

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades

HPILOT

ESTUDIO PROGRAMADO. ENSEÑANZA-APRENDIZAJE

GUSTAVO ALFREDO SILVA RODRIGUEZ

Modelo de trabajo profesional presentado para optar
al grado académico de Licenciatura
en Ciencias de la Computación



HPILOT

ESTUDIO PROGRAMADO. ENSEÑANZA-APRENDIZAJE

Vo. Bo.

(f) Luis R. Furlán
Ingeniero Luis Furlán
Asesor

Tribunal:

(f) D. Córdón

(f) [Signature]

(f) Luis R. Furlán

Fecha de aprobación:

CONTENIDO

	Página
I. INTRODUCCION	1
II. MANUAL PARA EL USUARIO	3
A. Funciones de HPILOT	3
1. Ingresar o modificar una lección	4
2. Crear un archivo de control académico	6
3. Modificar un archivo de control académico	8
4. Listar un archivo de una lección o un archivo de control académico	17
5. Ejecutar una lección	20
B. Diseño de una lección HPILOT	22
C. Formato de las Instrucciones	24
1. Texto de una línea	25
2. [Grupo], nombre de la variable	26
3. Nombre de la etiqueta	27
4. Grupo de respuestas correctas	28
5. Tiempo de retardación de una lección	28
6. Fila, Columna	29
D. Instrucciones básicas del EDIT/1000	31
III. DOCUMENTACION TECNICA	37
IV. OBSERVACIONES ADICIONALES SOBRE HPILOT	73
V. SUGERENCIAS PARA MEJORAR HPILOT	79
VI. RESUMEN	81
VII. BIBLIOGRAFIA	83

LISTA DE FIGURAS

Figura.		Página
1.	Diagrama de pantalla para la opción 2	7
2.	Diagrama de pantalla para la opción 3	10
3.	Diagrama de pantalla para la opción 3 (Cont.)	11
4.	Diagrama de pantalla para la opción 3 (Cont.)	12
5.	Diagrama de pantalla para la opción 3 (Cont.)	14
6.	Diagrama de pantalla para la opción 3 (Cont.)	15
7.	Diagrama de pantalla para la opción 3 (Cont.)	16
8.	Diagrama de pantalla para la opción 4	18
9.	Listado del archivo de control	19
10.	Diagrama de pantalla para la opción 5	21
11.	Listado del archivo de control	51

I. INTRODUCCION

El presente trabajo se elaboró con la idea fundamental de proporcionar al maestro una herramienta que pueda ayudarlo en su labor educativa.

La herramienta que se proporciona es la computadora, utilizándola como un tutor del estudiante, es decir, en una forma interactiva de presentar material, hacer preguntas al estudiante y de aceptar respuestas del mismo.

El nombre del trabajo es HPILOT y así será como lo llamaremos de aquí en adelante. Este nombre es la combinación del nombre HEWLETT PACKARD y PILOT acrónimo para "Programmed Inquiry Learning or Teaching" un programa muy popular en Estados Unidos.

HPILOT es capaz de llevar un control del rendimiento académico del estudiante en las diferentes áreas de estudio en que el maestro crea conveniente la utilización de la computadora como un recurso didáctico. Es muy importante hacer notar que la evaluación que puede hacerse de las áreas de estudio es meramente objetiva.

HPILOT es un "lenguaje" de programación que utiliza instrucciones muy fáciles o sencillas. Es por esto que no se requiere de grandes conocimientos de computación para poder aplicar la computadora como un recurso didáctico.

II. MANUAL PARA EL USUARIO

A. Funciones del HPILOT

HPILOT es un lenguaje de programación diseñado para propósitos educativos. El objetivo de éste es que los maestros puedan utilizar la computadora como un recurso didáctico en el proceso enseñanza-aprendizaje.

Cuando el maestro (usuario) utiliza HPILOT puede realizar las siguientes funciones:

1. Ingresar o modificar una lección.
2. Crear un archivo de control académico.
3. Modificar un archivo de control académico.
4. Listar un archivo de una lección o listar un archivo de control académico.
5. Ejecutar una lección.
6. Regresar al sistema operativo del computador. (El maestro verá ":" (dos puntos), que le indican que puede salirse del computador o correr de nuevo HPILOT).

Cuando las opciones hayan sido presentadas al maestro, éste deberá seleccionar la que más le convenga.

¿Qué hace cada una de las opciones y cómo se usan?

1. Ingresar o modificar una lección.

Las lecciones escritas en HPILOT están almacenadas en el computador en forma de un archivo, que para nuestro propósito no es más que una colección de información referente a un tópico en particular que puede ser modificada posteriormente.

Cuando el maestro seleccione la opción 1, HPILOT le permitirá ingresar o modificar una lección. Si la lección no existe (se está ingresando por primera vez al computador), el maestro deberá ingresar al texto de la misma, el cual puede ser modificado al mismo tiempo. Si la lección ya existe puede ser modificada según le convenga al maestro.

El mensaje que el maestro verá cuando seleccione esta opción es el siguiente:

```
/ Edit      : Use ? for help  
/ FI, NAME specifies file to edit  
/ EOF
```

Seguidamente el maestro deberá ingresar el nombre del archivo donde está la lección, exista ésta o no. Es muy importante que el maestro recuerde el nombre del archivo de control académico ya que el nombre de la lección se forma usando las primeras cuatro letras y/o dígitos del nombre del archivo de control académico. De esta forma,

si el archivo de control académico se llama CONFIL los nombres de las cinco lecciones que este archivo puede manejar serían:

CONF01 ----- Para la lección 1.

CONF02 ----- Para la lección 2.

CONF03 ----- Para la lección 3.

CONF04 ----- Para la lección 4.

CONF05 ----- Para la lección 5.

Siendo 5 el número máximo de archivos que puede haber.

Si el maestro desea trabajar la lección 2 por ejemplo su instrucción al computador deberá ser la siguiente:

/FI, CONF02

luego deberá presionar la tecla RETURN .

Cuando ya haya terminado de ingresar o corregir la lección y desea que los cambios o la lección que ingresó sea almacenada en el computador la instrucción que deberá dar es la siguiente:

/ER

Si en caso no quisiera que los cambios o la lección sean guardados la instrucción será la siguiente:

/a ó /A

El Edit/1000 presentará el siguiente mensaje:

/OK?

El maestro deberá presionar la tecla "Y" y la tecla RETURN. Cuando se esté ingresando el texto de la lección

el maestro deberá tener cuidado de hacerlo en la siguiente forma:

Escribir el nombre de la instrucción en la columna 2 de la pantalla, es decir así:

/bT: Cuál es tu nombre?

Note que hay que dejar un espacio en blanco (representado por b) entre el nombre de la instrucción y la diagonal, que es el signo que nos indica que EDIT/1000 está dispuesto a aceptar una nueva línea de la lección. EDIT/1000 es un editor o un programa que permite crear nuevos archivos o modificar archivos ya existentes.

2. Crear un archivo de control académico

EL archivo de control académico contiene información pertinente a cada estudiante que incluye el código o clave asignado al estudiante y los resultados que éste haya obtenido en las diferentes lecciones. El código asignado a los estudiantes debe ser mayor o igual a 1 y menor o igual a 99.

Cuando se seleccione la opción 2 el maestro verá en la pantalla lo siguiente:

NOMBRE DEL ARCHIVO DE CONTROL....?

Lo que significa que debe escribir el nombre que le dará al archivo y luego presionar la tecla RETURN.

Ver Figura No. 1.

El archivo de control académico debe ser único para ca-

Nombre del archivo de control...?

Diagrama de pantalla para la opción 2

Figura No. 1

da área de estudio, es decir, puede existir un archivo de control para MATEMATICA, uno para GEOGRAFIA, uno para HISTORIA, etc.

Cada archivo de control puede ser dividido en cinco lecciones, que a su vez pueden ser divididas en 10 objetivos o grupos de preguntas según le convenga al maestro.

Los campos de los grupos de respuestas están inicialmente con un valor de -1 (menos uno) que indican que no ha habido o que no hubo respuestas evaluadas en ese o esos grupos de preguntas. Ver figura No. 9.

EL grupo de respuestas es asignado o determinado por el maestro en el programa escrito en HPILOT, que es la lección.

Por ejemplo: Si estuviéramos en un programa escrito en HPILOT, una instrucción como la siguiente:

```
/ A:0,$ANSWER
```

Al ver el listado, el grupo que le corresponde a esta pregunta es el grupo 0. (0 indica en el programa principal el grupo al cual pertenece esta pregunta).

3. Modificar un archivo de control académico.

Esta opción se selecciona cuando el maestro desea modi-

ficar un archivo de control académico, ya sea para agregar un alumno, o para eliminar uno del archivo o para modificar el archivo de control académico para permitirle al estudiante repetir una lección.

Cuando se seleccione esta opción el maestro verá en su pantalla lo descrito en la figura No. 2

Luego verá el siguiente mensaje:

NOMBRE DEL ARCHIVO DE CONTROL....?

El maestro escribirá el nombre del archivo y presionará la tecla RETURN. Ver figura No. 3.

Si se quiere ingresar nuevos alumnos al archivo se verá el siguiente mensaje en la pantalla:

CLAVE DEL ESTUDIANTE (1 <= Clave <= 99)....?

NOMBRE DEL ESTUDIANTE (30 letras máximo)...?

Aquí el maestro deberá asignarle una clave al estudiante, la cual no deberá ser menor de 1 ni mayor de 99 y debe ser única para cada estudiante, presionará la tecla RETURN y luego ingresará el nombre del estudiante, el cual no deberá exceder de 30 caracteres y luego presionará la tecla RETURN. Ver figura No. 4

Si se tratara de ingresar una clave repetida por error o intencionalmente, el maestro verá en la pantalla un mensa-

Ingresar un nuevo estudiante	1
Suprimir un estudiante	2
Permitirle al estudiante	
repetir una lección	3
Terminar	0

Diagrama de pantalla para la opción 3

Figura No. 2

Nombre del archivo de control...?

Diagrama de pantalla para la opción 3 (cont.)

Figura No.3

Clave del estudiante (1 <= clave <= 99)....?

Nombre del estudiante (30 letras max)...?

Diagrama de pantalla para la opción 3 (cont.)

Figura No. 4

je indicándole que esta clave ya existe. Ver figura No. 5.

Si lo que desea es eliminar un estudiante del archivo deberá hacer lo siguiente:

Seleccionar la opción 2. Entonces verá el mensaje:

CLAVE DEL ESTUDIANTE (1 <= Clave <= 99)....?

El maestro ingresará la clave del estudiante que desea eliminar del archivo y presionar la tecla RETURN. Además puede verificar si es ése el estudiante que desea eliminar. Ver figura No. 6.

Si desea permitirle a un estudiante repetir una lección deberá seleccionar la opción 3. Esta opción es necesaria, ya que cuando un alumno ha tomado una lección no puede tomarla nuevamente, sino hasta que el maestro se lo permita. Cuando la haya seleccionado verá en la pantalla:

CLAVE DEL ESTUDIANTE (1 <= clave <= 99)....?

NUMERO DE LA LECCION.....?

El maestro ingresará la clave, presionará la tecla RETURN y seguidamente ingresará el número de la lección que desea que el estudiante repita. Si no hace esto, el estudiante no podrá repetir ninguna lección. Ver figura No. 7.

La clave 2 ya existe

Presione RETURN para continuar

Diagrama de pantalla para la opción 3 (cont.)

Figura No. 5

Clave del estudiante (1 <= clave <= 99)....?

Es este el estudiante- XXXXX XXXXX

Ingrese SI o NO

El estudiante XXXXX XXXXX No fue eliminado

Diagrama de pantalla para la opción 3 (cont.)

Figura No. 6

Clave del estudiante (1 <= clave <= 99)....?

Número de la lección.....?

Diagrama de pantalla para la opción 3 (cont.)

Figura No. 7

4. Listar una lección o un archivo de control académico

Esta opción permite al maestro listar en la impresora o en la pantalla una lección o un archivo de control académico.

Cuando se seleccione esta opción verá el mensaje descrito en la figura No. 8.

El maestro ingresará el nombre del archivo y presionara la tecla RETURN.

Se hace la selección (ingresando la palabra PANTALLA o la palabra IMPRESORA) y se presiona RETURN. Ver figura No. 8

El formato del listado de la lección es el mismo que tiene cuando el maestro ingresa la lección o lo modifica. El formato del listado del archivo de control académico puede verse en la figura No. 9

El maestro deberá interpretar el listado del archivo de control académico de la siguiente manera:

La información que aparece bajo la columna "NOMBRE/CODIGO" se refiere al nombre del estudiante y a la clave o código que el maestro le haya asignado a cada estudiante para usar HPILOT.

La información que aparece bajo la columna "LEC" se refiere al número de la lección en que se ha evaluado al estudiante, o a las lecciones que el estudiante puede to-

Unidad donde quiere el listado

Pantalla o

Impresora ?

Diagrama de pantalla para la opción 4

Figura No. 8

NOMBRE/CODIGO	LEC	PUNTAJE X GRUPO												
		0	1	2	3	4	5	6	7	8	9			
FERNANDO MENDOZA 25	1	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
	2	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
	3	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
	4	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
	5	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
	JORGE ROLDAN 40	1	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
			M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		2	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
			M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		3	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
M			-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
4		B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
5		B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	

LISTADO DEL ARCHIVO DE CONTROL

Figura No. 9

mar.

La información que aparece bajo la columna "PUNTAJE X GRUPO" se refiere a los diez grupos de respuestas en que puede ser dividida cada lección. Los punteos en la fila identificada con una "B" indican las respuestas correctas del estudiante, los puntos en la fila identificada con una "M" indican las respuestas incorrectas del estudiante.

5. Ejecutar una lección.

-Esta opción permite al maestro ejecutar una lección con el propósito de que el maestro esté seguro de cómo es ejecutada la lección, es decir, que la pruebe antes de darla al estudiante.

Cuando se seleccione esta opción, el maestro verá en la pantalla el mensaje:

NUMERO DE LA LECCION....?

Ver figura No. 10.

El rango de los números que identifican a las lecciones es de 1 a 5, cualquier otro número no es permitido.

Cuando sea un estudiante el usuario de HPILOT únicamente podrá ejecutar una lección, es decir, no tiene acceso a las otras opciones de HPILOT. Este control se logra por medio de la clave que se le ha asignado al estudiante, es

Número de la lección....?

Diagrama de pantalla para la opción 5

Figura No. 10

por eso que la clave no debe exceder del número 99.

Lo que el estudiante debe o necesita saber para ejecutar una lección es: el nombre del archivo de control académico, su código o clave y el número de la lección. Esta información es proporcionada por el maestro.

Cuando el estudiante termina con la lección, HPILOT actualiza el archivo de control académico con los punteos obtenidos en los diferentes objetivos en que el maestro haya dividido la lección.

Si el usuario es un maestro , deberá correr HPILOT así:

```
:PILOT,TO,LI,PH
```

Esta es la clave de seguridad para que los estudiantes no tengan acceso a ninguna de las opciones excepto la de ejecutar una lección. Si el maestro no lo hace de esta forma no tendrá acceso a todas las opciones de HPILOT.

B. Diseño de una lección en HPILOT.

Para el diseño de una lección se utilizan las siguientes instrucciones:

T: Permite al maestro imprimir el texto de una línea en la pantalla.

A: Permite al maestro preguntar al alumno.

I: Permite al maestro hacer preguntas al estudiante cu-

ya respuesta consta de una sola letra

M: Compara la respuesta correcta o posibles respuestas correctas con la que el estudiante da, además evalúa y lleva un control de las preguntas respondidas correcta e incorrectamente.

J: Permite transferir el flujo del programa hacia otras instrucciones dentro del mismo programa. Es decir, permite "saltar" hacia otras instrucciones.

U: Funciona básicamente como la anterior, pero con la diferencia de que el control se transfiere hacia una subrutina. (Esto es un grupo de instrucciones que puede ser ejecutadas varias veces en el programa únicamente llamando a la subrutina. No es necesario escribirlas cada vez).

E: Marca el fin de una subrutina. Cuando se ejecuta esta instrucción, la siguiente que se ejecutará es la que le sigue a la instrucción de donde se llamó a la subrutina.

R: Se utiliza para poner comentarios en la lección, los cuales pueden ser de gran ayuda para el maestro. El texto que acompaña a esta instrucción no es visto por el estudiante al momento de ejecutar la lección.

C: Sirve para borrar o limpiar la pantalla y poner el CURSOR en la fila 1 y columna 1.

P: Permite al maestro colocar el CURSOR en la fila y columna que éste desee.

/E: Marca el final de un programa escrito en HPILOT. Debe ser la última instrucción y no debe repetirse dentro del programa.

*: Identifica una etiqueta hacia donde se puede transferir el control del programa utilizando las instrucciones J o U.

D: Permite detener la ejecución del programa por un tiempo estipulado por el maestro.

C. FORMATO DE LAS INSTRUCCIONES.

Existen en HPILOT básicamente 3 formatos de instrucciones:

Formato 1: Nombre de la instrucción. : [Opciones]

Formato 2: Nombre de la instrucción. [condición][:]
[Opciones.]

Formato 3: * Nombre de una etiqueta.

El formato 1 es un tipo de instrucción sin condición lo que significa que esta instrucción se ejecutará siempre que el programa o la lógica del programa llegue a ella.

El formato 2 es un tipo de instrucción con condición lo

que significa que esta instrucción se ejecutará dependiendo del resultado que se haya obtenido al evaluar una respuesta del estudiante. La evaluación de la respuesta se realiza con la instrucción "M"

Para cada instrucción se necesita un complemento que depende de la instrucción misma. El uso de estos complementos se explicará más adelante.

El formato 3 se refiere a la forma de identificar las etiquetas en un programa en HPILOT. El * (asterisco) es siempre obligatorio y el nombre de la etiqueta también.

Nombre de la instrucción:

El nombre de la instrucción es siempre obligatorio y debe ir seguido de ":"

[Condición]:

La condición está representada por las letras "Y" (para sí) o "N" (para no) y significa que la ejecución de la instrucción está condicionada por la instrucción "M" anterior. La condición es opcional y sólo se usa si hubo una "M" inmediatamente precediéndola.

: Son siempre obligatorios.

Opciones de las instrucciones

1. Texto de una línea.

Solamente se usa con la instrucción "T".

Por texto de una línea debemos entender una línea de la

lección. Este texto puede también ser una línea en blanco en caso quisiéramos dejar líneas en blanco entre dos líneas de la lección.

Cualquier texto que se utilice como complemento de la instrucción "T" será impreso en la pantalla del maestro.

Ejemplo 1:

/ T: Cual es tu nombre?

/ TN: Lo siento la respuesta no es correcta.///

/ TY: Bravo, \$NOMBRE. Correcto.///

2. [Grupo], Nombre de la variable

El grupo y el nombre de la variable se utiliza siempre con la instrucción "A:".

Grupo identifica al área o material de estudio al cual pertenece la pregunta que se le hace al estudiante. Puede tener valores del 0 al 9 lo que nos da 10 grupos de respuestas para cada lección; el maestro puede dividir el material de estudio (la lección) en diez objetivos y evaluarlos independientemente unos de los otros.

Nombre de la variable identifica a la variable que contiene la respuesta del estudiante a una pregunta del maestro. Siempre va precedida del signo "\$" y puede tener de 1 a 10 letras y/o dígitos. Puede empezar con un dígito.

Ejemplo 2:

/ T: Cuál es tu nombre?

/ A:\$NOMBRE

/ T: Mucho gusto \$NOMBRE.

Ejemplo 3:

/ T: Cuál es la capital de Bulgaria?

/ A:1,\$RESPUESTA

/ M:SOFIA

/ TN: No, \$RESPUESTA no es la capital de Bulgaria.

/ T: la respuesta correcta es SOFIA.

El número 1 entre la coma y los dos puntos representa al grupo al cual pertenece la respuesta.

3. Nombre de la etiqueta

Se utiliza con la instrucción J o con la instrucción U, e indica que queremos saltar hacia otra instrucción en el programa.

El orden de ejecución de las instrucciones en un programa es normalmente secuencial. Cuando nosotros utilizamos las instrucciones J o U alteramos ese orden.

Ejemplo 4:

/ T:

/ *INICIO

/ T: Ingrese una vocal

/ A:1,\$IRES

/ M:A,E,I,O,U

/ JY: FINAL

/ JN: INICIO

4. Grupo de respuestas correctas

Solamente se usa con la instrucción "M".

Es una serie de caracteres alfanuméricos que representan la o las respuestas correctas. La respuesta puede formarse de cualquier combinación de caracteres permitidos como: las letras del alfabeto, los dígitos del 0 a 9, los operadores aritméticos y relacionales y ciertos signos especiales.

Ejemplo 5:

/ *INICIO

/ T: Ingrese vocal

/ A:5,\$RESPUESTA

/ M:A,E,I,O,U

/ JY:CORRECTO

/ U:INCORRECTO

/ J:INICIO

/ * INCORRECTO

/ T:

/ T: Lo siento, \$RESPUESTA no es una vocal

/ T:

/ E:

5. Tiempo de retardación de la lección:

Se utiliza sólo con la instrucción "D".

Consta de 4 dígitos del 0 al 9999 y representa al tiem-

po en segundos que se desea retardar o detener la ejecución de la lección.

Esta instrucción puede ser muy útil cuando el maestro desea esperar cierto tiempo para que el estudiante lea un segmento de la lección que está escrito en la pantalla.

Ejemplo 6

/.

.

.

/ D:2

6. Fila, columna

Se utiliza con la instrucción "P".

Consta de 2 pares de dígitos del 00 al 99 y representan la fila y columna de la pantalla donde se desea poner el CURSOR. (EL CURSOR es el caracter que aparece en la pantalla y que indica que la computadora está lista para recibir información del usuario). El máximo número para las filas es de 24 y para la columnas es de 80.

Ejemplo 7:

/ T:

/ T:

.

.

/ P:10,42

La instrucción "P" indica que se quiere poner el CUR-SOR en la fila 10 y columna 42.

Ejemplo 8:

Veamos un ejemplo más de HPILOT:

/ T: Ahora desarrolla este binomio: $(a+b)^3$

/ A:3,\$IANS

(1) / M: $a^3 + 3a^2 + ab^2 + b^3$

/ T:

/ TY: Correcto, \$NOMBRE

/ TN: Lo siento, \$IANS NO es la respuesta correcta.

En este caso lo que el estudiante verá en su pantalla depende de si la respuesta que él dio es correcta o no.

En caso fuera correcta verá:

Correcto, Pepe

Si no es correcta verá:

Lo siento, $\frac{a^3 + 3ab + 3ab^2 + b^3}{(2)}$ No es la respuesta correcta'

(1) es la respuesta correcta.

(2) es la respuesta del estudiante.

Algo que debe tomarse en cuenta cuando se diseña una lección en HPILOT es la forma en que se evalúan las respuestas, los caracteres que comprenden la respuesta correcta o las respuestas correctas deben ser exactamente las mismas que del estudiante, porque de otra forma la respuesta

será calificada como incorrecta.

Por ejemplo si la respuesta del estudiante fuera una "a" (minúscula) y la respuesta guardada en la lección fuera una "A" (mayúscula) el resultado de la evaluación será incorrecto. Esto se debe a que "a" y "A" son dos caracteres diferentes. Es por eso que se debe ser cuidadoso cuando se utilice HPILOT para evitarle confusiones al estudiante. Cuando se quieran aceptar ambas formas de una letra debe indicarlo en la instrucción "M".

D. Instrucciones básicas del EDIT/1000

A continuación se explicarán las instrucciones más básicas que el maestro o usuario debe saber para poder usar el EDIT/1000. Si el maestro necesitara una mayor explicación acerca de las instrucciones que aquí se explican puede referirse al manual del EDIT/1000. En este manual encontrará más instrucciones que le pueden servir de ayuda y que mejorarán su eficiencia en el manejo del EDIT/1000.

1. Ingresar una instrucción del programa. (Una línea de la lección).

Simplemente presione una vez la barra espaciadora del teclado y comience a ingresar la instrucción.

Ejemplo: /bT:HOLA

La "/" indica que EDIT/1000 espera que usted ingrese una instrucción o una línea de la lección.

La "b" es el espacio en blanco que debe dejar.

"T: HOLA" es el texto de la línea de la lección.

2. Listar el texto de una lección.

EDIT/1000 le permite listar en la pantalla el texto de la lección que usted está trabajando actualmente. Si desea listar toda la lección en la pantalla su instrucción deberá ser la siguiente:

Ejemplo: /L1\$

La "/" indica que se espera de usted una instrucción para EDIT/1000.

La "L" es una instrucción del EDIT/1000 que indica que usted quiere listar un segmento de la lección.

El número "1" y el signo "\$" indican que quiere listar toda la lección.

Esta instrucción puede escribirse de la siguiente forma:

/L1\$

Si desea listar un número determinado de líneas puede hacerlo así:

Ejemplo: /L10

Esta instrucción le indica a EDIT/1000 que usted desea listar 10 líneas a partir de la línea dónde usted está actualmente.

Si desea saber cuál es el número de las líneas de su lección puede tratar esta instrucción:

Ejemplo: /LN25

Con esta instrucción usted podrá ver cuáles son los números de línea de cada instrucción de su programa o lección.

Para eliminar el efecto de la instrucción anterior usted puede darle a EDIT/1000 la siguiente instrucción:

Ejemplo: /LU

Cuando usted presione la tecla RETURN, verá en su pantalla 20 líneas de su lección pero no estarán numeradas.

3. Modificar una línea de la lección.

Para poder modificar el texto de una línea usted deberá darle a EDIT/1000 la siguiente instrucción:

Ejemplo: /P

Supongamos que usted tiene en su lección esta instrucción:

/ T: Cuál es tu nombre?

Si desea cambiar nombre por nombre deberá hacerlo así:

/ T: Cuál es tu nombre? (línea en su lección)

/P//////////n (Instrucción para EDIT/1000)

Al presionar RETURN usted verá en la pantalla:

/ T: Cuál es tu nombre (Se cambió nombre por nombre)

Con la instrucción "P" y la "/", usted puede hacer varios cambios en una línea de su programa (lección).

Supongamos que usted tiene en su lección la siguiente línea:

/ T: CUAL ES LA CAPATAK DE GOSTEMALS

Usted puede corregirla así:

/ T: CUAL ES LA CAPATAK DE GOSTEMALS (línea en su lección)

/P//////////I//L////UA/////A (Inst. para EDIT/1000)

Luego usted verá en su pantalla la línea de su lección escrita correctamente así:

/ T: CUAL ES LA CAPITAL DE GUATEMALA

4. Eliminar una o varias líneas de la lección.

Para eliminar una o varias líneas de la lección o programa usted puede darle a EDIT/1000 la siguiente instrucción:

Ejemplo: /K ó /K

Supongamos que desea eliminar la instrucción de su programa:

/ A:1,\$IANS

Puede hacerlo así:

/ A:1,\$IANS (línea de su programa)

/K (Instrucción para EDIT/1000).

Con esta instrucción queda eliminada esa línea de su lección.

Veamos un ejemplo más de la instrucción "K"

Supongamos que hasta ahora usted tiene las siguientes instrucciones en su programa (lección):

```

/ T: CUAL ES TU NOMBRE?           (1)
/ T:                               (2)
/ T:                               (3)
/ A:$NOMBRE                       (4)
/ T: HOLA, $NOMBRE                (5)
/ T: $NOMBRE,VAMOS A ESTUDIAR UN RATO (6)

```

Supongamos también que desea eliminar la línea 3 del programa, entonces deberá darle a EDIT/1000 las siguientes instrucciones:

```

/3           (Para ponerse en línea 3)
/ T:         (Línea de su programa (lección))
/K

```

Luego verá en su pantalla la siguiente instrucción de su programa, que en este caso es:

```

/ A:$NOMBRE

```

Cuando esté usando la instrucción "K" "k" debe ser muy cuidadoso y estar seguro de si realmente desea eliminar

esa línea. Esto es necesario para no obtener resultados inesperados.

Puede referirse al manual del EDIT/1000 para mayor información sobre esta instrucción.

III. DOCUMENTACIÓN TÉCNICA

Nombre del programa: PILOT

Descripción:

Este programa presenta las diferentes opciones que representan los mandatos que HPILOT posee para:

1. Editar y crear un archivo donde se escribirá una lección.
2. Crear un archivo de control académico.
3. Listar un archivo de control académico o un archivo que contiene la lección.
4. Actualizar un archivo de control académico.
5. Ejecutar una lección.
6. Salir de HPILOT y regresar al sistema operativo.

Cada mandato se ejecuta seleccionando el dígito que acompaña a cada selección.

Posee dos módulos principales uno para el maestro y uno para el estudiante. En el módulo del maestro se presentan todas las opciones mientras que en el módulo para el

estudiante sólo se puede ejecutar una lección (opción 5).

El programa maneja dos archivos principales:

1. El archivo de control académico.

Este archivo contiene información acerca de los estudiantes: el código que le fue asignado para poder usar HPILOT y que lo identifica únicamente a él, su nombre y los puntajes que haya obtenido en las diferentes lecciones que el maestro haya diseñado para su estudio. El número máximo de lecciones que puede controlar es cinco. Cada lección está dividida en diez grupos diferentes que representan diferentes objetivos en que el maestro ha dividido la lección. Estos objetivos pueden o no ser utilizadas en su totalidad, es decir, el maestro puede usar los que crea conveniente hasta un máximo de diez.

2. El archivo que contiene la lección.

Este archivo es único para cada lección, Por lección debemos entender las instrucciones que HPILOT posee y que se hayan usado y el texto propio de la lección que es proporcionado por el maestro.

Descripción de los campos de cada archivo:

1. Registro del archivo de control académico:

<u>Posición en Palabras</u>	<u>Contenido</u>
1	Clave o código del estudiante.

- 2 - 16 Nombre del estudiante.
- 17 Campo indicador de que el estudiante ya tomó la lección 1 y si la puede tomar de nuevo (puede ser modificado por el maestro).
- 18 - 27 Contienen el punteo del estudiante en cada grupo de respuestas en la lección 1. Únicamente las correctas.
- 28 - 37 Contienen el punteo del estudiante en cada grupo de respuestas en la lección 1. Únicamente las incorrectas.
- 38 Igual que el campo 17 para la lección 2.
- 39 - 48 Igual que los campos 18-27 para la lección 2.
- 49 - 58 Igual que los campos 28-37 para la lección 2.
- 59 Igual que el campo 17 para la lección 3.
- 60 - 69 Igual que los campos 18-27 para la lección 3.
- 70 - 79 Igual que los campos 28-37 para

la lección 3.

80 Igual que el campo 17 para la
lección 4.

81 - 90 Igual que los campos 18-27 para
la lección 4.

81 - 90 Igual que los campos 18-27 para
la lección 4.

91 - 100 Igual que los campos 28-37 para
la lección 4.

101 Igual que el campo 17 para la
lección 5.

102 - 111 Igual que los campos 18-27 para
la lección 5.

112 - 121 Igual que los campos 28-37 para
la lección 5.

Nombre del Archivo	CONPIL		Descripción del Archivo	Garretín	Fuente del Archivo	-----	Nombre del registro	IBUF
No. Bases por Registro	No. Registros por Base	Revisión No.	Procesos que utilizan este archivo	HPILLOT	Preparado por	GUSTAVO SILVA	Fecha	12/6/84
242	7	---						

C L A V E	NOMBRE DEL ESTUDIANTE																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

BUENAS EN LA LECCION 1.																				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

BUENAS EN LA LECCION 2.																				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

BUENAS EN LA LECCION 3.																				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

BUENAS EN LA LECCION 4.																				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

DESCRIPCION DEL REGISTRO DEL ARCHIVO DE CONTROL ACADEMICO USADO POR HPILLOT

2. Registro del Archivo de una lección.

<u>Posición en Palabras</u>	<u>Contenido</u>
1	Una instrucción de HPILOT.
2	Una variable lógica de HPILOT (OPCIONAL).
3	Dos puntos ":" .
4 - 71	Texto de la lección.

La instrucción y los dos puntos son siempre obligatorios, mientras que el texto depende de la instrucción que se esté usando. En la posición 72 va siempre un asterisco (cuando se trata del archivo de acceso directo) mientras que en el de acceso secuencial (el que usa el maestro desde el editor) no se utiliza.

Este diseño de registro se usa en ambos archivos. HPILOT utiliza realmente un archivo de acceso directo para el diseño de las instrucciones de bifurcación es por eso que es necesario hacer una copia temporal del archivo de acceso secuencial en uno de acceso directo.

Flujo.

1. Identifica el tipo de usuario, que puede ser un maestro o un estudiante. Lo reconoce mediante la clave de acceso a HPILOT.
2. Si el usuario es estudiante:

2.1 Pregunta el número de la lección que va a estudiar (1 a 5).

2.2 Llama a la subrutina MAIN, verifica si puede tomar o no la lección y la ejecuta.

2.3 Regresa al programa principal.

2.4 Actualiza el archivo de control académico con los puntajes obtenidos por el estudiante en la lección. Los puntajes representan las preguntas respondidas correcta e incorrectamente.

2.5 Finaliza.

3. Si el usuario es un maestro:

3.1 Presenta todas las opciones que HPILOT posee.

3.2 El usuario selecciona la opción que desea.

3.3 Llama a MAIN y ejecuta la selección y regresa al programa principal.

3.4 Presenta de nuevo todas las opciones.

3.5 Si el usuario selecciona la opción 6 termina.

3.6 Va a 3.2.

Módulos Principales del programa.

Módulo para el estudiante:

1. Este módulo pregunta la clave al estudiante y la lección que le corresponde estudiar y realiza la ejecución de la misma.

2. Cuando el estudiante haya terminado con la lección actualiza el archivo de control académico y termina.

Módulo para el maestro:

1. Presenta las opciones ya descritas.
2. El usuario selecciona una.
3. Llama a la subrutina MAIN para ejecutar la selección.
4. Regresa al programa principal.
5. Termina.

Nombre de la subrutina: MAIN

Esta subrutina ejecuta las opciones presentadas en el programa principal.

Módulos de la subrutina

Para la ejecución de las opciones la subrutina se dividió en varios módulos:

1. Módulo para la edición y creación del archivo que contiene la lección.
2. Módulo para crear un archivo de control académico.
3. Módulo para actualizar un archivo de control académico.
4. Módulo para listar un archivo de control académico o un archivo que contiene la lección.
5. Módulo para ejecución de una lección.

1. Módulo para edición y creación:

Este módulo se ejecuta cuando el maestro selecciona la opción 1.

Simplemente ejecuta el programa EDIT/1000 que HP proporciona a sus usuarios cuando se quiere modificar o ingresar la lección. Cuando se desea crear una lección se pregunta el nombre del archivo de control académico y en base a este nombre se crea el nombre del archivo que contiene la lección, es decir, se utilizan las primeras cuatro letras del nombre del archivo de control y las últimas dos letras corresponden a los dígitos que identifican a una lección. Estos dígitos se representan como 01 al 05 en ASCII, sin embargo el maestro puede ingresar solamente un dígito que indica cuál es el número de la lección que se desea crear.

2. Módulo para crear un archivo de control académico:

- 2.1 Pregunta el nombre que se le quiere dar al archivo
- 2.2 Crea el archivo con un código de seguridad para escritura únicamente. El código de seguridad es 124.
- 2.3 Pone los campos alfabéticos en blanco y los campos numéricos en -1.
- 2.4 Regresa el programa principal.

Cuando el maestro selecciona esta opción verá en la pantalla lo siguiente:

NOMBRE DEL ARCHIVO DE CONTROL.....?

Lo que indica que debe ingresar el nombre del archivo.

3. Módulo para actualización del archivo de control académico:

3.1 Presenta una lista con las opciones:

- Ingresar un nuevo estudiante 1
- Suprimir un estudiante 2
- Permitirle al estudiante
repetir una lección 3
- Terminar 0

3.2 Pregunta el nombre del archivo que se quiere modificar.

3.3 Abre el archivo para modificación.

3.4 Si se quiere ingresar un nuevo alumno:

3.4.1 Pregunta la clave y el nombre.

3.4.2 Lo graba en el archivo.

3.4.3 Regresa al inciso 1.

El maestro verá en la pantalla el siguiente mensaje:

CLAVE DEL ESTUDIANTE (1 <= clave <= 99)....?

NOMBRE DEL ESTUDIANTE (30 letras máximo)....?

3.5 Si se quiere suprimir un estudiante:

3.5.1 Pregunta la clave del estudiante.

3.5.2 Limpia el registro.

3.5.3 Regresa al inciso 1.

El maestro verá en la pantalla el mensaje:

CLAVE DEL ESTUDIANTE (1 <= clave <= 99)....?

3.6 Si se quiere permitirle al estudiante repetir una lección:

3.6.1 Pregunta la clave del estudiante y el número de la lección.

3.6.2 Regresa al inciso 1.

El maestro verá en la pantalla el mensaje siguiente:

CLAVE DEL ESTUDIANTE (1 <= clave <= 99)....?

NUMERO DE LA LECCIÓN.....?

3.7 Si escoge "0" regresa el programa principal.

4. Módulo para listar:

4.1 Pregunta el nombre del archivo que quiere listar.

NOMBRE DEL ARCHIVO....?

4.2 Pregunta la unidad donde quiere el listado.

UNIDAD DONDE QUIERE EL LISTADO

PANTALLA O

IMPRESORA

4.3 Pregunta el tipo de archivo (si es una lección o un archivo de control académico).

LISTAR UN ARCHIVO DE CONTROL INGRESE 1

LISTAR UNA LECCION INGRESE 2

>

El formato del listado del archivo de control aca-

defínico puede verse en la figura 11.

Los campos que no han sido evaluados siempre tendrán un -1.

5. Módulo para ejecución:

En este módulo se ejecutan las instrucciones de HPILOT utilizando la subrutina PRFRM.

- 5.1 . Pregunta el número de la lección que se quiere estudiar.
- 5.2 Abre el archivo que contiene la lección.
- 5.3 Crea un archivo temporal que es idéntico al que contiene la lección. La razón para hacer esto es que el archivo original es tipo 4 (de acceso secuencial) y HPILOT necesita un archivo tipo 2 (de acceso directo) para la implementación de las instrucciones "U" y "J".
- 5.4 Copia el archivo original en el archivo temporal.
- 5.5 Lee todo el archivo temporal para encontrar todas las etiquetas hacia donde se puede transferir el control del programa.
 - 5.5.1 Guarda la etiqueta en la tabla correspondiente usando la subrutina HASHG.
- 5.6 Hace una verificación de la sintaxis de las instrucciones que el maestro ha usado.

NOMBRE/CODIGO	LEC	PUNTAJE X GRUPO										
		0	1	2	3	4	5	6	7	8	9	
RICARDO ARAGON 90	1	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	2	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	3	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	4	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	5	B	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
		M	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

LISTADO DEL ARCHIVO DE CONTROL

Figura No. 11

- 5.7 Si encuentra un error no ejecuta la lección.
Un error significa una instrucción no reconocida.
- 5.8 Si no hay errores ejecuta la lección.
- 5.9 Obtiene del archivo la instrucción que se quiere ejecutar
- 5.10 Llama a la subrutina PRFRM y ejecuta la instrucción.
- 5.11 Regresa a la subrutina MAIN.
- 5.12 Va a 5.9
- 5.13. Si terminó destruye el archivo temporal y regresa al programa principal.

Cuando se seleccione esta opción, el maestro verá en la pantalla el mensaje:

NUMERO DE LA LECCION....?

Nombre de la subrutina: PRFRM

Descripción:

Esta subrutina interpreta las instrucciones de un programa escrito en HPILOT. La subrutina no reconoce sino un código numérico equivalente a la letra que representa la instrucción.

Las instrucciones que puede ejecutar esta subrutina son las siguientes:

- 1. T: Despliega el texto de una línea de la lección que el alumno está estudiando actualmente.

2. I: Acepta como entrada al programa simplemente un caracter.
3. A: Acepta como entrada al programa una cadena alfanumérica que representa una respuesta del estudiante.
4. M: Prepara la expresión lógica para una instrucción del tipo IF THEN. Cuando se encuentra en el programa se compara la respuesta del estudiante con la que está en el archivo que contiene la lección y que es la correcta.
5. J: Representa la instrucción GOTO. Esta instrucción va acompañada de una etiqueta que indica hacia donde bifurcará condicional o incondicionalmente el flujo del programa.
6. U: Representa la instrucción GOSUB. Se acompaña siempre de una etiqueta que representa el nombre de una subrutina que también puede ser ejecutada condicional o incondicionalmente.
7. E: Representa la instrucción RETURN e indica el final de una subrutina.
8. C: Limpia la pantalla y posiciona el CURSOR en la primera fila y primera columna de la misma.
9. D: Detiene la ejecución del programa de un tiempo determinado por el programador (maestro).
10. P: Posiciona el CURSOR en la fila y columna de la pantalla determinados por el programador.

Estructura de los datos utilizados:

La estructura es la misma que la descrita para los datos utilizados en el programa principal PILOT.

Flujo

Para la implementación de esta subrutina se utilizó la estructura lógica similar a la de Pascal "CASE", que en RATFOR se llama WHEN.

1. Seleccione el código que representa la instrucción que se desea ejecutar.

Cuando el código es 1: Ejecute la instrucción T.

Cuando el código es 2: Ejecute la instrucción I.

Cuando el código es 3: Ejecute la instrucción A.

Cuando el código es 4: Ejecute la instrucción M.

Cuando el código es 5: Ejecute la instrucción J.

Cuando el código es 6: Ejecute la instrucción U.

Cuando el código es 7: Ejecute la instrucción E.

Cuando el código es 9: Ejecute la instrucción C.

Cuando el código es 11: Ejecute la instrucción D.

Cuando el código es 13: Ejecute la instrucción P.

Después de que ha ejecutado cada una de las instrucciones debe regresar al programa principal.

Los códigos 8, 10 y 12 han sido reservados por si en un futuro se quieren agregar nuevas instrucciones.

Módulos de la subrutina:

Esta subrutina posee un módulo para cada instrucción

que se desea ejecutar en el programa. Así la estructura es la siguiente:

1. Módulo para la instrucción T:
Llama a la subrutina GETO para actualizar la variable de edición "IEDIT", luego imprime o despliega "IEDIT" en la pantalla.
2. Módulo para la instrucción I:
Simplemente se ejecuta la instrucción READ (unidad de lectura, formato), y lee desde la pantalla un caracter (letra).
3. Módulo para la instrucción A:
 - 3.1 Determina el grupo al cual pertenece la respuesta que el estudiante debe dar.
 - 3.2 Lee la respuesta desde la pantalla en la variable IRESP.
 - 3.3 Llama a la subrutina HASHG para guardar en la tabla de variables el nombre de la variable y el texto de la respuesta. El texto se guarda para poder desplegarlo, en caso necesario, posteriormente, si el maestro así lo quiere
 - 3.4 Si la respuesta es mayor de 10 caracteres guarda el resto en la tabla auxiliar de variables VTABLE
 - 3.5 Regresa al programa principal.
4. Módulo para la instrucción M:
 - 4.1 Verifica si la respuesta es del tipo de las

de un solo caracter (Y o N) o (S o N).

4.1.1 Si es de ese tipo las compara con las que el maestro espera y actualiza la variable para la implementación de la instrucción IFTHEN.

4.2 Si se trata de una respuesta cuya forma es la de una cadena alfanumérica de 1 a 20 caracteres (Letras y/o dígitos):

4.2.1 calcula la longitud de la respuesta del estudiante.

4.2.2 Compara la respuesta del estudiante con la correcta o correctas.

4.2.3 Si la respuesta es correcta actualiza la variable de la instrucción IFTHEN con un valor que indica que la respuesta es correcta. Este valor puede ser utilizado por la instrucción siguiente.

4.2.4 Si la respuesta es incorrecta se actualiza la variable de la instrucción IFTHEN con un valor que indica que la respuesta es incorrecta. Este valor puede ser utilizado por la instrucción siguiente.

4.3 Actualiza el punteo del estudiante. Incrementa

el contador de respuestas correctas o incorrectas

5. Módulo para la instrucción J y U.

5.1 Si la instrucción es U significa que se está llamando a una subrutina. Si es J se quiere saltar a una etiqueta.

5.2 Si la instrucción es U, guarda la dirección a donde debe regresar cuando salga de la subrutina. Usa un STACK.

5.3 Independientemente de si la instrucción es J o U llama a la subrutina HASHG para buscar la etiqueta hacia donde se quiere saltar dentro del programa.

5.4 Si la etiqueta se encuentra, se determina el registro o dirección hacia donde saltará. Esta dirección se encuentra almacenada en la tabla para etiquetas.

5.5 Si no se encuentra la etiqueta se avisa al usuario y se continúa con la ejecución.

5.6 Regresa al programa principal.

6. Módulo para la instrucción E.

6.1 Se encuentra la dirección de retorno la cual está almacenada en un stack.

6.2 Regresa al programa principal.

7. Módulo para la instrucción C.

7.1 Borra la pantalla y posiciona el cursor en la fila uno y columna uno de la misma.

7.2 Regresa al programa principal.

8. Módulo para la instrucción D.

8.1 Obtiene el tiempo que el maestro desea retardar el programa. Este tiempo se encuentra en la lección y forma parte de la misma.

8.2 Llama a la subrutina EXEC para obtener la hora actual.

8.3 Espera hasta que haya transcurrido el tiempo estipulado por el maestro. Este tiempo es igual la hora actual más el tiempo que se quiere esperar.

8.4 Regresa al programa principal.

9. Módulo para la instrucción P.

9.1 Obtiene la fila y columna donde pondrá el "cursor" del archivo de la lección.

9.2 Posiciona el "cursor". Lo hace llamando a la subrutina CURSR.

9.3 Regresa al programa principal.

Nombre de la subrutina: HASHG

Descripción:

Esta subrutina realiza las operaciones de búsqueda e inserción de las etiquetas y variables en sus tablas correspondientes.

Las variables y etiquetas usadas en HPILOT están en código ASCII, este es convertido a un código numérico que se obtiene del nombre de la etiqueta o variable con el propósito de poder aplicar la función de Hash utilizada para la inserción y búsqueda.

Una vez encontrado el código de la variable o etiqueta se procede a aplicar la función de Hash para encontrar la posición que le corresponde dentro de la tabla (VRIABL si es una variable y LABELS si es una etiqueta).

Las tablas son tratadas como tablas circulares, por lo que contienen el área de desborde en la misma tabla.

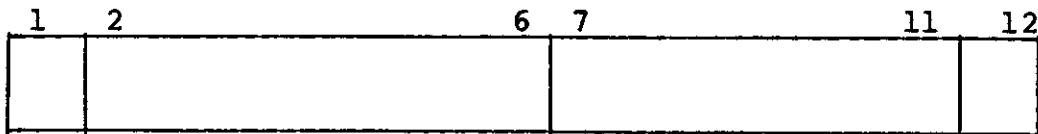
Estructuras de datos utilizadas:

La estructura de datos principal utilizada por esta subrutina es una matriz que almacena o guarda las etiquetas o variables que son utilizadas en un programa en HPILOT.

El nombre de la matriz es ITABLE y puede representar cualquiera de las dos siguientes tablas:

VRIABL o LABELS

Cuando representa a la tabla VRIABL (tabla para etiquetas) su estructura es la siguiente:



<u>Posición en palabras</u>	<u>Contenido</u>
-----------------------------	------------------

- | | |
|--------|--|
| 1 | Registro del archivo donde se encuentra la variable. |
| 2 - 6 | Nombre de la variable. |
| 7 - 11 | Contenido de la variable, o lo que es lo mismo: la respuesta del estudiante. |
| 12 | Señalador (<u>Pointer</u>) a la posición en la tabla auxiliar donde se encuentra la continuación de la respuesta del estudiante. |

Cuando esta tabla representa a la tabla LABELS tiene la siguiente estructura:



<u>Posición en palabras</u>	<u>Contenido</u>
-----------------------------	------------------

- | | |
|-------|--|
| 1 | Registro del archivo donde se encuentra la etiqueta. |
| 2 - 6 | Nombre de la etiqueta. |

Cuando la respuesta del estudiante no cabe en la tabla VRIABL por tener más de 10 caracteres se utiliza una tabla auxiliar que tiene la siguiente estructura:

1	5
---	---

<u>Posición en palabras</u>	<u>Contenido</u>
-----------------------------	------------------

1 - 5	Continuación de la respuesta del estudiante.
-------	--

Módulos principales de la subrutina:

1. El módulo de inserción trabaja de la siguiente forma:
 - 1.1 Si la posición que le corresponde a la variable o etiqueta está disponible (vacía) se realiza la inserción.
 - 1.2 Si no está disponible:
 - 1.2.1 Comparamos si lo que ya está en la tabla tiene el mismo nombre; de ser así simplemente reinicializamos la tabla en esa posición. (Esto implica que para las variables o etiquetas que tiene el mismo nombre dentro del programa se utiliza la misma posición dentro de la tabla).

- 1.2.2 Si no tiene el mismo nombre obtenemos la siguiente posición y vamos a 1. (La siguiente posición se obtiene secuencialmente, no se aplica de nuevo la función de hash).
- 1.3 Se repite el ciclo hasta recorrer toda la tabla o realizar la inserción. Si tenemos que recorrer toda la tabla se indica al usuario que la tabla está llena.
2. El módulo de búsqueda trabaja así:
 - 2.1 Buscamos en la posición que le corresponde según la función de hash.
 - 2.2 Si está en esa posición asignamos un valor de "verdadero" a la variable que nos indica que hemos encontrado la etiqueta o variable.
 - 2.3 Si no está en esa posición, nos preparamos para buscar en la siguiente posición (secuencialmente) y vamos a 1.
 - 2.4 El ciclo se repite hasta encontrar la etiqueta o la variable o cuando hemos recorrido toda la tabla infructuosamente esto implica que la variable no se encuentra, y así se le hace saber al usuario.
 - 2.5 Regresamos al programa o rutina que hizo la llamada.

Función de hash utilizada:

La función de hash que se utiliza en esta subrutina es la de la división o módulo. El módulo que se le

saca a la clave (código real de la etiqueta o variable) es 53, ya que la tabla tiene 52 posiciones disponibles.

POS IFIX (AMOD (CLAVE, DIVSOR) 1)

donde DIVSOR es igual a 53.

Los conflictos múltiples son tratados secuencialmente dentro de la misma tabla y ésta es tratada como una tabla circular.

Cuando la tabla se llena simplemente se le avisa al usuario que la misma ya está llena, lo cual parece una restricción pero dado al tamaño que pueden alcanzar las lecciones se considera que es probable que nunca se llene.

El tamaño de la tabla permite guardar hasta 52 variables o etiquetas.

Flujo:

1. Determinar cuál es la posición que le corresponde a la etiqueta o variable. Esto se hace aplicando la función de hash.
2. Determinar qué tipo de operación se desea realizar en la tabla: Inserción o Búsqueda.
3. Si es inserción insertamos en la que corresponde según la función de hash o en la primera disponible en la tabla (secuencialmente).
4. Repetir el ciclo hasta realizar la inserción o has-

ta determinar que la tabla está llena.

5. Si es una búsqueda, buscamos en la posición correspondiente (según la función de hash) o en la siguiente si no se cumple con lo anterior, es decir, si no está en la que le corresponde.
6. Repetir el ciclo hasta encontrar la variable o la etiqueta o determinar que ésta no existe en la tabla correspondiente.
7. Regresar al programa que hizo la llamada.

Ejemplo: Supongamos que en el programa encontramos una instrucción de este tipo:

```
/ *LABEL1
```

Esta nos indica que debemos guardar la etiqueta en la tabla (está indicada por el asterisco). Así es que se obtiene el código decimal correspondiente (312531) y se le aplica la función. Determinamos si está disponible o no y lo insertamos en su posición correspondiente o en la siguiente disponible.

Si encontramos una instrucción como la siguiente:

```
/ J:LABEL 1
```

Nos indica que queremos hacer una búsqueda (indicada por la J que significa un jump a una instrucción). Luego aplicamos la función y la buscamos en esa posición si está ahí regresamos al programa que hizo la llamada y si no buscamos en las siguientes hasta encontrarla y deter-

minar en qué registro del archivo que contiene la lección está la etiqueta. El salto se hace a un registro del archivo que está indicado en la misma tabla que contiene la etiqueta como se puede ver en la descripción de la tabla correspondiente en esta misma documentación.

El caso anterior ejemplifica el uso de la subrutina HASHG para trabajar con las etiquetas que se usen en HPILOT.

Para el caso de la variable esta subrutina se utiliza de la misma forma, ya que es transparente para ella si se está trabajando con una etiqueta o con una variable.

Si encontramos una instrucción como esta:

/ A:\$NOMBRE

Significa que queremos guardar una variable y su contenido en la tabla correspondiente, por lo que procedemos de la misma forma que lo haríamos con una etiqueta.

Y si encontramos una instrucción como ésta:

/ T: la respuesta \$IANS no es la correcta.

Significa que queremos buscar una variable en su tabla correspondiente y procedemos como ya se explicó anteriormente en el caso de querer buscar una etiqueta.

Nombre de la subrutina: GETO

Descripción:

Esta subrutina realiza concatenación del texto de una

línea de una lección con los contenidos o textos de las respuestas de los estudiantes.

La concatenación se realiza dentro de la variable de edición; (la variable de edición es la que se imprime en la pantalla). Si en caso no se encuentra una variable, solamente se vacía el texto de la lección a la variable de edición.

La concatenación consiste en insertar el texto de la respuesta del estudiante en la variable de edición para poder desplegar en la pantalla la respuesta del estudiante. La inserción puede ocurrir en cualquier parte de la variable de edición, la única restricción es que la variable de edición tiene sólo 72 caracteres que comprenden el texto y la respuesta.

Para encontrar el texto de la variable (i.e. la respuesta del estudiante) se llama a la subrutina HASHG.

Estructuras de datos utilizadas:

VRIABL: Tabla que contiene las variables utilizadas en HPILOT. Su estructura está descrita en la subrutina HASHG.

VTABLE: Tabla que contiene la parte de la respuesta del estudiante que no cupo en VRIABL por tener más de 10 caracteres. Su estructura es la descrita para LTABLE.

EDIT: Vector que se utiliza para la edición del texto de la lección (una línea en HPILOT). En esta varia-

ble se realiza la concatenación de una línea y una respuesta.

ITEXTO: Vector que contiene el texto de una línea de una lección en HPILOT.

Módulos de la subrutina:

1. Módulo de vaciado del texto de la línea en la variable de edición.
 - 1.1 Repetir el módulo de vaciado (vaciar ITEXTO en IEDIT) mientras no se encuentre una variable en la otra. Cuando se haya terminado regresar a la rutina desde donde se llamó. (El vaciado se realiza caracter por caracter.)
2. Módulo para conseguir el texto de la respuesta del estudiante y hacer la concatenación de ambos textos dentro de la variable de edición.
 - 2.1 Buscar la variable en la tabla (usando la subrutina HASHG).
 - 2.2 Concatenar el texto de la línea de la lección, con el texto de la respuesta.
 - 2.3 Regresar al programa que hizo la llamada.
3. Módulo para obtener el complemento de la respuesta del estudiante, que no cupo en la tabla principal, de la tabla auxiliar.
 - 3.1 Buscar el complemento de la respuesta del estudiante, en la posición señalada desde la tabla principal.

3.2 Terminar la concatenación.

3.3 Regresar al programa que hizo la llamada.

Este módulo se ejecuta cuando la respuesta del estudiante es mayor de 10 letras y/o dígitos.

La longitud máxima de una respuesta es de 20 caracteres, sin embargo, en la tabla principal sólo se guardan los 10 primeros (se tomó como principio que la mayoría de respuestas no excederán de 10 caracteres), por lo que es necesario utilizar una tabla auxiliar para guardar el texto de las respuestas que excedan de los 10 caracteres.

La razón para hacer esto es tratar de hacer un uso más eficiente de la memoria utilizada por el programa.

Flujo:

1. Vaciar el texto de la lección en la variable de edición hasta vaciarlo todo o encontrar una variable de HPILOT dentro del texto de la línea. Si se encuentra una variable de HPILOT hay que proceder con la concatenación.
2. Si se encontró una variable de HPILOT:
 - 2.1 Llama a la subrutina HASHG para buscar el texto o contenido de la variable encontrada en la línea de la lección.
 - 2.2 Obtiene el texto de la respuesta del estudiante y efectúa la concatenación de los 10 primeros caracteres.

2.3 Si la respuesta contiene más de 10 caracteres busca en la tabla auxiliar y termina la concatenación.

3. Termina de vaciar el texto de la línea en la variable de edición.

4. Regresa al programa que hizo la llamada.

Ejemplo:

Supongamos que el texto de una línea de la lección en HPILOT es el siguiente:

/ T:Cuál es tu nombre? o

/ T:Lo siento la respuesta no es la correcta.

En ambos casos habrá que vaciar el texto en la variable de edición (EDIT), para lo cual se ejecuta únicamente el módulo de vaciado, de tal forma que el estudiante verá en su pantalla lo siguiente:

Cual es tu nombre?

Lo siento la respuesta no es la correcta.

Si el texto de la lección fuera el siguiente:

/ T:Hola \$NOMBRE, estás listo para empezar? o

/ T:Lo siento, \$RES no es correcto.

Para efectuar la edición esperada por el maestro del texto de la lección, se ejecutarán los módulos de vaciado y el módulo que busca la variable que contiene el nombre del estudiante (\$NAME) o la variable que contiene la respuesta del estudiante (\$RES). Así es que lo que el estudiante vera en su pantalla es lo siguiente:

Hola Pepe, estás listo para empezar o

Lo siento París no es correcto.

Si el texto de la respuesta excediera de los 10 caracteres se ejecutaría el módulo que busca en la tabla auxiliar para obtener el texto completo.

/ T: Ahora desarrolla este binomio $(a + b)^3$

/ A: 3, \$ANSWER

(1) / M: $a^3 + 3a^2b + 3ab^2 + b^3$

/ T:

/ TY: Correcto, \$NOMBRE

/ TN: Lo siento, \$ANSWER NO es la respuesta correcta.

En este caso lo que el estudiante verá en la pantalla depende de si la respuesta que él dio es correcta o no.

En caso fuera correcta verá lo siguiente:

lo siento, $\frac{a^3 + 3ab^2 + b^3}{(2)}$ no es la respuesta correcta.

(1) es la respuesta correcta.

(2) es la respuesta del estudiante

Nombre de la subrutina: CONVR

Descripción:

La subrutina CONVR convierte una cadena de caracteres alfanuméricos a una cadena numérica con dígitos del 0 al 9 del tal forma que le hace corresponder a cada letra y dígito de la cadena un dígito.

Esta subrutina le proporciona a la subrutina HASHG el código numérico correspondiente para encontrar la posición de la variable o etiqueta en la tabla correspondiente.

Estructuras de datos utilizadas:

ITEXTO: Contiene la cadena alfanumérica (ASCII) que representa una etiqueta o una variable en HPILOT.

CODIGO: Variable que contiene el código numérico que le corresponde a la variable o etiqueta contenida en ITEXTO.

TABLE: Tabla de referencia para encontrar el código numérico.

Su estructura es así:

ABCDEFGHIJKLMNPNOPQRSTUVWXYZ0123456789

(2 letras/dígitos x palabra)

ICODE: Tabla que contiene el número correspondiente a cada letra y/o dígito de la variable o etiqueta.

Su estructura es así:

123456789123456789123456789123456789

(2 dígitos x palabras)

Flujo:

1. Repite el ciclo de conversión (obtiene para cada carácter en ITEXTO un dígito en ICODE) hasta que termina con todas

las letras y/ o dígitos de la variable o etiqueta o hasta que lo ejecuta 10 veces.

1.1 Buscar y comparar cada letra y/o dígitos de la variable con cada uno de los elementos de TABLE para asignarles el código numérico dentro del ICODE.

2. Finaliza la conversión y regresa al programa o rutina que hizo la llamada.

Ejemplo:

Supongamos que dentro de ITEXTO viene la etiqueta LABEL1, la subrutina CONVR le asignaría un código numérico equivalente a 312531 (este código está aún en código ASCII), luego este código ASCII se convierte a un número decimal equivalente a 312531, finalmente éste es el código que se le regresa a la subrutina HASHG para encontrar la posición de la variable o etiqueta en su tabla.

IV. OBSERVACIONES ADICIONALES SOBRE HPILOT

HPILOT es un interpretador de instrucciones de uno a tres caracteres como máximo; posee las instrucciones básicas para realizar operaciones de entrada y salida, bifurcación incondicional, bifurcación condicional, subrutinas y terminación de ejecución.

La operación de entrada de datos se refiere a aceptar una cadena alfanumérica desde la terminal del usuario; esta cadena puede variar desde un solo caracter hasta un máximo de veinte. La cadena puede ser guardada bajo un nombre en una tabla, donde queda registrado el nombre de la variable y su contenido.

La operación de salida se refiere a la impresión del texto de una lección. El texto puede comprender una línea de una lección y una variable. La impresión se realiza en la terminal del usuario.

La operación de bifurcación incondicional se refiere a alterar el flujo de la ejecución del programa hacia una posición etiquetada especificada en la instrucción. La etiqueta presenta un registro del archivo que contiene la lección. La dirección hacia donde debe "saltar" el programa se encuentra almacenada en una tabla que contiene todas las direcciones a donde "saltará" el programa en un determinado momento: estas

direcciones se determinan inicialmente por HPILOT antes de iniciar la ejecución de un programa leyendo todo el archivo que contiene para encontrar todas las etiquetas de la lección.

La bifurcación condicional funciona de la misma manera que la operación anterior excepto que debe cumplirse una condición para que se realice la operación. El cumplimiento de la condición se establece mediante la respuesta del estudiante a una pregunta de la lección.

La terminación de la ejecución de un programa en HPILOT se realiza con la instrucción `"/E"`, la cual siempre se coloca al final del programa, es decir debe ser la última instrucción del programa. HPILOT no posee una instrucción equivalente a un `"STOP"` de otros lenguajes.

Las instrucciones para simular las llamadas a subrutinas y terminación de las mismas son `"U"` y `"E"` respectivamente las cuales equivalen a un `"GOSUB"` y `RETURN` en BASIC.

En la implementación de las instrucciones descritas anteriormente, excepto `"/E"`, se utilizaron estructuras de datos como tablas de hash y stacks. Las tablas se usaron en la implementación de las instrucciones de entrada/ salida, bifurcación condicional e incondicional; los stacks para la instrucción `"E"`.

Se utilizó la técnica de "hash" por considerarla más eficiente en cuanto a tiempo de acceso y más elegante que una búsqueda secuencial. La función $(X \text{ MOD } Y)$ fue la se usó para encontrar la dirección o posición de una variable o etiqueta en su respectiva tabla. La descripción y contenido de cada tabla están en la documentación técnica.

Las colisiones se manejan dentro de la misma tabla; es decir, la tabla se trata circularmente. El uso de esta técnica se debe a que no se espera que haya un número significativo de colisiones y que en el peor de los casos (que a todas las variables o etiquetas les corresponda la misma posición) únicamente habría que hacer una búsqueda secuencial en una tabla relativamente pequeña (52 posiciones como máximo) lo cual no degrada el tiempo de búsqueda ni de inserción. Existen técnicas más elegantes para tratar las colisiones y desbordes pero no se justifica su utilización en este caso ya que ni el volumen de etiquetas ni el de variables es muy grande, por lo que no hubiera mejorado en materia de tiempo de acceso.

El uso de Stacks para el diseño de la instrucción "E" se debe a que se consideran ideales para la simulación de la instrucción "RETURN". El contenido de los stacks son direcciones que representan registro del archivo que contiene una lección en HPILOT.

La simulación de la instrucción "E" funciona de la siguiente manera: cuando se encuentra una instrucción "U" se guarda en el stack el número del registro que sigue al que contiene la instrucción de llamada y cuando se encuentra en el la subrutina la instrucción "E" se toma del stack el registro hacia donde debe dirigirse la lectura del archivo.

HPILOT no es solamente un intepretador sino además permite llevar un control del rendimiento académico del estudiante en las diferentes áreas donde se use HPILOT. Tanto el control académico como las lecciones están guardados en archivos. Por esta razón se han empleado dos tipos u organizaciones de archivos: de acceso secuencial (tipo 4 para HP) y de acceso directo (tipo 2 para HP).

El empleo de archivos secuenciales se justifica porque HPILOT permite al usuario utilizar el programa producto de HP EDIT/1000, el cual le permite al maestro (usuario) actualizar una lección. Sin embargo EDIT/1000 maneja únicamente archivos tipo 4 (acceso secuencial y longitud de registro variable) y HPILOT necesita para el diseño de las instrucciones de bifurcación y llamadas a subrutinas un archivo de acceso directo, ya que estas instrucciones hacen referencia a determinados registros del archivo que contiene la lección.

El registro del control académico necesita un archivo

de acceso directo, ya que HPILOT hace referencia a registros específicos durante su ejecución. La identificación del estudiante (código numérico) se convierte en el registro correspondiente.



V. SUGERENCIA PARA MEJORAR HPILOT

1. Si el maestro no quiere asignarle una nueva clave al estudiante para poder correr HPILOT, además de la que ya le ha sido asignada en su clase, podría guardarse ésta en una tabla, la cual se almacenaría en un archivo auxiliar de claves y cuando el estudiante corra HPILOT se busca con la clave en el archivo auxiliar el registro que le corresponde en el archivo de control académico para poder realizar la evaluación.

Al archivo auxiliar de claves le podría dar mantenimiento el encargado de HPILOT o el maestro mismo usando el editor, esto es si se usara un archivo de acceso secuencial. Como puede verse en el archivo de claves contendría la clave del estudiante y el registro que le corresponde en el archivo de control académico.

2. Si el maestro quisiera saber cuánto tiempo le dedicó un estudiante a una lección en particular, y que esta información se almacenará en el archivo de control académico, podría diseñarse una nueva instrucción que obtuviera la hora actual de la máquina cuando se inicia la ejecución de la lección; al final de la lección se ejecutaría la misma instrucción y se haría la resta correspondiente, determinando así con esta diferencia, el tiempo que el estudiante se tardó en el estudio de la lección.

Para completar habría que definir cinco nuevos campos en el archivo de control académico (uno para cada lección) y escribir el resultado de la resta de los tiempos en el campo correspondiente a cada lección.

Para diseñar esta instrucción, se podrían aprovechar los códigos 8, 10 ó 12 que están disponibles para nuevas instrucciones en la subrutina PRFRM.

3. La operación de concatenación que se menciona en la subrutina GETO, podría mejorarse en cuanto a no permitir que el texto de la lección y la variable de HPILOT que se quiere editar, excedan el tamaño de la variable de edición IEDIT.

VI. RESUMEN

1. Las lecciones en HPILOT están almacenadas en un archivo que puede ser modificado por el maestro.

El número máximo de lecciones que puede manejar HPILOT es cinco (numeradas del 1 al 5). Cualquier número mayor que cinco no es permitido.

El maestro puede modificar una lección simplemente llamando al EDITOR y dando el nombre de la lección a la que le desea hacer los cambios

2. HPILOT lleva un control del rendimiento académico del estudiante usando un archivo que el maestro crea para dicho propósito. Este archivo contiene el nombre del estudiante o los punteos que haya obtenido en la diferente lecciones, las cuales están divididas en grupos de respuestas (del 0 al 9).
3. El archivo del control académico puede ser modificado por el maestro, ya sea para ingresar un nuevo estudiante, eliminar un estudiante, o permitir a un estudiante repetir una nueva lección.
4. El maestro puede obtener una lista de la lección y del archivo de control académico. Ambos archivos pueden listarse en la impresora del sistema o en la pantalla del usuario.
5. El maestro puede usar la computadora para que el estudiante tome las lecciones que haya diseñado.

6. Para el diseño de las lecciones, el maestro cuenta con un lenguaje de programación elemental, con instrucciones sencillas. Estas instrucciones permiten al maestro imprimir en la pantalla el texto de la lección, aceptar y evaluar respuestas del estudiante, controlar el flujo de la ejecución de las lecciones. Además cuenta con otras instrucciones que se detallan en el manual del usuario.

BIBLIOGRAFIA

- ACM. Sigcue Bulletin. Computer Uses in Education. New
1980 York: Association for Computer Machinery, Vol. 14
1981 Nos. 2 y 3 (April, July) y Vol. 15 Nos. 1, 2 y 3
(January, April y July).
- Albrecht, Kurt. "An Educators Best Friends. Programmed
1983 Inquiry, Learning or Teaching". Remark Official-
Magazine for users of Heath Zenith computer equip-
ment (issue 39 April). Pág. 28-33.
- Batch Spool Monitor. Reference Manual. Part 92060-
1980 90013. Fifth Edition. (USA) April. Hewllet-Packard
Company.
- Dick, Walter y Lou Carey. The Systematic Design of
1978 Instruction. USA. Scott, Foresman and Company,
216 págs.
- EDIT/1000. Users Guide. Part 92074-90001. First
1982 Edition. USA (August, July) Hewllet-Packard Comp.
- Gagne, Robert M. y Leslie Briggs. Principles of
1979 Instructional Design. USA Holt, Rinehart and Wilson.
321 págs.
- Gago Hugust, Antonio. Modelo de Sistematización del
1977 proceso enseñanza-aprendizaje. México: Editorial
Trillas. 81 págs.
- Hugnes, Charles et al. Advanced Programming
1978 Techniques. A Second Course in Programming Using
Fortran. Usa: John Wiley & Sons Inc.
- RTE FORTRAN IV. Reference Manual. Part No. 92060-
1980 90023 Seventh edition. Printed in USA, Hewllet-
Packard Company. Data Systems División. Cupertino,
California (March).
- RTE IV Programer's. Reference Manual. Part. No.
1981 92068- 90004. Second Edition. (USA) (Jan. 1980-
Oct. 1981). Hewllet Packard company.