de la primera versión, por lo que esta versión supondría únicamente mejoras en el despliegue de la información, en la funcionalidad y tiempo de respuesta.

Los requerimientos funcionales para la segunda versión incluyeron muchos de los requerimientos que ya habían sido tomados para la primera versión del sistema en línea, y el resto suponían mejoras a los problemas identificados en la versión que se estaba siendo utilizada. Entre los requerimientos de la primera versión se incluyeron:

- Acceso restringido por usuario y contraseña.
 - A este requerimiento fue agregado que el ingreso al sistema se realizara con el correo electrónico de la empresa y la contraseña del mismo, para facilitar a los usuarios y evitar problemas de contraseñas olvidadas.
- Mostrar información correspondiente al trimestre. Entre la información que debe mostrarse se encuentra:
 - Fechas de inicio y finalización del trimestre.
 - Días disponibles en el trimestre, es decir los días que han pasado y los días que quedan.
 - El día en el que se encuentra el trimestre.
- En todas las pantallas del sistema deberían desplegarse la misión y visión de la empresa.
- Realizar un cálculo del porcentaje esperado al día actual del trimestre, ya que los avances de las rocas deberían realizarse diaria o semanalmente y esto les permite a los usuarios visualizar el avance que deberían de llevar al día actual.
- Para cada usuario debe mostrarse la unidad a la que pertenece.
- Deben definirse roles para nivel administrativo, líderes de área y miembros

de cada área.

- **Rol de miembro del equipo:** tendrá acceso a actualizar y ver únicamente sus rocas.
 - **Rol de líder de área:** tendrá acceso a ingresar rocas de primer y segundo nivel e ingresar y actualizar sus propias rocas. El líder de área también podrá ver las rocas de los miembros de su equipo y el avance de las otras unidades, sin ver el detalle de las mismas.
 - **Rol administrativo:** tendrá acceso al manejo del sistema y accesos a todos los reportes. Podrá realizar todo lo que hacen los roles definidos anteriormente.
- Deben generarse reportes semanalmente con el cierre semanal de cada unidad.
 - Deben poderse descargar las rocas de toda la unidad a un archivo de MS Excel.

Como se mencionó anteriormente, en esta versión del sistema se hicieron diversos cambios para que la nueva tecnología pudiera acoplarse a cada una de las unidades de la empresa. Aunque al inicio esto pareció una buena idea, con el tiempo se fueron solicitando más y más cambios, lo que ocasionó que el desarrollo del sistema continuara por meses. Algunas solicitudes de cada líder de área fueron:

- Permitir reasignar las rocas ingresadas a otro miembro del equipo.
- Permitir más de un líder por área, que pueda ver el avance de los demás miembros del equipo.
- Permitir que cada unidad se subdivida en departamentos o áreas más pequeñas.

Debido a los problemas que surgieron en la versión anterior y a las mejoras que debían hacerse en el diseño de la base de datos se solicitó realizar una nueva y mejorada versión del sistema de rocas para *EBclosion*. Las mejoras que fueron solicitadas para la segunda versión del sistema fueron:

- Utilizar un *framework* para el desarrollo, lo que permitió hacer un sistema robusto en menos tiempo.
- Establecer una fecha límite para el ingreso y edición de rocas de primer y segundo nivel.
- Hacer los reportes visibles para todos.
- Diseñar un sistema de reportes de cierre semanal en línea, ya que en la

versión anterior estos reportes eran calculados manualmente y enviados por correo electrónico.

- Diseñar reportes para el despliegue de avances.
- Cambiar el sistema de servidor.
- Agregar módulos para administración de usuarios, trimestres, EBRocas y todo lo necesario para que el sistema se autónomo.
 - En la administración de trimestres, agregar un campo de fecha que indique hasta cuándo las rocas de primer nivel pueden editarse.
- Que una unidad pueda estar subdividida en departamentos.
 - A nivel gerencial la unidad se sigue viendo como una sola sin importar la cantidad de departamentos en los que se encuentre dividida.
- Permitir que un usuario pueda estar asignado a uno o más departamentos dentro de la unidad.
- Crear manuales para el uso correcto del sistema.

Para la creación de la segunda versión del sistema fueron tomados en cuenta estos requerimientos y se hizo énfasis en mejorar la seguridad del sistema anterior; así como mejorar el tiempo de respuesta a las acciones de actualizar el porcentaje de avance de una roca. Se enfatizó también en la creación de reportes visibles en línea, que ayudarán a la gerencia a tomar decisiones de forma más rápida y eficiente.

V. DISEÑO DEL SISTEMA (FASE 2)

A. Diseño de la aplicación

El propósito de la aplicación es llevar el control del avance de las rocas definidas trimestralmente para cada unidad de la empresa *EBclosion*. La nueva versión del sistema de rocas busca corregir y mejorar los problemas que presentaba la versión anterior, incluyendo mejoras.

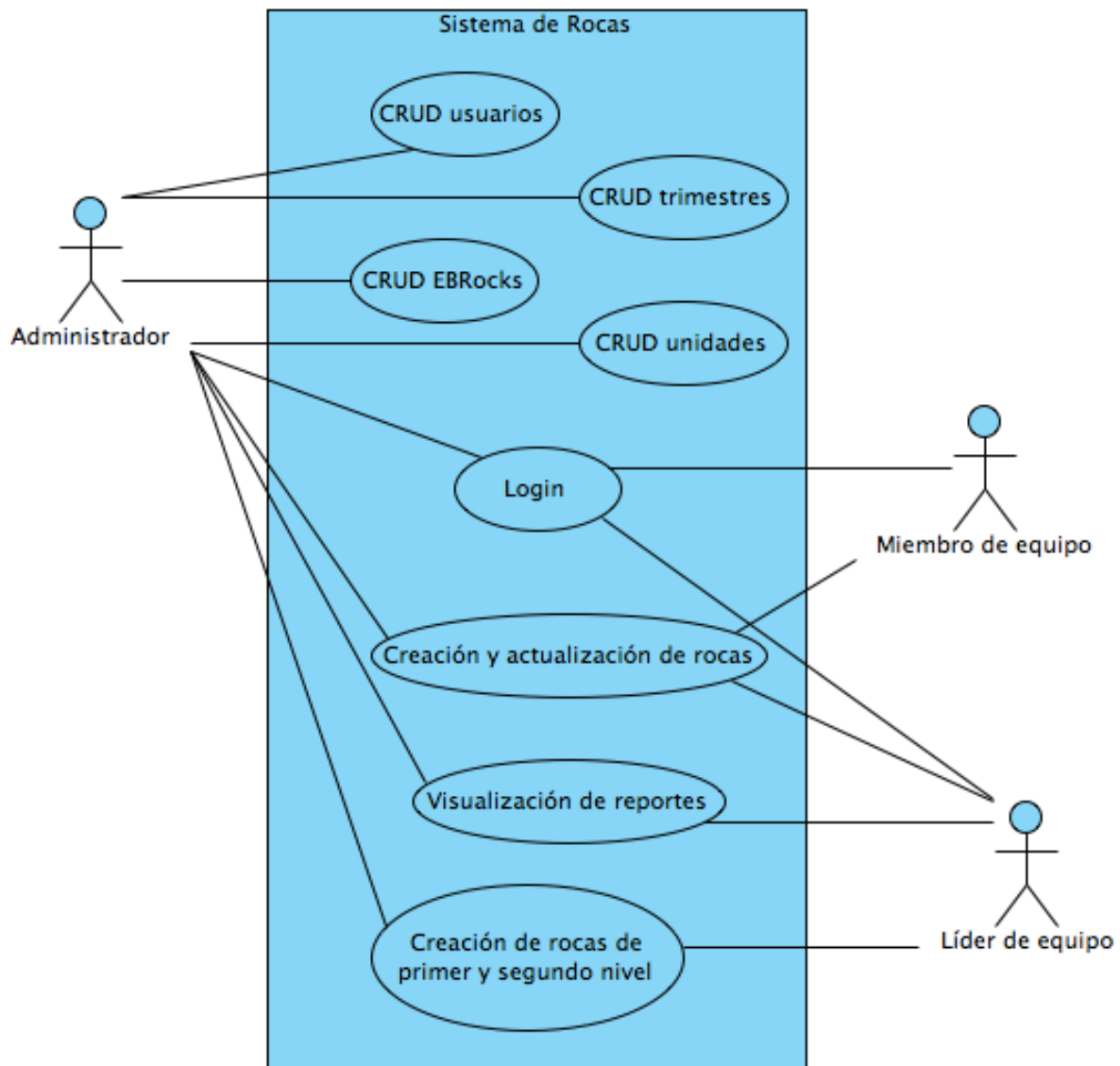
Para llevar a cabo el diseño de la aplicación web se realizaron diferentes actividades para satisfacer los requerimientos encontrados y el desarrollo de la metodología DSDM. La primera de esas actividades fue el diseño de diagramas UML de casos de uso y de secuencia, con el fin de relatar y de identificar las partes del sistema que iban a ser realizadas.

Una vez se ejecutaron los diagramas UML, se procedió a realizar una priorización de MoSCoW para determinar qué funcionalidades se implementarían en qué iteración. Cuando se tuvo clara la planificación, se procedió a realizar prototipos para la aplicación. Cuando los prototipos fueron aprobados se inició la fase de implementación por iteraciones.

B. Diagramas UML

1. Diagrama de casos de uso. El diagrama de casos de uso del sistema fue elaborado con el fin de ilustrar las actividades globales que debe hacer el sistema, desde la parte administrativa hasta las actividades básicas que deben estar disponibles para todos los usuarios.

Figura no. 11: Diagrama de casos de uso del sistema



2. Diagrama de secuencia. El diagrama de secuencia que se presenta a continuación tiene como propósito ilustrar el proceso de crear una nueva roca de nivel uno y dos, que son los niveles creados por el líder de cada equipo y luego la creación de rocas del nivel tres y cuatro para su posterior actualización.

Este diagrama fue elaborado con el propósito de ilustrar el proceso de creación y actualización de las rocas, ya que existen restricciones de permisos para los miembros del equipo que requieren que el líder haya ingresado rocas previamente.

Figura no. 12: Diagrama de secuencia de la creación de rocas para un nuevo trimestre

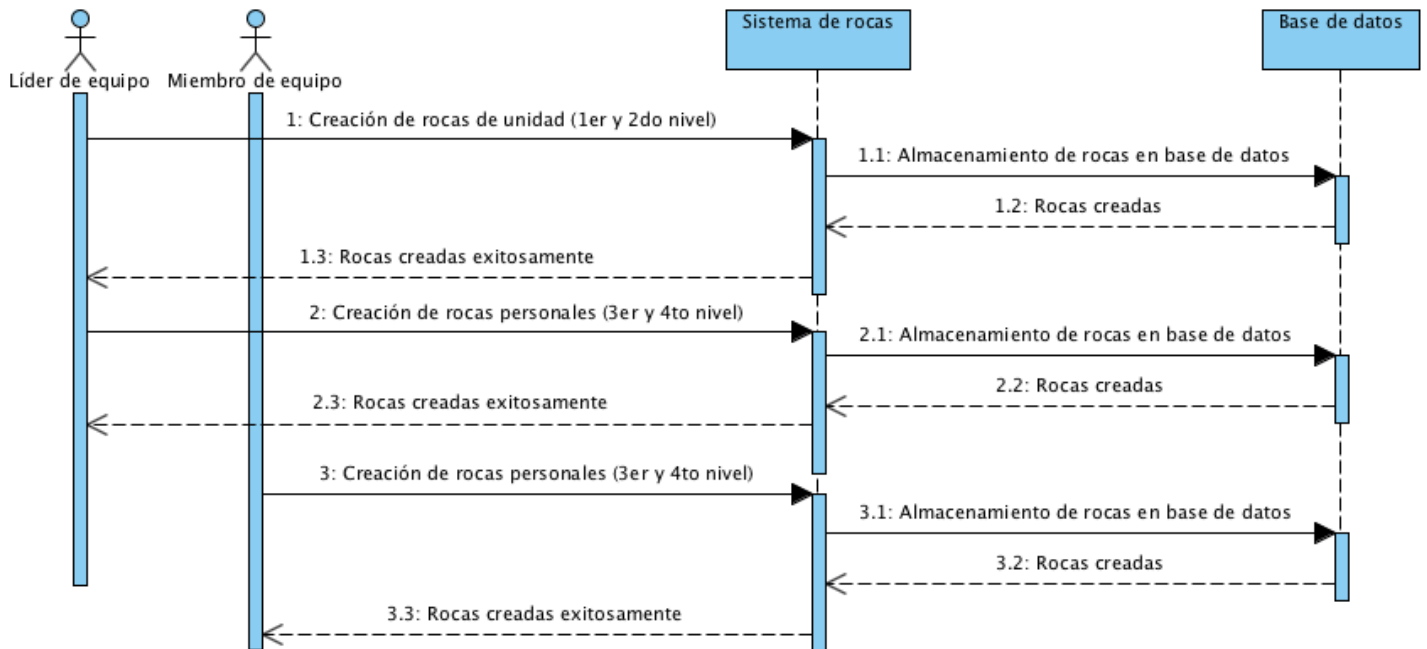
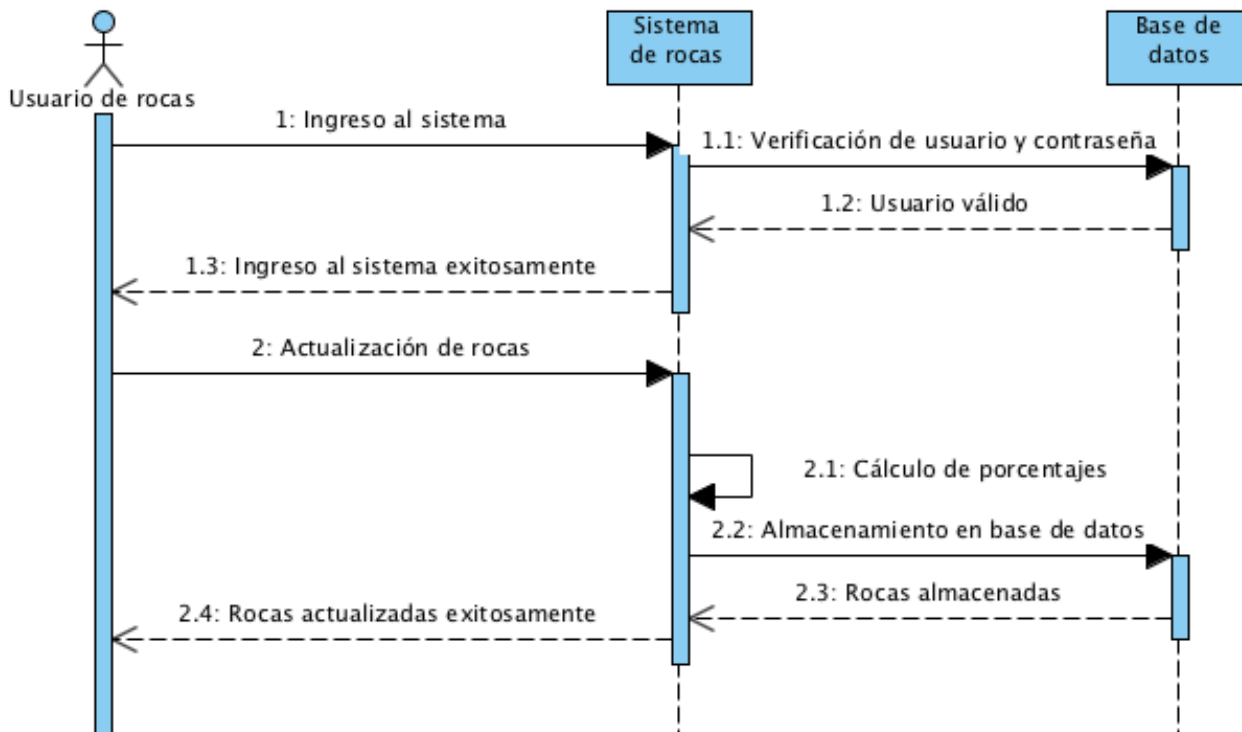


Figura no. 13: Diagrama de secuencia de la actualización de rocas



3. Priorización de MoSCoW. El método de priorización de MoSCoW planifica y divide el desarrollo en iteraciones, y en cada iteración se selecciona qué tiene que tener el sistema en ese momento, que debería de tener, que podría tener y no tendrá en la iteración analizada.

Para el desarrollo del presente proyecto se realizaron cinco iteraciones. A continuación se describen las actividades que fueron realizadas en cada iteración. Las actividades fueron obtenidas del diagrama de casos de uso, en donde se determinaron las características que debía tener el producto desarrollado.

Tabla no. 1: Priorización de MoSCoW en la primera iteración

Iteración 1	
Tiene que tener	Creación y maquetado del sitio con el diseño gráfico final, creación y conexión de la base de datos del sistema
Debe tener	El desarrollo y configuración del sistema deben estar realizados con <i>Symfony</i> 1.4
Podría tener	Autenticación del usuario con correo y contraseña de la empresa
No tendrá en esta iteración	Creación, edición y eliminación de rocas

Tabla no. 2: Priorización de MoSCoW en la segunda iteración

Iteración 2	
Tiene que tener	Ingreso con usuario y contraseña, creación, eliminación y edición de rocas
Debe tener	Una visualización granular de las rocas y protagonismo a las rocas de <i>EBclosion</i> .
Podría tener	Cálculo de avances
No tendrá en esta iteración	Área administrativa (creación, lectura, actualización y eliminación de usuarios, trimestres, unidades y EBRocas)

Tabla no. 3: Priorización de MoSCoW en la tercera iteración

Iteración 3	
Tiene que tener	Cálculo de avances de rocas
Debe tener	Mejora en el tiempo de respuesta para la realización de los cálculos
Podría tener	Área administrativa (creación, lectura, actualización y eliminación de usuarios, trimestres, unidades y EBRocas)
No tendrá en esta iteración	Sección de reportes

Tabla no. 4: Priorización de MoSCoW en la cuarta iteración

Iteración 4	
Tiene que tener	Área administrativa (creación, lectura, actualización y eliminación de usuarios, trimestres, unidades y EBRocas)
Debe tener	Forma de acceso rápido al área administrativa y correcta división de permisos
Podría tener	Sección de reportes
No tendrá en esta iteración	Manuales

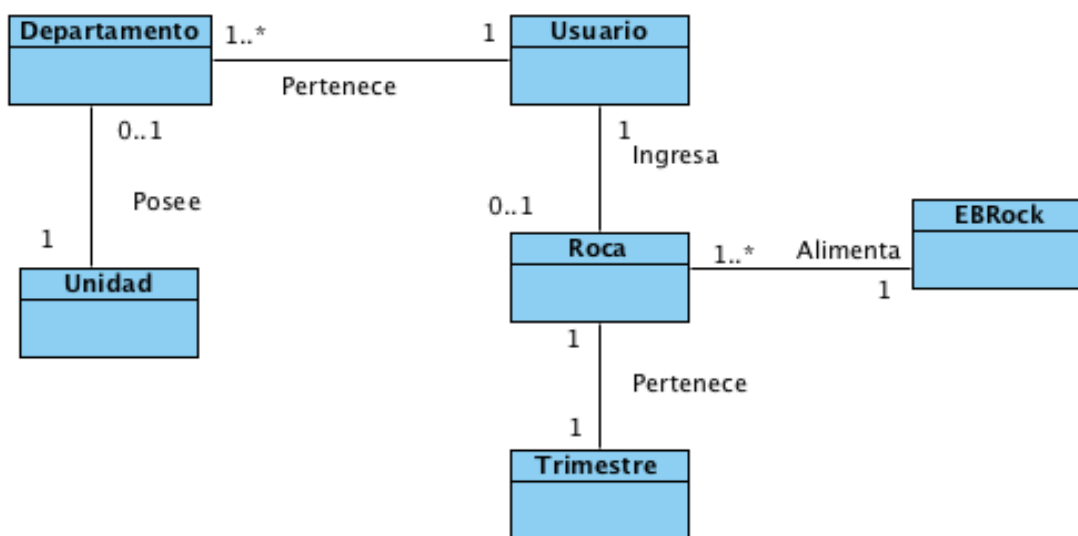
Tabla no. 5: Priorización de MoSCoW en la quinta iteración

Iteración 5	
Tiene que tener	Sección de reportes
Debe tener	Manuales del sistema
Podría tener	Descargar a MS Excel los reportes

4. Esquema de datos. Ya que se tiene en la planificación del proyecto los casos de uso que deben realizarse, se elaboró un esquema de datos en donde se definen las relaciones básicas que debe tener cada elemento que interactúa en el sistema.

Los elementos básicos que interactúan con el sistema son: usuarios, unidades y departamentos, trimestres y rocas. Cada uno de estos elementos interactúa para desarrollar el sistema y pueden ser representados como clases. Basado en los casos de uso, el esquema de datos o diagrama de clases principal es el que se presenta a continuación:

Figura no. 14: Diagrama de clases que representa el esquema de datos del proyecto



5. Prototipos. Para la fase de elaboración de prototipos se tomaron en cuenta los aspectos positivos de la primera versión del sistema de rocas y se intentó mejorar los aspectos que no eran del agrado de los usuarios. Esta información fue obtenida de la retroalimentación que los usuarios finales daban del sistema.

Los prototipos representan la vista en el modelo MVC utilizado para el desarrollo del proyecto. Los prototipos permiten al usuario interactuar con el sistema de forma indirecta y les permite hacer sugerencias sobre los cambios que deban realizarse.

A continuación se muestran diferentes vistas propuestas para los cambios en el sistema de rocas. El resultado final no fue el propuesto exactamente al inicio debido a los cambios en los requerimientos, pero la utilización de esta herramienta permitió que los cambios fuesen hechos de forma rápida y precisa.

Figura no. 15: Pantalla de ingreso

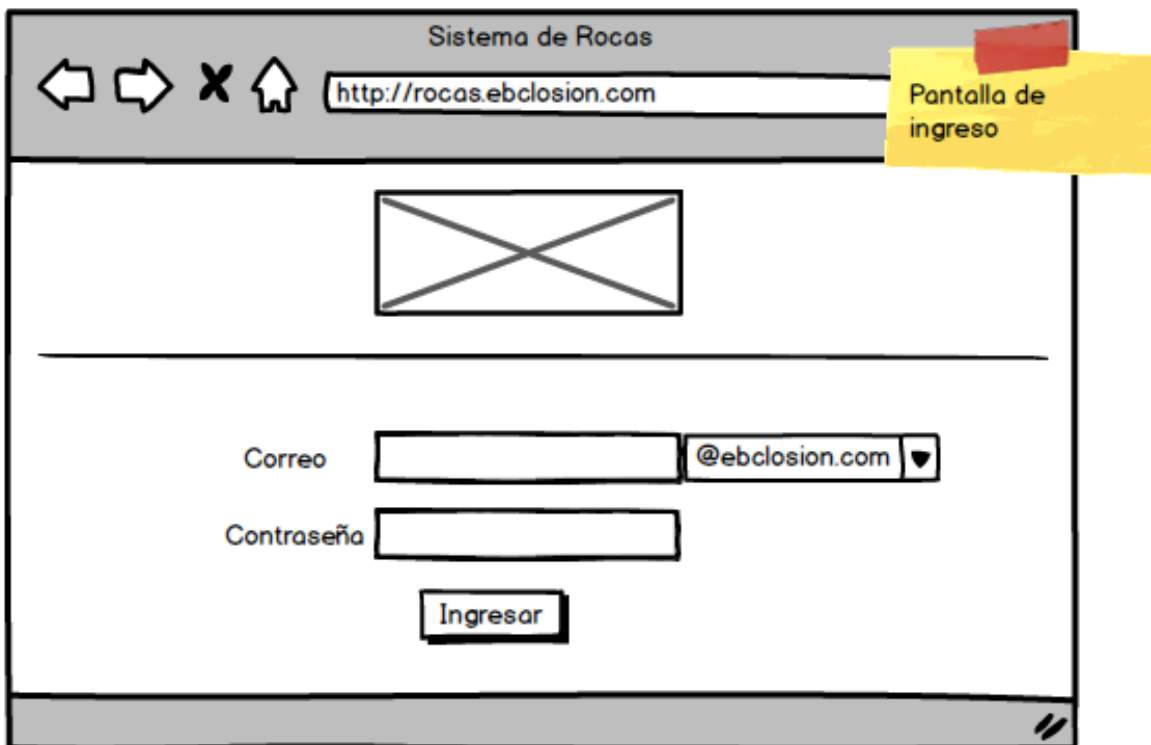


Figura no. 16: Pantalla de inicio de la aplicación, visualización de rocas

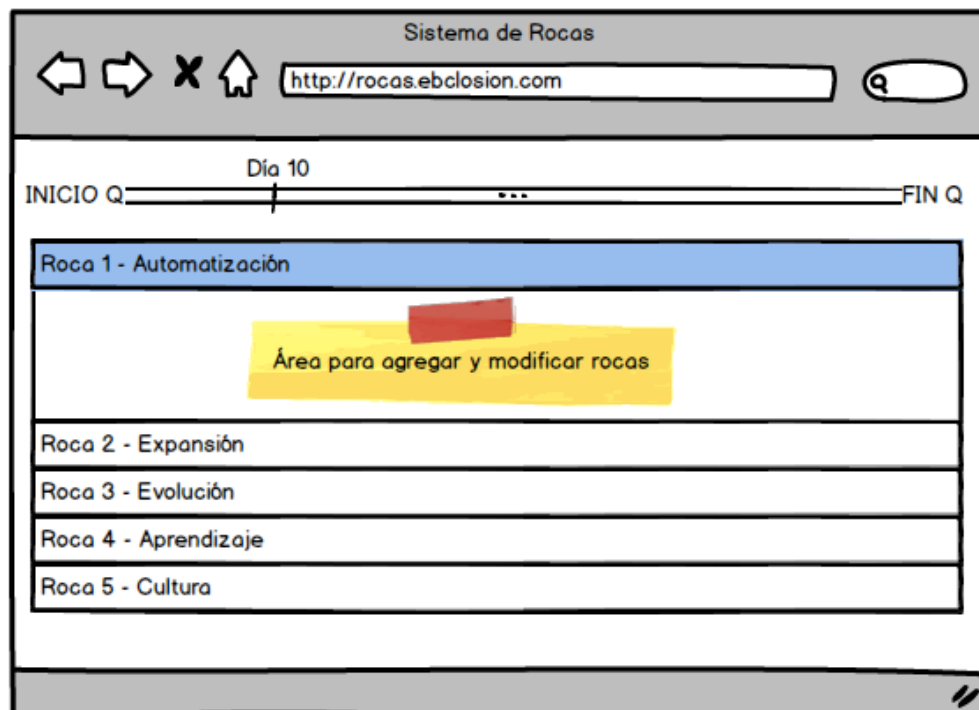


Figura no. 17: Área de ingreso de rocas

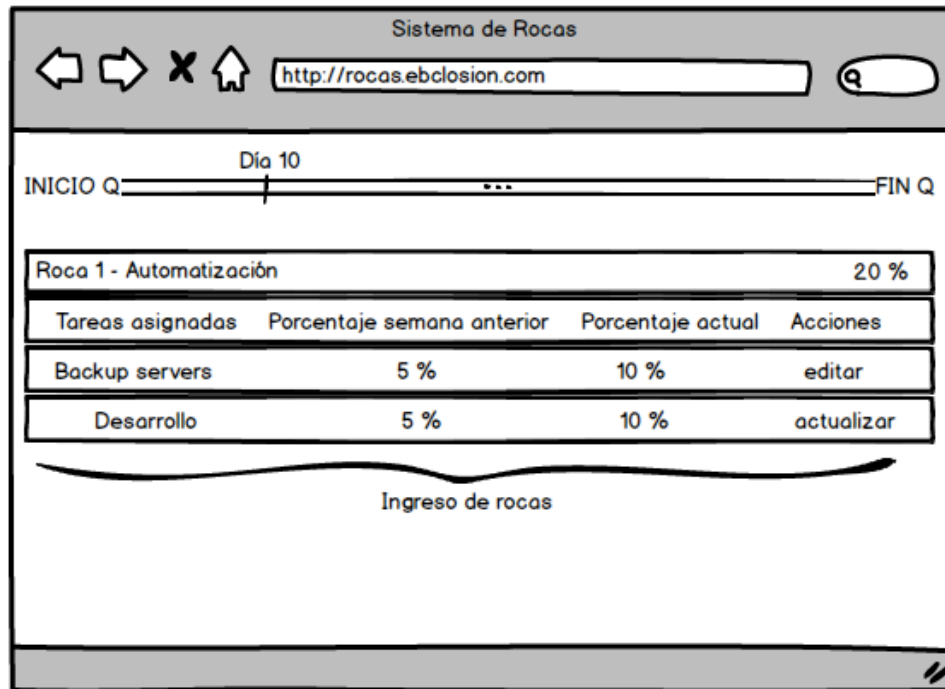


Figura no. 18: Sección de administración de usuarios. Esta vista sería utilizada para la administración completa del sistema

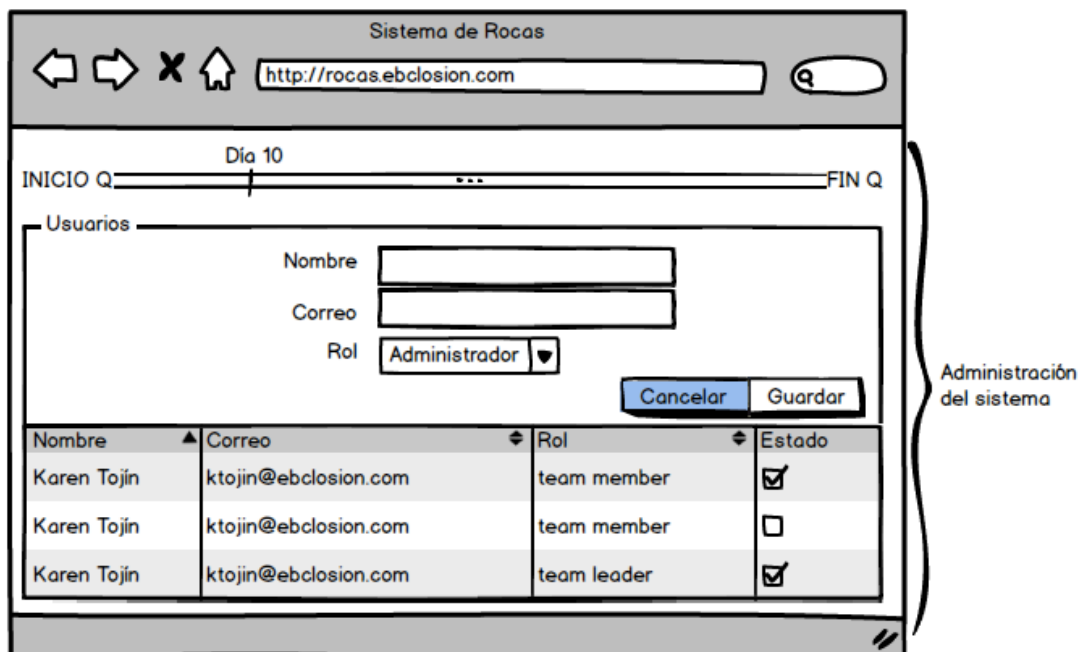


Figura no. 19: Sección de reportes de avances

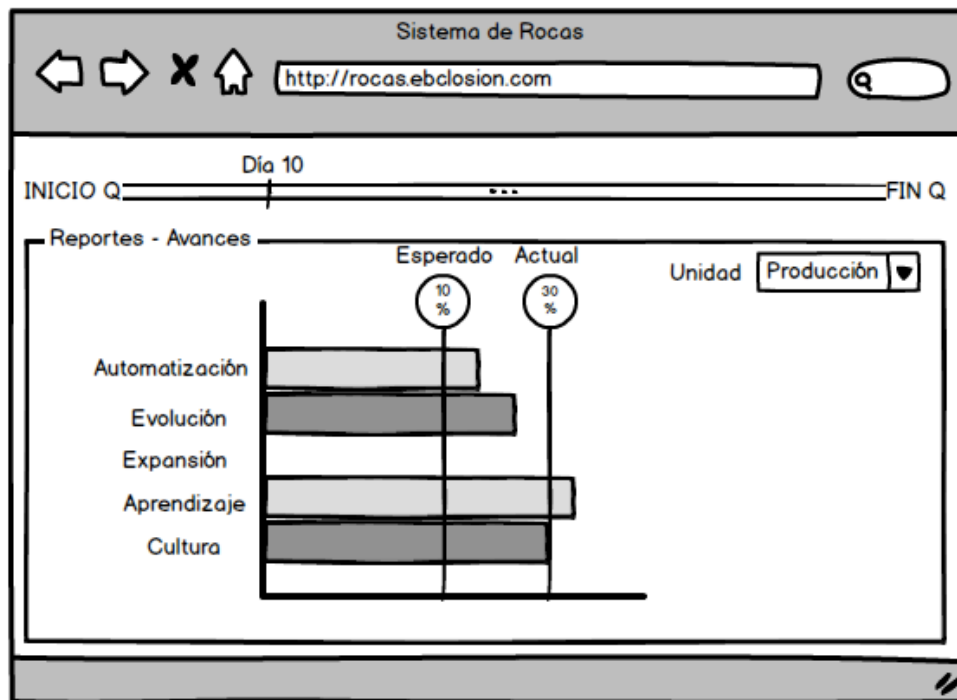
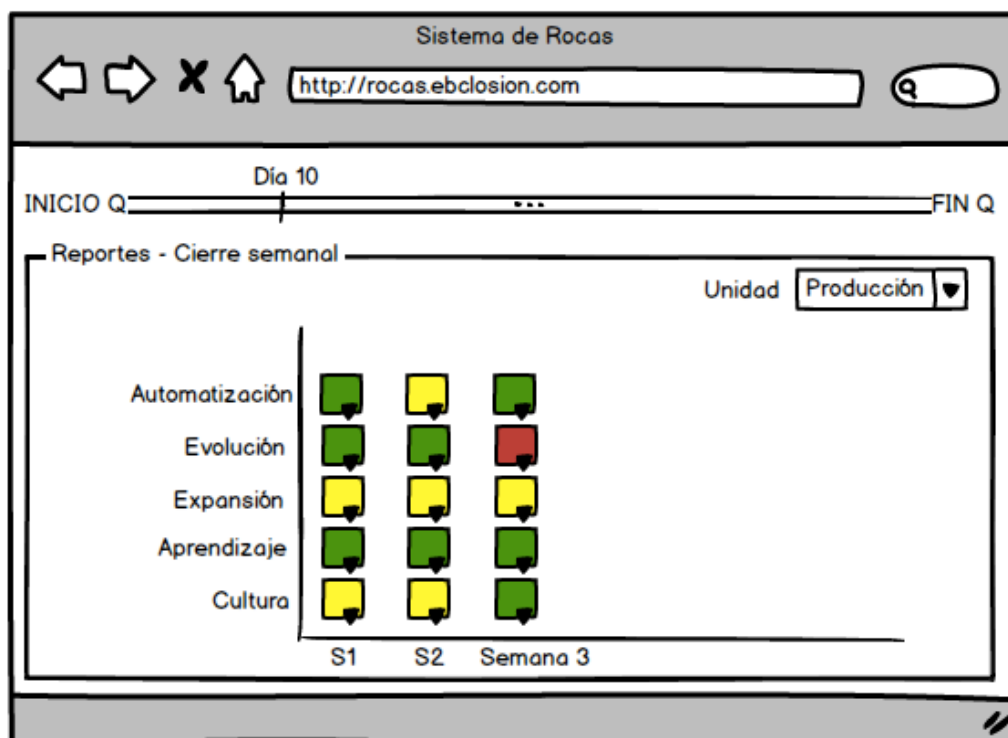


Figura no. 20: Sección de reportes de cierre semanal



VI. IMPLEMENTACIÓN (FASE 3 A 5)

El desarrollo real de la aplicación, es decir, la implementación es realizado durante las fases tres a cinco de la metodología DSDM. Esta fase explica el desarrollo final del sistema, basándose en los requerimientos obtenidos de la fase de análisis y la información y planificación determinada en la fase de diseño. La implementación del sistema tuvo una duración de aproximadamente tres meses.

La elaboración e implementación del sistema fue realizada en cinco iteraciones. En cada una de ellas se realizó labor de diseño funcional y su correspondiente desarrollo. También, en cada iteración se procuró recibir retroalimentación de los usuarios, ya que es más fácil realizar cambios y ajustes sobre un módulo o fase que sobre el sistema terminado. A continuación se presenta un detalle de cada iteración y lo que fue realizado en cada una de ellas.

A. Iteración 1

Esta es la primera iteración del proyecto. En ella se da inicio al desarrollo e implementación del sistema de rocas para *EBclosion*. Para cada iteración se definieron sus propósitos utilizando el método de priorización de MoSCow. Las metas definidas para esta iteración son:

- Creación del diseño gráfico del sistema basado en los prototipos propuestos en la fase de diseño.
- Creación del proyecto en *Symfony* y maquetado del sitio con el diseño gráfico final.
- Creación de la base de datos del sistema.
- Conexión del proyecto con la base de datos del sistema.

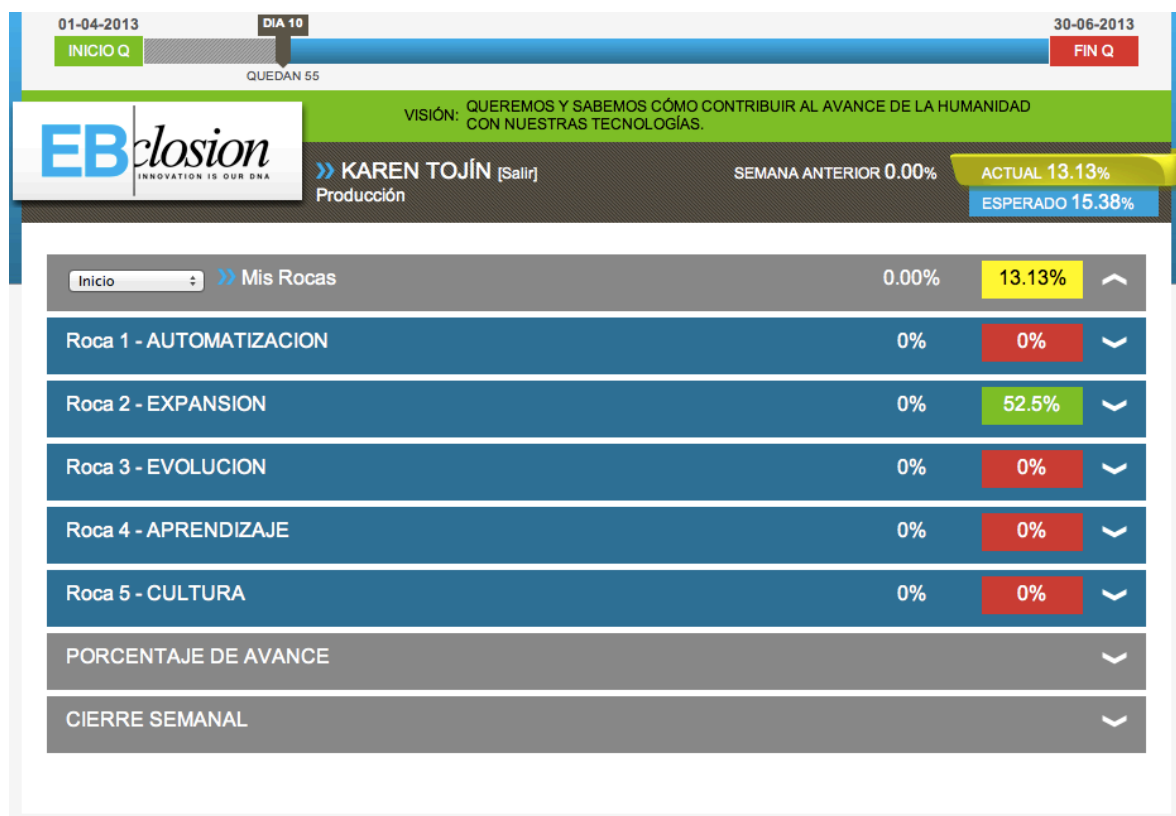
1. Diseño gráfico del sistema. El entorno visual del sistema fue realizado por un diseñador gráfico que labora en *EBclosion*. El sistema fue basado en los prototipos presentados en la fase de diseño. Para el sistema se seleccionaron colores que representan a la empresa y que tengan un significado positivo y motivador. El diseño de los reportes también fue elaborado en esta sección, seleccionando reportes que transmitan de forma eficiente lo que gerencia desea evaluar con el sistema.

2. Creación del proyecto en *Symfony* y maquetado del sitio. Antes de codificar los requerimientos solicitados, es necesario configurar el ambiente en el que se estará desarrollando. Para el inicio de este proyecto ya se contaba con todas las herramientas necesarias configuradas, por lo que lo único que fue necesario para darle inicio oficialmente al desarrollo del proyecto fue su creación utilizando el *framework Symfony*. El proyecto fue creado en el servidor de desarrollo con el comando:

```
./ symfony generate:project rocks
```

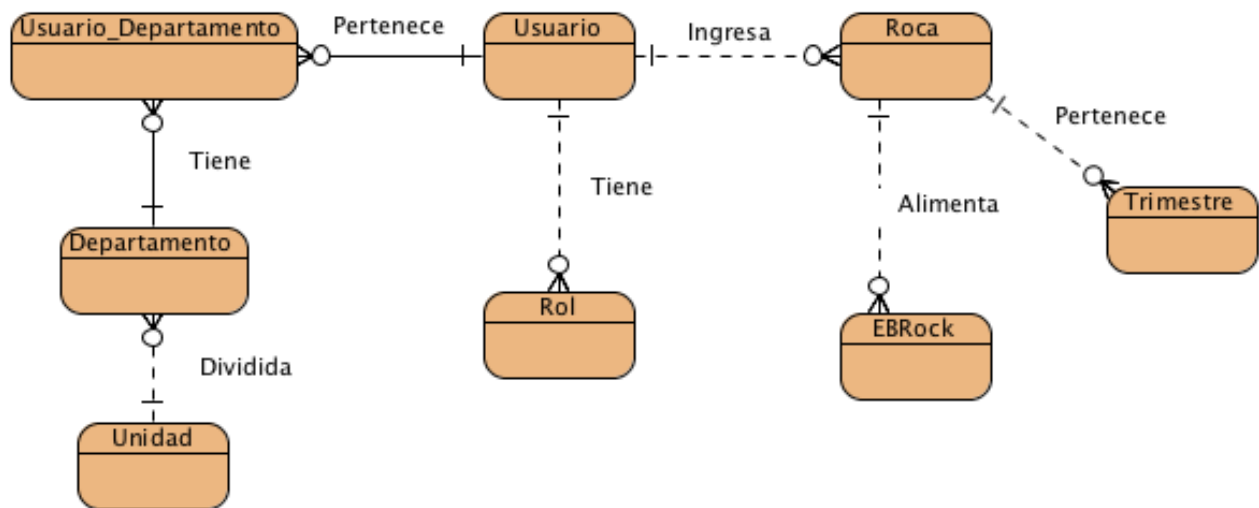
Cuando un proyecto de *Symfony* es generado, se genera también el árbol de directorio que compone el *framework*.. Sobre la plantilla principal fue maquetado el sitio y fueron creados los estilos que se utilizarían para el resto de la aplicación.

Figura no. 21: Diseño gráfico aprobado para el sistema de rocas



3. Creación de la base de datos del sistema. Cuando ya se tenía la aplicación creada con el *framework* se procedió a crear la base de datos que sería utilizada por el sistema, así como los usuarios creados para accederla. Para la creación de la base de datos del sistema fueron utilizadas ideas de la base de datos del sistema anterior y las mejoras propuestas para la segunda versión del sistema. A continuación se presenta un diagrama entidad relación de la base de datos del sistema de rocas.

Figura no. 22: Diagrama entidad relación de la base de datos

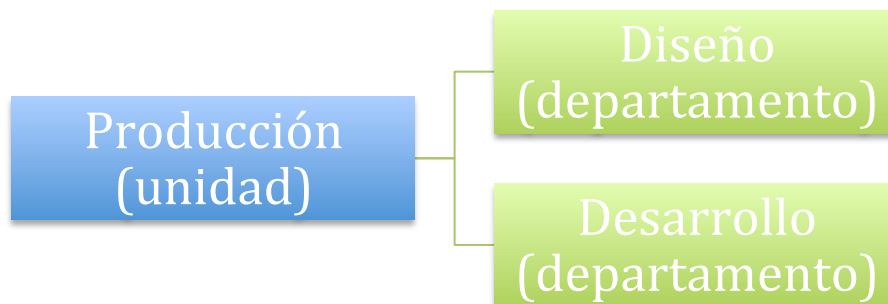


Las principales entidades de la base de datos del sistema son:

- **Usuario:** persona que labora dentro de la empresa que utilizará el sistema de rocas. Cada usuario tendrá asignado un rol que delimita los permisos y las acciones que podrá realizar en el sistema. A continuación se definen los permisos:
 - **Rol administrador:** podrá utilizar la administración del sistema, que incluye trimestres, unidades, departamentos, usuarios, EBRocas. Además podrá crear, editar y actualizar rocas de cualquier nivel. Tendrá accesos para visualizar todos los reportes, hasta filtrar por unidad y ver el detalle de cada miembro de dicha unidad.
 - **Rol líder de equipo (o líder de unidad):** podrá crear, editar y actualizar rocas de cualquier nivel. Tendrá accesos para visualizar todos los reportes, hasta filtrar por unidad y ver el detalle de cada miembro de dicha unidad.

- **Rol de miembro de equipo (o miembro de unidad):** podrá crear, editar y actualizar rocas de tercer nivel en adelante. Requiere que el líder de equipo ingrese con anterioridad rocas de primer y segundo nivel. Podrá ver los avances de su unidad únicamente.
- **Unidad:** división o área dentro de la empresa. Existen doce unidades que son: finanzas, entretenimiento móvil (ME), IT, producción, Globalogik, AudioLab, administración de oficina, servicios legales, Zootropic, proyectos macro, expansión y gerencia. Algunas de estas unidades están divididas en departamentos internos, aunque a la vista de gerencia sea únicamente una unidad. Un ejemplo de esto es la unidad de producción, que está dividida en los departamentos de diseño y desarrollo.
 - **Departamento:** subdivisión de algunas unidades en partes más pequeñas y específicas.

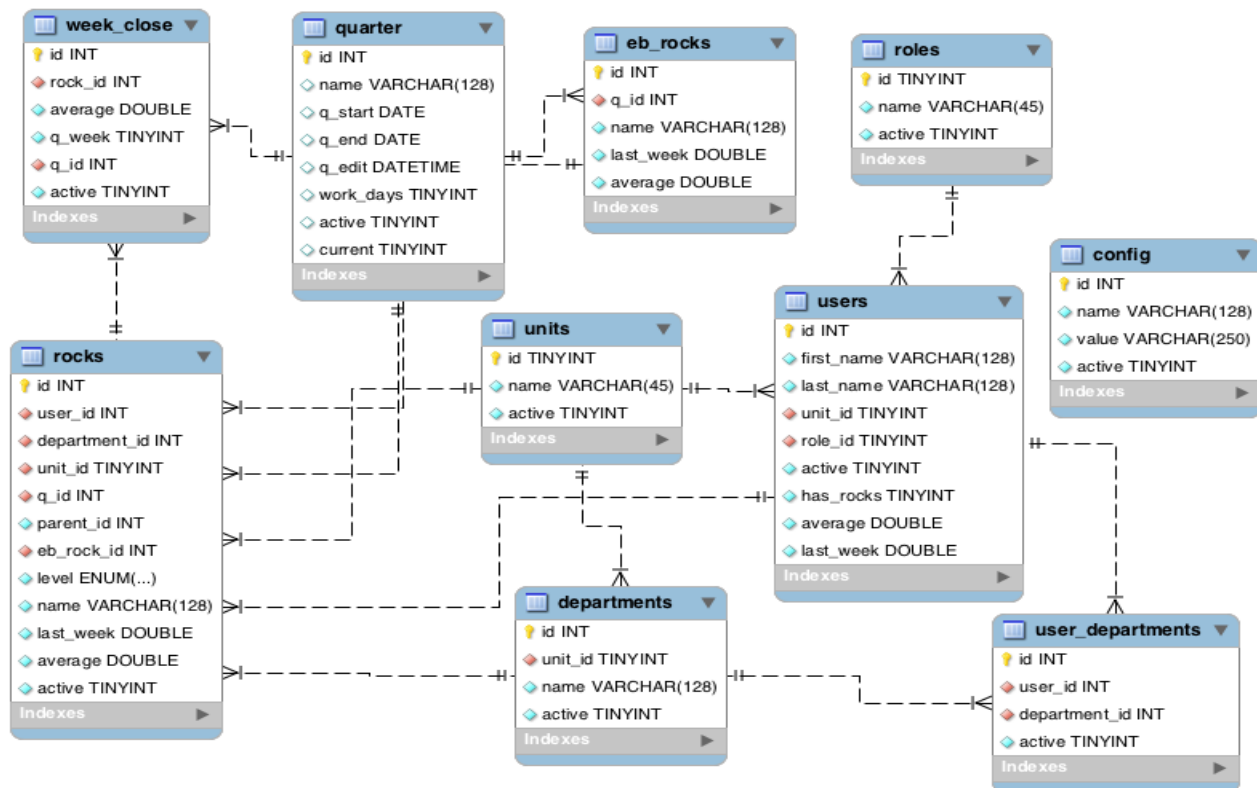
Figura no. 23: Ejemplo de unidades y departamentos dentro de la empresa



- Cada usuario está asociado a uno o más departamentos de una única unidad, esto quiere decir que, por ejemplo, un usuario no puede estar asociado a finanzas y producción al mismo tiempo.
- **EBRock:** son las rocas principales, también vistas como de nivel cero, a las que alimentan el resto de rocas. Cada una de las rocas creadas por los usuarios alimentan a una de las rocas de *EBclosure*. Éstas están definidas como: automatización, expansión, evolución, aprendizaje y cultura.
- **Trimestre:** representa a la unidad temporal en la que está dividida la organización de la empresa.
- **Roca:** elemento que tiene un nombre, un porcentaje de avance, un avance de la semana anterior como historial, una roca padre y un nivel. A su vez, una roca está asociada a un usuario, que representa a la persona que creó la roca, a un trimestre, a una unidad y también a un departamento. Una roca también posee un estado, ya que puede ser colocada en pausa o borrarla del sistema.

En la sección de anexos se presenta el esquema de las entidades de la base de datos generadas por *Doctrine* en *Symfony*. A continuación se presenta el diagrama de tablas de la base de datos.

Figura no. 24: Diagrama de base de datos



Una vez que la base de datos fue creada, un esquema fue generado a partir de ella. Este esquema representará dentro del proyecto los objetos que forman a la base de datos. Esta labor es realizada por *Doctrine* dentro de *Symfony*. Su objetivo es la creación de objetos de base de datos que son manipulables dentro de la aplicación .

4. Conclusiones de la iteración. La primera iteración fue finalizada con éxito, ya que fue posible realizar todas las actividades que se tenían planeadas. No fue posible adelantar con las actividades de “podría tener”, pero las actividades incluidas en “tiene que tener” fueron realizadas en tiempo y con éxito.

B. Iteración 2

Se inició la segunda iteración con un entorno de trabajo completamente configurado y probado. En esta iteración se inicia oficialmente la programación lógica del proyecto. Las metas definidas para esta iteración son:

- Ingreso con el correo y contraseña de la empresa.
- Creación, edición y eliminación de rocas.

1. Ingreso con el correo y contraseña de la empresa. Para evitar que los usuarios del sistema tengan que interactuar y aprender diferentes contraseñas para todos los servicios, se solicitó que el ingreso al sistema de rocas fuera realizado utilizando una autenticación con Google ya que en la empresa se cuenta con Google Apps como servidor de correos.

Esto funciona también como medida de seguridad, ya que de esta manera no se manejan en la base de datos credenciales de los usuarios. Para realizar la autenticación se utilizó el API de Google *ClientLogin*, este ingreso fue creado desde la primera versión del sistema, y no fue actualizado para la segunda versión.

La autenticación con ClientLogin requiere la creación de una conexión segura y el envío del correo de Google Apps en conjunto con la contraseña a una URL brindada por Google. Sin embargo, es necesario y sumamente importante migrar a una autenticación utilizando el API OAuth 2.0 de Google, ya que el API ClientLogin fue dado de baja en el 2012. La realización de este cambio está programado como desarrollo futuro del sistema, aunque es imperativo que se realice la actualización en el sistema.















2. Creación, edición y eliminación de rocas . La creación, edición y eliminación de rocas forma parte de los pilares del sistema. Es a partir de la manipulación de esta información que se realizan las tareas más importantes y no administrativas. Para realizar esta actividad fue necesario tomar en cuenta la jerarquía de las rocas y los permisos de cada una.

Se inició esta fase con el análisis de qué tecnología se utilizaría para cargar la información en la página principal de la aplicación. Se decidió utilizar *Javascript* y *jQuery*, ya que estas librerías permitirían a la aplicación ser más flexible y dinámica. Además que en el diseño del sistema toda manipulación de las rocas debía estar cargada en la misma página, en lugar de hacer saltos a otras.

El resultado final de esta sección de la segunda iteración fue un sistema en

donde pueden agregarse elementos de forma dinámica, sin recargar la página tras cada acción.

Figura no. 25: Ingreso de rocas del sistema de *EBclosion*

TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO	 	AVANCE
Proyectos	0%	53.33%	Desarrollo	 	
Red de Vendedores CEMPRO	0%	0%	Desarrollo	 	
Inventarios CEMPRO	0%	53.33%	Desarrollo	 	
Definición de objetivos	0%	<input type="text" value="100.00"/> %	Desarrollo	 	
Definición de equipo de desarrollo	0%	<input type="text" value="100.00"/> %	Desarrollo	 	

3. Conclusiones de la iteración. La segunda iteración fue finalizada con éxito, ya que fue posible realizar todas las actividades que se tenían planificadas. Se cumplió con el requisito del negocio de permitir una visualización granular de las rocas y que se mantuviese el protagonismo de las EBRocas. No fue posible adelantar con la actividad del cálculo de avances.

C. Iteración 3

En la tercera iteración debía realizarse una de las partes más complejas del desarrollo del sistema de rocas, el cálculo de los porcentajes de avance por cada roca. Debido a la complejidad de la tarea, su realización fue la única meta planificada para esta iteración.

1. Cálculo de avances. El cálculo de los avances de las rocas es acumulativo por nivel. Es decir que si una roca tiene N cantidad de rocas hijas, su porcentaje de avance será el promedio de sus rocas hijas. Este proceso se repite desde el nivel cuatro hasta llegar a las rocas de primer nivel. A continuación se presenta un diagrama en donde se explica el cálculo de los avances en las rocas.

Figura no. 26: Cálculo de avance de rocas



En la figura no. 25 se muestra cómo las rocas de tercer nivel (en color celeste) alimentan a las rocas de segundo nivel (color morado) y el promedio de estas afecta a las rocas de primer nivel (color verde). En un nivel más alto, las rocas de *EBclosion* (color azul) se ven afectadas por todas las rocas asociadas a ella.

De esta forma se calcula el porcentaje de avance de las rocas desde los niveles inferiores hasta los superiores. Es importante mencionar que cada avance en una roca afecta el porcentaje de todas sus rocas padre.

Otro aspecto que debió tomarse en cuenta fue que como en el ejemplo, si la roca de nivel dos tiene rocas hijas, no puede actualizarse directamente, ya que su porcentaje de avance depende de sus rocas hijas.

Debido a experiencia previa, si cada vez que se realiza una actualización de una roca, todos los valores de sus rocas padre son almacenados en base de datos, el tiempo de respuesta disminuye considerablemente. Este problema ocurría con la primera versión del sistema, por lo que para esta versión la actualización en base de datos es realizada únicamente sobre la roca que está siendo actualizada.

Aún así, debe calcularse el porcentaje para las rocas padre, por lo que se creó un *script* que corre tres veces al día y que realiza y almacena estos cálculos sin necesidad de esperar a que la plataforma reaccione si esta función se realizara en tiempo real.



2. Conclusiones de la iteración. La tercera iteración fue la más importante, debido a que en ella se desarrolló el cálculo de los avances de las rocas. Fue también la iteración más larga, ya que era necesario realizar pruebas para corroborar que los cálculos fuesen correctos. El cálculo de los avances fue la única actividad realizada durante esta iteración. Se cumplió el requisito de mejorar el tiempo de respuesta con la creación del script que actualiza las rocas padre por separado.

D. Iteración 4


Para la cuarta iteración se planificó la elaboración del área administrativa para el sistema de rocas. En esta área se incluyó el manejo de usuarios, trimestres, unidades, EBRocas y configuración general del sistema. Es importante mencionar que todas las eliminaciones son realizadas con una actualización para que los campos ya no sean visibles, no se realiza una eliminación real en la base de datos. La elaboración de los módulos administrativos del sistema fue la única meta planificada para esta iteración.


1. Módulo de usuarios. En la elaboración del módulo de usuarios se incluyó la creación, visualización, actualización y eliminación, CRUD por sus siglas en inglés, de los mismos. En la elaboración del módulo de usuarios se incluyó también la asociación de un usuario a un rol, lo que determina sus permisos en el sistema. Otro aspecto importante de la creación de usuarios incluye la asociación de usuarios de unidades y departamentos de la empresa, lo que determinará la información que puedan visualizar.

Figura no. 27: Módulo de usuarios

Ingreso Usuarios	
Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Email:	<input type="text"/>
Rol:	Seleccione un rol ▾
Unidad:	Seleccione una unidad ▾
Departamento:	Seleccione un departamento ▾
Activo:	<input type="checkbox"/>
Posee rocas:	<input type="checkbox"/>
 	

Usuarios						
NOMBRE ▾	APELLIDO ▾	EMAIL ▾	ROL ▾	ESTADO ▾	ROCAS ▾	EDITAR
Karla	Cruz	kcruz@ebclosion.com	Team Leader	Activo	Acceso	EDITAR
Abimael	López Cano	alopez@ebclosion.com	Team Member	Activo	Acceso	EDITAR
Eduard	Torres	etorres@ebclosion.com	Team Member	Activo	Acceso	EDITAR
Javier Ernesto	Rincón Cuellar	jrincon@ebclosion.com	Team Leader	Activo	Acceso	EDITAR
Miriam	Moran	mmoran@ebclosion.com	Team Member	Activo	Acceso	EDITAR






2. Módulo de unidades y departamentos. El módulo de unidades y departamentos permite crear una nueva unidad a la empresa o eliminar una existente. También se permite crear subdivisiones de la unidad dentro de la misma. Estas subdivisiones son invisibles al nivel gerencial y son utilizados para que un líder de área tenga una mejor organización sobre su unidad.

Un usuario puede pertenecer a uno o a todos los departamentos de la unidad. La creación de subdivisiones o departamentos es opcional, ya que existen casos en donde crear subdivisiones es innecesario. La importancia de los departamentos es evidenciada en los permisos, ya que aunque un usuario esté asociado a una unidad, éste podrá ver únicamente los departamentos a los que esté asignado.

Figura no. 28: Módulo de unidades y trimestres

Ingreso Unidades y Departamentos

Unidad:	<input type="text" value="Producción"/>	<input style="border: none; background: none; width: 20px; height: 20px; vertical-align: middle;" type="button" value="+"/>	
Departamento:	<input type="text" value="Desarrollo"/>	<input checked="" type="checkbox"/>	
Departamento:	<input type="text" value="Diseño"/>	<input checked="" type="checkbox"/>	
Departamento:	<input type="text" value="PM"/>	<input checked="" type="checkbox"/>	



Unidades y Departamentos

Finanzas	EDITAR
ME	EDITAR
IT	EDITAR
Producción	EDITAR
Globalogik	EDITAR


NOMBRE	ESTADO
Dirección	Activo
Gestión	Activo
Prod, dev y QA	Activo

3. Módulo de trimestres. La elaboración del módulo de trimestres incluyó la creación, visualización, actualización y eliminación de cada trimestre. El aspecto más importante de este administrador es el campo de trimestre actual, ya que al seleccionar el trimestre que se está creando como actual, cualquier trimestre que estuviese seleccionado automáticamente cambiará. El trimestre que esté seleccionado como actual será el único que podrá ser visto por todos los usuarios.

Figura no. 29: Módulo de trimestres


Ingreso Trimestres


Nombre:	<input type="text"/>
Fecha Inicio:	<input type="text"/>
Fecha Fin:	<input type="text"/>
Fecha de Edición:	<input type="text"/>
Días Laborales:	<input type="text"/>
Activo:	<input type="checkbox"/>
Trimestre Actual:	<input type="checkbox"/>

Trimestres

NOMBRE	INICIO	FIN	EDICIÓN	DÍAS LABORALES	ESTADO	ACTUAL	EDITAR
Q1 2013	01-01-2013	31-03-2013	18-01-2013	65	Activo	Q NO Actual	EDITAR
Q2 2013	01-04-2013	30-06-2013	18-04-2013	65	Activo	Q Actual	EDITAR
Q3 2013	01-07-2012	30-09-2013	18-07-2012	65	Activo	Q NO Actual	EDITAR
Q4 2013	01-10-2012	31-12-2012	18-10-2012	65	Activo	Q NO Actual	EDITAR





4. Módulo de EBRocas. El módulo de EBRocas permite la creación de las rocas asociadas a cada trimestre y las rocas que serán alimentadas por las ingresadas por los usuarios. Ya que por lo general las EBRocas no cambian a lo largo del tiempo, dentro del módulo, existe la posibilidad de duplicar las rocas que fueron ingresadas en un trimestre a otro.

Figura no. 30: Módulo de EBRocas

Ingreso EBRocas

Trimestre: Seleccione un trimestre ▾






Rocas EBClosion Q2 2013 ▾

NOMBRE	EDITAR
AUTOMATIZACION	EDITAR
EXPANSION	EDITAR
EVOLUCION	EDITAR
APRENDIZAJE	EDITAR
CULTURA	EDITAR

Duplicar para el trimestre:
Seleccione un trimestre ▾


5. Conclusiones de la iteración. Esta iteración no supuso mayor complicación, ya que incluía únicamente la creación de administradores para la configuración del sistema, es decir los aspectos que deben ser especificados para que el sistema funcione correctamente. En esta iteración también fue posible adelantar con la sección de reportes que fue finalizada en la siguiente iteración.

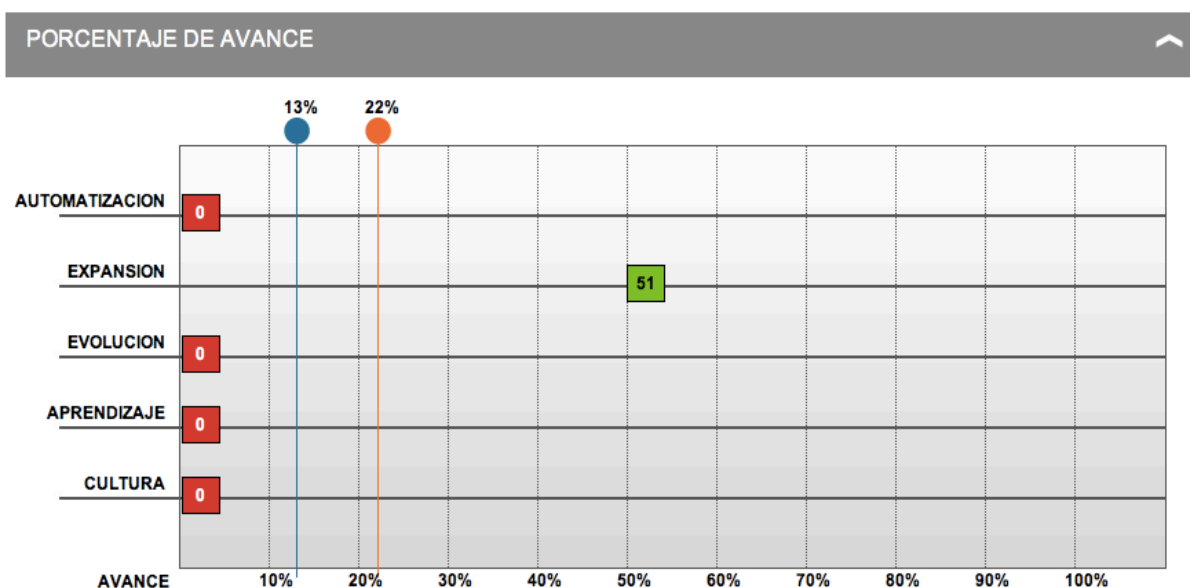
E. Iteración 5

La quinta y última iteración del sistema consistió en otro de los aspectos más importantes que debía incluirse en la segunda versión del sistema de rocas. Este aspecto es la sección de reportes. Con la elaboración de esta sección se busca apoyar a la gerencia en la toma de decisiones, ya que ofrece una forma más sencilla de visualizar los avances en las rocas. La sección de reportes apoya también a los líderes de área a evaluar de forma más eficiente a cada miembro del equipo.

1. Sección de reportes. Los reportes generados por el sistema se dividen en dos: un área para visualizar el avance de las rocas en el momento y otra para visualizar el cierre semanal de cada área de la empresa. Se buscaron soluciones gratuitas que generaran gráficas, pero ninguna se adaptó a los requerimientos de las gráficas para el sistema. Por esta razón los reportes fueron generados manualmente, con imágenes y estilos para simular las gráficas.

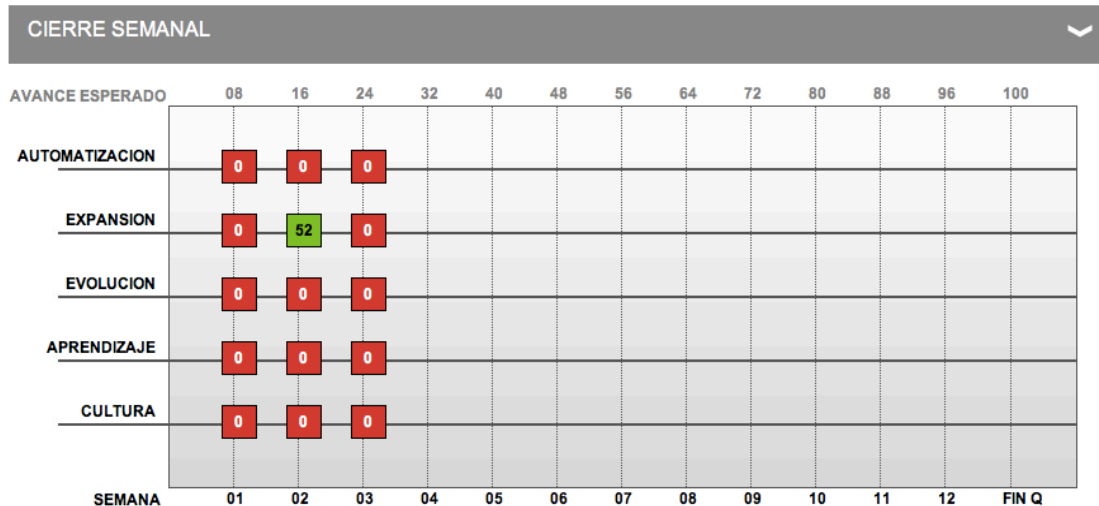
Los reportes se encuentran disponibles en todas las secciones del sistema, desde la vista individual, la vista por unidad, la vista de toda la empresa y una sección aparte para visualizar únicamente los reportes. En la sección de reportes se pueden realizar filtros para mostrar información de determinada unidad o determinada EBRoca.

Figura no. 31: Reportes de porcentaje de avance



Los reportes de cierre semanal muestran el cierre que tuvieron las diferentes unidades semanalmente. Este reporte es utilizado para las reuniones semanales de avances que sostienen los líderes de área con gerencia y con sus equipos. Estos reportes permiten evaluar rápidamente en qué EBRocas y proyectos se debe trabajar durante la semana.

Figura no. 32: Reportes de cierre semanal



2. Manuales del sistema. Se elaboraron tres manuales del sistema de rocas, cada uno destinado a los tipos de usuarios que utilizarían el sistema. En los manuales se cubren los aspectos que cada usuario puede realizar y cómo realizarlos. Los manuales creados fueron enviados a los usuarios cuando se dio por terminado el proyecto.

3. Conclusiones de la iteración. Con esta iteración se dio por terminado el proyecto del sistema de rocas. El sistema fue terminado con éxito, ya que se llegó a los requerimientos deseados en el tiempo que se tenía estipulado al inicio del desarrollo.

VII. DESPLIEGUE, PRUEBAS Y MANTENIMIENTO (FASES 6 Y 7)

A. Despliegue

Todo el desarrollo del sistema de rocas fue realizado en el servidor de desarrollo de la empresa, por lo que al terminarlo fue necesario enviar el proyecto al servidor de producción para que pudiera ser accedido por toda la empresa. Esta acción fue realizada utilizando un comando de *Symfony*. Es por esto que la utilización de un marco de trabajo es muy recomendada.

Antes de enviar y poner en marcha el proyecto en el servidor de producción fue necesario crear la base de datos en ese servidor. Una vez que el proyecto fue enviado, el único paso restante fue la configuración del dominio para poder acceder al sistema. El dominio fue configurado y nombrado como rocas.ebclosion.com.

B. Pruebas

Durante el desarrollo del proyecto diferentes pruebas fueron realizadas para que la funcionalidad del proyecto fuese correcta. *Symfony* ofrece la elaboración de pruebas unitarias y pruebas funcionales. Ambas pruebas fueron realizadas durante el desarrollo del sistema de rocas.

1. Pruebas unitarias. Las pruebas unitarias verifican que cada función desarrollada trabaje correctamente. Las pruebas unitarias deben desarrollarse de la forma más independiente posible una de otra para evitar propagar un error no identificado o corregido. El enfoque que provee *Symfony* es pragmático, por lo que no recomienda hacer pruebas sobre cada una de las funciones desarrolladas, pero sí realizar pruebas sobre las funciones importantes.

Symfony provee una librería llamada *lime* para facilitar la escritura de pruebas unitarias. *Lime* provee diferentes métodos que facilitan realizar pruebas. Entre estos métodos se pueden probar excepciones, igualdad de valores, tipos de datos, entre otros.

Fueron realizadas pruebas unitarias para el ingreso de rocas, su almacenamiento y principalmente para los cálculos de los avances. También fueron utilizadas para la evaluación del módulo administrativo del sistema de rocas. Para la elaboración de pruebas unitarias utilizando Doctrine no se configuró una base de

datos dedicada para pruebas, se utilizó la base de datos de desarrollo.

2. Pruebas funcionales. Las pruebas funcionales son una herramienta que ayuda a probar una aplicación desde que el usuario realiza una acción, hasta lo que se le devuelve una vez realizada dicha acción. Es decir, que las pruebas funcionales ejecutan un escenario del caso de uso que se acaba de implementar. Las pruebas funcionales en *Symfony* permiten simular escenarios que pueden ser probados cada vez que un caso de uso es implementado.

En *Symfony*, las pruebas funcionales se ejecutan a través de un navegador especial, implementado por la clase *sfBrowser* que actúa como un navegador adaptado para la aplicación y está directamente conectada a ella, sin necesidad de un servidor Web. Le da acceso a todos los objetos *Symfony* antes y después de cada petición, brindando la oportunidad de inspeccionarlos y hacer los controles que desee mediante programación.

Para el sistema de rocas las pruebas funcionales fueron realizadas para el ingreso de rocas y para la ejecución del sistema de reportes. Debido a que la elaboración de pruebas funcionales es más extensa que las pruebas unitarias, las pruebas funcionales fueron realizadas únicamente para las actividades críticas del sistema.

C. Mantenimiento

El mantenimiento del sistema fue realizado gracias a la retroalimentación de los usuarios una vez fue puesto en producción. El mantenimiento consistió en el monitoreo del sistema y la solución de problemas reportados por los usuarios en el primer trimestre del uso del mismo. En este primer trimestre de uso del sistema se planteó el desarrollo futuro que el sistema tendrá.

VIII. IMPACTO EMPRESARIAL

El desarrollo del nuevo sistema de rocas le trajo muchas ventajas a *EBclosion*, ya que demuestran ser una empresa que está buscando mejorar constantemente. Esto es comprobado con el desarrollo de sistemas como el presentado en este trabajo, ya que les permite automatizar procesos y obtener mejores resultados de aspectos y evaluaciones importantes para la empresa.

La creación del sistema de rocas implicó un gran avance desde que se inició con el desarrollo de un sistema en línea en el 2011. Y el avance fue más evidente con el desarrollo de la segunda versión del sistema, ya que no sólo se mejoró visualmente el sistema anterior, sino que se incluyeron también nuevos aspectos que le agregaron valor al sistema y apoyaron el proceso de toma de decisiones dentro de la empresa.

A. Desarrollo futuro

Cuando el proyecto fue finalizado se hizo evidente que era necesario agregar funcionalidad al sistema de rocas. Todos los requerimientos obtenidos durante las fase de aceptación y el mantenimiento fueron informados a los encargados del proyecto, pero aún no han sido calendarizados ni forman parte del alcance de este proyecto.

IX. CONCLUSIONES

1. El sistema de rocas desarrollado permite el manejo y evaluación de las metas para la empresa *EBclosion*, ya que además de realizar el ingreso y actualización de las rocas, el sistema cuenta con una sección de reportes que le da valor agregado respecto a las versiones anteriores.
2. Gracias a la constante retroalimentación de la interfaz gráfica por medio de los prototipos diseñados, se creó un sistema amigable y eficiente que permite el registro de las rocas de la empresa.
3. Con el diseño de la base de datos, la asociación usuarios a unidades y departamentos y la creación de roles se logró que en el sistema se visualice únicamente lo pertinente a cada unidad de trabajo.
4. El implementar un sistema con autenticación de Google permite que los usuarios se sientan identificados con el sistema y facilita el ingreso de los mismos al sistema de rocas.
5. Al realizar un desarrollo estándar para el sistema de rocas, se garantiza que la forma de medir el progreso de en las rocas sea igual para cada unidad de la empresa.
6. La creación de un sistema de reportes personalizado permite evaluar de una forma más eficiente el progreso semanal de cada unidad de la empresa y de todas las personas que laboran en ella.

X. RECOMENDACIONES

1. En todo desarrollo de software, lo más importante es tomar en cuenta el objetivo que el negocio quiere alcanzar con el sistema que vaya a realizarse.
2. Utilizar un marco de trabajo permite simplificar el desarrollo de un proyecto, al incluir funciones predefinidas y un gran esquema de trabajo.
3. La elaboración de pruebas dentro del sistema permiten mejorar la calidad del código y del producto final.
4. Verificar la vigencia de los API's de ingreso de Google antes de implementarlos, para evitar utilizar productos sin soporte.
5. Proponer a la empresa migrar a la última versión de los marcos de trabajo, para utilizar la última tecnología estable disponible e incluir más herramientas de trabajo.
6. El uso de metodologías como DSDM permite entregar productos de calidad cuando se tiene un tiempo limitado.
7. La elaboración de prototipos permite a los usuarios finales participar en el proceso de desarrollo del sistema y dar su retroalimentación en una etapa temprana del desarrollo del proyecto.
8. Tomar en cuenta la retroalimentación de los usuarios finales para el desarrollo de un sistema.

XI. BIBLIOGRAFÍA

- [1] Artes Plásticas. Estudio de factibilidad y proyectos. 2012. En línea: <http://estudiodefactibilidadyproyectos.blogspot.com/2010/09/factibilidad-y-viabilidad.html>
- [2] Bodoff, S., Green, D., Haase, K., Jendrock, E., Pawlan, M. and Stearns, B. 2002. *The J2EE Tutorial*. Addison Wesley.
- [3] Bowman, David. 2009. What is a Prototyping Methodology? En línea: <http://www.information-management-architect.com/prototyping-methodology.html>
- [4] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M. 1996. *Pattern-Oriented Software Architecture*. John Wiley and Sons. ISBN 0-471-95869-7.
- [5] Carnegie Mellon. 2012. CMMI for Development, Version 1.3. 470 pp. En línea: <http://www.sei.cmu.edu/reports/10tr033.pdf>
- [6] Clifton, Marc; Dunlap, J. 2003. What is DSDM? En línea: <http://www.codeproject.com/Articles/5097/What-Is-DSDM>
- [7] Creative & Cultural Skills. MoSCoW prioritisation. 2013. En línea: <http://business-survival-toolkit.co.uk/stage-four/prioritisation/moscow-prioritisation>
- [8] Davies, Rachel. Agile Alliance Member. 2004. DSDM explained. 17 pp. En línea: <http://www.agilexp.com/presentations/DSDMexplained.pdf>
- [9] Diccionario de la Real Academia Española. Factibilidad. En línea: <http://lema.rae.es/drae/?val=factibilidad>
- [10] Doctrine Project. About Doctrine. 2013. En línea: <http://www.doctrine-project.org>
- [11] DSDM Consortium . 2012. DSDM. En línea: <http://www.dsdm.org>
- [12] Fabien Potencier. Symphony at a Glance. 2013. En línea: <http://symfony.com/symfony-at-a-glance>
- [13] Google. ClientLogin in the Google Data Protocol Client Libraries. 2012. En línea. <https://developers.google.com/gdata/docs/auth/clientlogin>
- [14] Harnish, Verne. 2002. *Mastering the Rockefeller Habits*. Gazelles, Inc. 176 pp.

- [15] Project Management Institute. 2012. PMI. En línea: <http://www.pmi.org>
- [16] Rouse, Margaret. Object-Relational Mapping (ORM). 2013. En línea: <http://searchwindevelopment.techtarget.com/definition/object-relational-mapping>
- [17] Sensio Labs. Day 8: The Unit Tests. 2012. En línea: http://symfony.com/legacy/doc/jobeeet/1_4/en/08?orm=Doctrine
- [18] Sensio Labs. Day 9: The Functional Tests. 2012. En línea: http://symfony.com/legacy/doc/jobeeet/1_4/en/09?orm=Doctrine
- [19] Shklar, L. and Rosen, R. 2003. *Web Application Architecture: Principles, Protocols and Practices*. John Wiley and Sons. ISBN 0-471-48656-6.
- [20] Soegaard, Mads. InteractionDesignFoundation. 2012. Prototyping. En línea: <http://www.interaction-design.org/encyclopedia/prototyping.html>

XII. GLOSARIO

API (*Application Programming Interface*): Interfaz de programación de aplicaciones, es un conjunto de funciones y procedimientos que ofrece un software para ser utilizado como una capa de abstracción.

CRUD (*Create, read, update, delete*): Acrónimo de crear, leer, actualizar y borrar, por las palabras en inglés. Se utiliza para referirse a las funciones básicas en base de datos y persistencia de software.

Doctrine: Es una serie de librerías de PHP que se enfocan en proveer persistencia y un ORM a proyectos de software. Además del ORM, proveen una capa de abstracción de datos para los sistemas en los que se utilizan.

ERP (*Enterprise Resource Planning*): Sistemas de planificación de recursos empresariales, son sistemas de información gerenciales que manejan la información asociada con los aspectos de operaciones y producción del negocio.

Framework (marco de trabajo): Es una estructura definida que cuenta con diferentes herramientas, librerías y buenas prácticas que sirven de base para el desarrollo de software. Proveen módulos y facilidades para limitar el tiempo desarrollo en aspectos repetitivos al desarrollar un sistema, con el fin de enfocarse en la lógica del mismo.

HTML (*HyperText Markup Language*): Lenguaje marcado de hipertexto. Es el lenguaje predominante en la elaboración de páginas web. Se escribe en forma de etiquetas y se usa para describir y traducir la información a texto plano.

MVC (*Modelo Vista Controlador*): Es un patrón de arquitectura de software que separa los datos, la lógica del negocio y la interfaz de usuario de una aplicación. Se divide en los componentes del modelo, que es la representación de la información; la vista, que se refiere a la interfaz de usuario; y el controlador, que responde a los eventos generados por el usuario y hace un llamado a los modelos.

MySQL: Es un sistema de gestión de bases de datos relacional, *multithread* y multiusuario de software libre licenciado para GNU GPL y productos privativos. Es uno de los gestores de bases de datos más utilizados.

ORM (*Object Relational Mapping*): El mapeo objeto-relacional es una técnica de programación para convertir datos entre el sistema de tipos utilizado por un lenguaje de programación orientado a objetos y el sistema de tipos utilizado por una base de datos relacional. Utiliza un motor de persistencia que crea un objeto a partir de una entidad en la base de datos, utilizando sus atributos como propiedades y sus relaciones como relaciones entre objetos.

PHP (*PHP Hypertext Pre-processor*): es un lenguaje de programación de código del lado del servidor, diseñado principalmente para desarrollo web.

POO (*Programación Orientada a Objetos*): Es un paradigma de programación que usa los objetos, usualmente definidos por clases, y sus interacciones, relaciones entre objetos, para diseñar aplicaciones.

Script: Es un programa que se almacena como texto plano que realiza determinado funcionalidad dentro de un sistema.

Timeboxing: Es una técnica utilizada en planeación de proyectos que divide el calendario de actividades en períodos de tiempo, en donde cada período cuenta con entregables, fechas límite y presupuesto. Se enfoca en fijar el tiempo para el desarrollo del proyecto.

Toolbox (caja de herramientas): Es un conjunto de herramientas o funciones empaquetadas, que implementan funcionalidades que al núcleo del sistema le faltan.

URL (*Uniform Resource Locator*): Localizador uniforme de recursos. Es una secuencia de caracteres que se utiliza para nombrar recursos en internet.

XIII. ANEXOS

A. Historias de usuario para los casos de uso

Tabla no. 6: Historia de usuario no. 1 – Creación de rocas de primer y segundo nivel

Historia de usuario no. 1	Nombre: Creación de rocas de primer y segundo nivel
Iteración asignada: 2	Actores: Usuario con rol de líder de equipo
Prioridad en el negocio: Alta	Riesgo de desarrollo: Bajo
Descripción: El líder de cada equipo podrá crear las rocas de primer y segundo nivel que serán la base de las rocas para el resto de su equipo.	
Precondiciones: Debe haber un trimestre seleccionado como actual. Deben estar creadas las rocas de <i>EBclosion</i> (EBRocas) y asociadas al trimestre actual.	
Post-condiciones: Las rocas de primer y segundo nivel deben estar asociadas a una EBRoca.	

Tabla no. 7: Historia de usuario no. 2 – Creación de rocas de tercer y cuarto nivel

Historia de usuario no. 2	Nombre: Creación de rocas de tercer y cuarto nivel
Iteración asignada: 2	Actores: Usuario del sistema
Prioridad en el negocio: Alta	Riesgo de desarrollo: Bajo
Descripción: Cada usuario ingresará sus rocas al sistema.	
Precondiciones: El usuario líder de equipo debe haber ingresado las rocas de primer y segundo nivel previamente.	
Post-condiciones: -	

Tabla no. 8: Historia de usuario no. 3 – Cálculo de avances

Historia no. 3	Nombre: Cálculo de avances
Iteración asignada: 3	Actores: Usuario del sistema
Prioridad en el negocio: Alta	Riesgo de desarrollo: Alto
Descripción: Las rocas del sistema son actualizadas cuando los usuarios ingresan porcentajes de avance.	
Precondiciones: Cada usuario ha ingresado rocas en el sistema.	
Post-condiciones: Las rocas de <i>EBclosion</i> son actualizadas.	

Tabla no. 9: Historia de usuario no. 4 – CRUD de unidades

Historia no. 4	Nombre: CRUD de unidades
Iteración asignada: 4	Actores: Usuario con rol de administrador
Prioridad en el negocio: Media	Riesgo de desarrollo: Bajo
Descripción: El CRUD de las unidades y departamentos del sistema es realizado.	
Precondiciones: Debe existir al menos un usuario con rol administrador en la base de datos para tener acceso a esta acción.	
Post-condiciones: Es posible asociar usuarios a unidades y departamentos.	

Tabla no. 10: Historia de usuario no. 5 – CRUD de usuarios

Historia no. 5	Nombre: CRUD de usuarios
Iteración asignada: 4	Actores: Usuario con rol de administrador
Prioridad en el negocio: Media	Riesgo de desarrollo: Bajo
Descripción: El CRUD de los usuarios del sistema es realizado.	
Precondiciones: Debe existir al menos un usuario con rol administrador en la base de datos para tener acceso a esta acción y debe estar creada al menos una unidad.	
Post-condiciones: Los usuarios pueden ingresar al sistema.	

Tabla no. 11: Historia de usuario no. 6 – CRUD de trimestres

Historia no. 6	Nombre: CRUD de trimestres
Iteración asignada: 4	Actores: Usuario con rol de administrador
Prioridad en el negocio: Media	Riesgo de desarrollo: Bajo
Descripción: El CRUD de los trimestres del sistema es realizado.	
Precondiciones: -	
Post-condiciones: Es posible seleccionar un trimestre como actual en el sistema.	

Tabla no. 12: Historia de usuario no. 7 – CRUD de EBRocas

Historia no. 7	Nombre: CRUD de EBRocas
Iteración asignada: 4	Actores: Usuario con rol de administrador
Prioridad en el negocio: Media	Riesgo de desarrollo: Bajo
Descripción: El CRUD de EBRocas del sistema es realizado.	
Precondiciones: -	
Post-condiciones: Es posible crear rocas para determinado trimestre.	

Tabla no. 13: Historia de usuario no. 8 – Sección de reportes

Historia no. 8	Nombre: Sección de reportes
Iteración asignada: 5	Actores: Usuario del sistema
Prioridad en el negocio: Alta	Riesgo de desarrollo: Alto
Descripción: La sección de reportes de cierre semanal y de avances es creada.	
Precondiciones: -	
Post-condiciones: Es posible visualizar los reportes de cada trimestre en el sistema.	

B. Manual de administrador

Manual de Rocas - **Administrador**



01- LOG IN

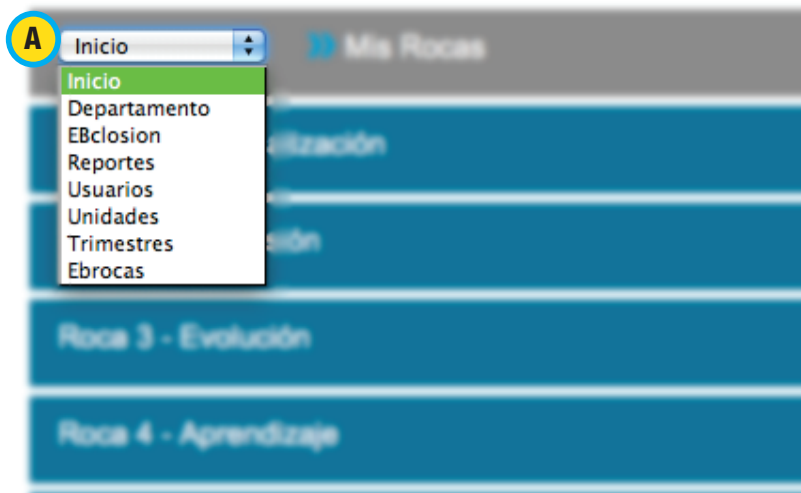
Ingresar a página web <http://rocas.ebclosure.com/>

Para entrar ingrese su usuario y contraseña de correo electrónico de EBclosure.

02- MENU DEL ADMINISTRADOR

A - El menú de la persona ingresada como Administrador desplegará los campos como los demás usuarios, pero además tendrá las secciones de:

- Usuarios
- Unidades
- Trimestres
- Ebrocas



03- USUARIOS

En esta sección se agregan usuarios nuevos, se les asigna el rol, la unidad y el departamento al que pertenecen.

Si se desea editar un usuario ya existente, se da click en EDITAR en la línea del nombre del usuario y los datos de éste aparecerán en la parte de Ingreso para poder ser modificados.

Luego de Editar la información, se presiona el botón de GUARDAR.



Si se desea eliminar un usuario, se selecciona como en los pasos de Editar, pero se presiona el botón del basurero para eliminarlo permanentemente.

Usuarios
USUARIOS
0.00%
0.00%

Ingreso Usuarios

Nombre:

Apellido:

Email:

Rol:

Unidad:

Departamento:

Activo:

Posee rocas:

Usuarios


NOMBRE	APELLIDO	EMAIL	ROL	ESTADO	ROCAS	EDITAR
Karla	Cruz	kcruz@ebclosion.com	Team Leader	Activo	Acceso	EDITAR
Abimael	López Cano	alopez@ebclosion.com	Team Member	Activo	Acceso	EDITAR
Eduard	Torres	etorres@ebclosion.com	Team Member	Activo	Acceso	EDITAR
Javier Ernesto	Rincón Cuellar	jrincon@ebclosion.com	Team Leader	Activo	Acceso	EDITAR
Miriam	Moran	mmoran@ebclosion.com	Team Member	Activo	Acceso	EDITAR

1/12
2 por pag

04- UNIDADES

En esta sección se agregan unidades nuevas bajo los Departamentos ya creados.

Si se desea editar una unidad ya existente, se da click en EDITAR en la barra azul y los datos de éste aparecerán en la parte de Ingreso para poder ser modificados.

Luego de Editar la información, se presiona el botón de GUARDAR. 

Si se desea eliminar una unidad, se selecciona como en los pasos de Editar, pero se presiona el botón del basurero para eliminarlo permanentemente.

Unidades ▾
UNIDADES
0.00%
0.00%
⤴

Ingreso Unidades y Departamentos

Unidad:

Departamento:

+
☑

🗑
💾

Unidades y Departamentos

Finanzas EDITAR ⤴

NOMBRE	ESTADO
Finanzas	Activo

ME EDITAR ▾

IT EDITAR ▾

05- TRIMESTRES

En esta sección se agregan los períodos de Trimestres y las fechas de duración (inicio - fin).


El Campo de "Fecha de Edición" es la fecha límite hasta cuando se permite el ingreso de Rocas Mayores por los TeamLeaders.

Los Días Laborales se sacan en base a los días hábiles dentro del período del Q.

Si se selecciona "Activo" ese trimestre se verá visible en el listado de Trimestres.

Si se selecciona "Trimestre Actual" el seleccionado pasará a ser el Q que todos los usuarios vean en sus sistema de Rocas.

Si se desea editar un Trimestre ya existente, se da click en EDITAR y los datos de éste aparecerán en la parte de Ingreso para poder ser modificados.

Luego de Editar la información, se presiona el botón de GUARDAR. 

Si se desea eliminar un Trimestre, se selecciona como en los pasos de Editar, pero se presiona el botón del basurero para eliminarlo permanentemente.

Trimestres							
NOMBRE	INICIO	FIN	EDICIÓN	DÍAS LABORALES	ESTADO	ACTUAL	EDITAR
Q1 2013	01-01-2013	31-03-2013	18-01-2013	65	Activo	Q Actual	EDITAR
Q2 2013	01-04-2013	30-06-2013	18-04-2013	65	Activo	Q NO Actual	EDITAR
Q3 2013	01-07-2012	30-09-2013	18-07-2012	65	Activo	Q NO Actual	EDITAR
Q4 2013	01-10-2012	31-12-2012	18-10-2012	65	Activo	Q NO Actual	EDITAR

1/1 2 por pag

05- EB ROCAS

En esta sección se agregan o modifican las Rocas de la Empresa (EB Rocas).

Primero se debe elegir del listado el Trimestre al que se desea ingresar una roca nueva y se presiona el botón de “+”.

Si se desea editar una EB roca ya existente, se da click en EDITAR y los datos de ésta aparecerá en la parte de Ingreso para poder ser modificados.

Luego de Editar la información, se presiona el botón de GUARDAR.



Si se desea eliminar una EB Roca, se selecciona como en los pasos de Editar, pero se presiona el botón del basurero para eliminarla permanentemente.

NOMBRE	EDITAR
Automatización	EDITAR
Expansión	EDITAR
Evolución	EDITAR
Aprendizaje	EDITAR
Cultura	EDITAR

C. Manual de líder de equipo

Manual de Rocas - Team Leader



01- LOG IN

Ingresar a página web <http://rocas.ebclosure.com/>

Para entrar ingrese su usuario y contraseña de correo electrónico de EBclosure.

02- PANTALLA DE INICIO

A - Vista gráfica del avance de tiempo en el Q.

B - Porcentajes esperados de avance, actual y anterior

C - Menú de navegación

D - Menú en que como Team Leader, se pueden visualizar las rocas de cada uno de sus Team Members

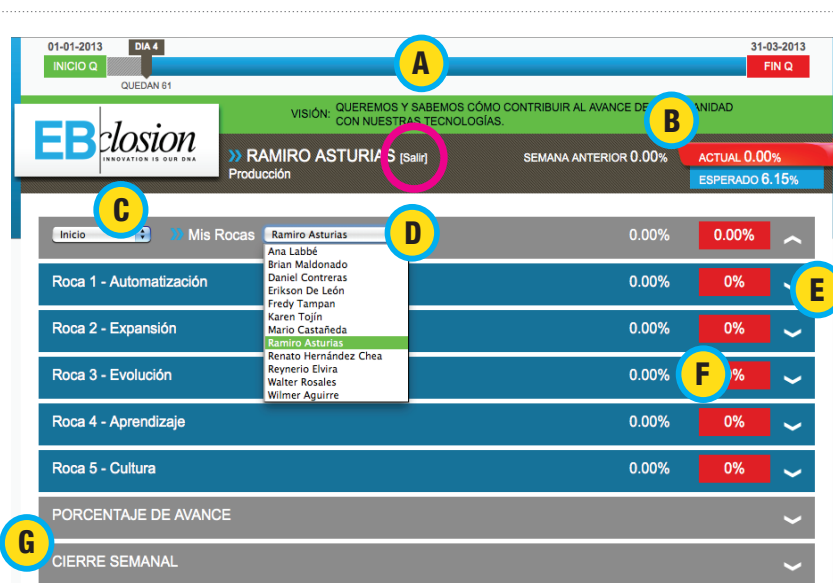
E - Abre y cierra la pestaña para mostrar el contenido de cada una

F - Vista rápida del porcentaje de cada roca con función de semáforo

G - Resúmenes de avances

» Porcentaje de Avance: muestra cuál es el porcentaje esperado (círculo naranja) y cuál es el del team member (en cuadrado y con función de semáforo)

» Cierre Semanal: muestra en cuánto debió cerrarse las rocas por semana y en cuánto lo hizo el team member.



 Link para Salir del Sistema de Rocas

03- AGREGAR ROCAS EN NIVEL 1 Y 2

El **Primer Nivel** de roca se agrega al presionar el botón de “+” en la barra negra. Sólo los Team Leaders tienen acceso a agregar este nivel de roca.

ROCA NIVEL 1

Roca 4 - Aprendizaje				0.00%	0%	⌵
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO	+	-	AVANCE
Plan de Estudio Tecnico	0.00%	0%	Desarrollo	+	-	Agregar Roca
Configuraciones Avanzadas Symphony	0.00%	0%	Desarrollo	+	-	
Android Development	0.00%	0%	Desarrollo	+	-	
Configuraciones de Kannel	0.00%	0%	Desarrollo	+	-	

Al final de las rocas Nivel 1 ya existentes, aparece un nuevo campo para ingresar la nueva roca. Se habilita un menú con las áreas que el Team Member tuviera asignadas para que pueda indicar a cuál se le asigna la roca.

Presionar el botón de “Guardar”.

Refuerzo de Basic Training	0.00%	0%	Desarrollo	+	-	
Refuerzo de Basic Training	0.00%	0%	Diseño	+	-	
			Desarrollo	+	-	
			Diseño	+	-	
			PM	+	-	

Roca 5 - Cultura

0.00% 0% ⌵

El **Segundo Nivel** de roca se agrega al presionar el botón de “+” en la barra gris oscuro. Sólo los Team Leaders tienen acceso a agregar este nivel de roca.

ROCA NIVEL 2

Roca 4 - Aprendizaje				0.00%	0%	⌵
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO	+	-	AVANCE
Plan de Estudio Tecnico	0.00%	0%	Desarrollo	+	-	Agregar Roca
Configuraciones Avanzadas Symphony	0.00%	0%	Desarrollo	+	-	
Android Development	0.00%	0%	Desarrollo	+	-	
Configuraciones de Kannel	0.00%	0%	Desarrollo	+	-	

El desglose de las rocas por los Team Members inicia a partir del nivel 3.

04- AGREGAR ROCAS EN NIVEL 3 Y 4

1 - Primer Nivel de roca. Esta la ingresa el Team Leader.

2 - Segundo Nivel de roca. Ingresado por Team Leader.

3 - Tercer Nivel de roca. Este ya es ingresado por el Team Member. Cuando se está ingresando, aparece un botón de 'Guardar', que luego es reemplazado por uno de 'Agregar' cuando ya la roca ya fue ingresada.

4 - Actualización de roca. Se ingresa el porcentaje de avance y se da click fuera del área de texto para que el sistema automáticamente actualice.

Roca 5 - Cultura					0.00%	0%	⬆
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO		AVANCE		
Foros	0.00%	0%	Diseño				
Analisis de Sitios Web	0.00%	0%	Diseño				
<input type="text"/>	0.00%	<input type="text" value="0.00 %"/>	Diseño				
Analisis de Aplicaciones Moviles	0.00%	0%	Diseño				
Lecturas Grupales	0.00%	0%	Diseño				

○ Al presionar el botón de "+", se crea una roca de Tercer Nivel bajo la roca 2 asignada por el Team Leader.

○ Al presionar el botón de "+" se crea una roca de Cuarto Nivel bajo la roca 3 ya creada por el Team Member. Estas se pueden eliminar con el botón de Basura. Eliminar una roca de nivel 3 elimina las que se hayan ingresado en el nivel 4.

04- EDITAR ROCAS EXISTENTES

Colocarse sobre el nombre de la roca a Editar y será marcada con subrayado.

Hacer click sobre el texto y se convertirá en un campo de texto para volver a escribir en él.

Roca 5 - Cultura					0.00%	0%	⬆
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO		AVANCE		
Foros	0.00%	0%	Diseño				
Analisis de Sitios Web	0.00%	0%	Diseño				
Pruebas	0.00%	<input type="text" value="0.00 %"/>	Diseño				

Roca 5 - Cultura					0.00%	0%	⬆
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO		AVANCE		
Foros	0.00%	0%	Diseño				
Analisis de Sitios Web	0.00%	0%	Diseño				
<input type="text"/>	0.00%	<input type="text" value="0.00 %"/>	Diseño				

D. Manual de miembro de equipo

Manual de Rocas - Team Member



01- LOG IN

Ingresa a página web <http://rocas.ebclosure.com/>

Para entrar ingrese su usuario y contraseña de correo electrónico de EBClosure.

02- PANTALLA DE INICIO

A - Vista gráfica del avance de tiempo en el Q.

B - Porcentajes esperados de avance, actual y anterior

C - Menú de navegación

D - Abre y cierra la pestaña para mostrar el contenido de cada una

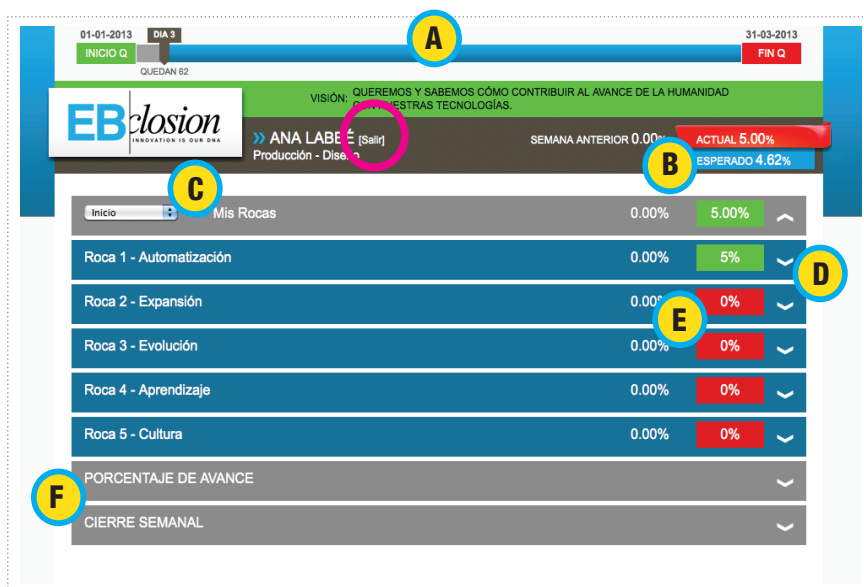
E - Vista rápida del porcentaje de cada roca con función de semáforo

F - Resúmenes de avances

» Porcentaje de Avance: muestra cuál es el porcentaje esperado (círculo naranja) y cuál es el del team member (en cuadrado y con función de semáforo)

» Cierre Semanal: muestra en cuánto debió cerrarse las rocas por semana y en cuánto lo hizo el team member.

 Link para Salir del Sistema de Rocas



03- AGREGAR ROCAS EN NIVEL 3 Y 4

1 - Primer Nivel de roca. Esta la ingresa el Team Leader.

2 - Segundo Nivel de roca. Ingresado por Team Leader.

3 - Tercer Nivel de roca. Este ya es ingresado por el Team Member. Cuando se está ingresando, aparece un botón de 'Guardar', que luego es reemplazado por uno de 'Agregar' cuando ya la roca ya fue ingresada.

4 - Actualización de roca. Se ingresa el porcentaje de avance y se da click fuera del área de texto para que el sistema automáticamente actualice.

Roca 5 - Cultura					0.00%	0%	⌵
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO		AVANCE		
Foros	0.00%	0%	Diseño				
Analisis de Sitios Web	0.00%	0%	Diseño				
<input type="text"/>	0.00%	<input type="text" value="0.00 %"/>	Diseño				
Analisis de Aplicaciones Moviles	0.00%	0%	Diseño				
Lecturas Grupales	0.00%	0%	Diseño				

○ Al presionar el botón de "+", se crea una roca de Tercer Nivel bajo la roca 2 asignada por el Team Leader.

○ Al presionar el botón de "+" se crea una roca de Cuarto Nivel bajo la roca 3 ya creada por el Team Member. Estas se pueden eliminar con el botón de Basura. Eliminar una roca de nivel 3 elimina las que se hayan ingresado en el nivel 4.

04- EDITAR ROCAS EXISTENTES

Colocarse sobre el nombre de la roca a Editar y será marcada con subrayado.

Hacer click sobre el texto y se convertirá en un campo de texto para volver a escribir en él.

Roca 5 - Cultura					0.00%	0%	⌵
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO		AVANCE		
Foros	0.00%	0%	Diseño				
Analisis de Sitios Web	0.00%	0%	Diseño				
<u>Pruebas</u>	0.00%	<input type="text" value="0.00 %"/>	Diseño				

Roca 5 - Cultura					0.00%	0%	⌵
TAREAS ASIGNADAS	% ANTERIOR	% ACTUAL	DEPTO		AVANCE		
Foros	0.00%	0%	Diseño				
Analisis de Sitios Web	0.00%	0%	Diseño				
<input type="text"/>	0.00%	<input type="text" value="0.00 %"/>	Diseño				