PROGRAMANDO SOBRE CALCULADORAS SERIE HP-48 Y HP-50 EN LENGUAJE USER-RPL Y HP50PL

Por Jaime Meza

Versión 0.1.0 Revisión 2, feb 15 del 2017

Comments, suggestions, etc. are welcome at eonicasys@gmail.com

Updates of this document go to: www.eonicasys.com.co

ÍNDICE

0: INTRODUCCIÓN BREVE AL USER-RPL Y HP50PL

Los lenguajes que puede soportar la calculadora hp50 es lenguaje interpretado (HP50PL y USER-RPL) y compilados como (ASM, C y SYSTEM-RPL)

Este documento en un comienzo solo muestra el lenguaje que yo lo denomino (HP50PL) sigla proveniente de [HP50 Programming Language], el HP50PL también es conocido como programación en modo algebraico pues usa la notación similar a la mayoría de los lenguajes computacionales, luego en una segunda parte se muestra el lenguaje en notación RPN, sigla proveniente de (notación polaca inversa ó notación de postfijo) que es el modo inverso de operación de la notación polaca (PN). el lenguaje en notación RPN se denota como RPL y existen dos versiones para las calcualdoras serie HP48 y HP49(50), una llamada USER-RPL sigla proveniente de (USER-Reverse Polish Lisp) y SYSTEM-RPL.

El PN/RPN es importante por ejemplo en el desarrollo de intérpretes y lenguajes para crear sistemas expertos como CAS (computer Algebra Systems) como lo es MAXIMA

RPN info:

https://es.wikipedia.org/wiki/Notaci%C3%B3n_polaca_inversa

PN info:

https://es.wikipedia.org/wiki/Notaci%C3%B3n_polaca

1: Caracteres de edición

Un programa para las calculadoras Hewlett Packard HP-48 y HP-50 usa 256 caracteres, los 128 caracteres ASCII

```
 ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?@
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _
 `a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
```

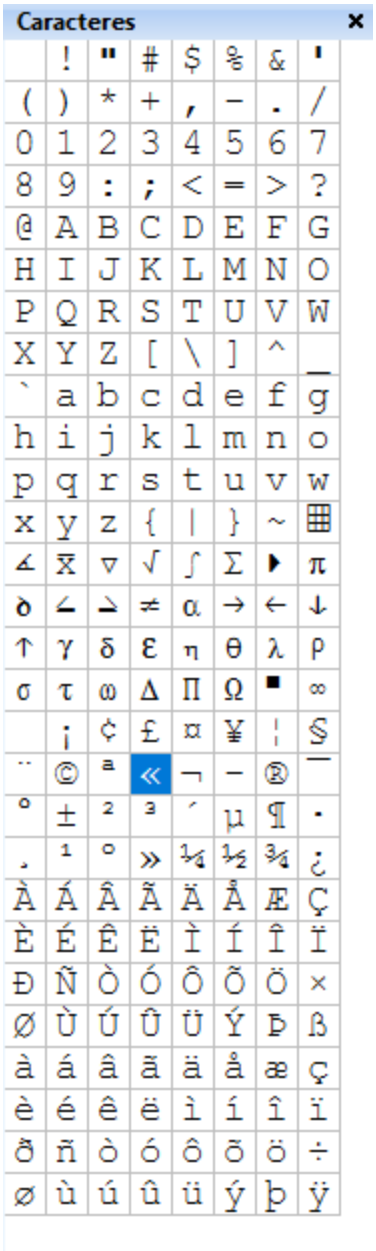
y los caracteres de (ISO Latín 1) https://es.wikipedia.org/wiki/ISO/IEC_8859-1 pero con algunos símbolos modificados, de ahí la necesidad de preinstalar la fuente HP48.TTF (anexa en el siguiente archivo zip)

Los siguientes caracteres deben verse igual a la siguiente imagen y tabla, si la fuente de texto, está debidamente preinstalada HP48.TTF

```
▦ ∡ x̄ ▽ √ ∫ Σ ▸ π ∂ ≤ ≥ ≠ α
→ ← ↓ ↑ γ δ ε η θ λ ρ σ τ ω Δ Π Ω ■ ∞ € ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ - ® ¯ ° ± ² ³ ´
µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

La vista o tabla de edición de caracteres se muestra en la siguiente graficas

```
▦ ∡ x̄ ▽ √ ∫ Σ ▸ π ∂ ≤ ≥ ≠ α
→ ← ↓ ↑ γ δ ε η θ λ ρ σ τ ω Δ Π Ω ■ ∞ € ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ - ® ¯ ° ± ² ³ ´
µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```


Caracteres
! " # $ % & '
( ) * + , - . /
0 1 2 3 4 5 6 7
8 9 : ; < = > ?
@ A B C D E F G
H I J K L M N O
P Q R S T U V W
X Y Z [\ ] ^ _
` a b c d e f g
h i j k l m n o
p q r s t u v w
x y z { | } ~
∠ x̄ ▽ √ ∫ Σ ▶ π
∂ ≮ ≯ ≠ α → ← ↓
↑ γ δ ε η θ λ ρ
σ τ ω Δ Π Ω ■ ∞
¡ ¢ £ ¤ ¥ ¦ §
¨ © ª « ¬ - ® ¯
° ± ² ³ ´ µ ¶ ·
¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç
È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö ×
Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç
è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷
ø ù ú û ü ý þ ÿ

2: CUERPO DE UN PROGRAMA Y ALMACENAMIENTO EN MEMORIA.

Los símbolos o caracteres contenedores del cuerpo de un programa por ejemplo en lenguaje C son un par de corchetes: { }. En HP50PL usa los símbolos de comillas latinas de apertura y de cierre, caracteres «…» respectivamente. También los caracteres anteriores son conocidos como paréntesis angulares dobles.

Ejemplo #01

SEUDOCÓDIGO	
Proceso ejemplo01() // esto es un comentario Limpiar Pantalla; Escribir "Hola Mundo"; Fin_Proceso	
HP50PL (código del cuerpo del programa equivalente)	
« `CLLCD`; `DISP("Hola Mundo",1)`; `FREEZE(7)`; »	 @ ESTO ES UN COMENTARIO, usa el símbolo arroba @ limpia la pantalla @ imprime en la primera línea del LCD una cadena de caracteres @ retiene o congela las 7 líneas del LCD

```
«
                              @ ESTO ES UN COMENTARIO, usa el símbolo arroba
  `CLLCD`;                    @ limpia la pantalla
  `DISP("Hola Mundo",1)`;     @ imprime en la primera línea del LCD una cadena
  `FREEZE(7)`;                @ retiene o congela las 7 líneas del LCD
»
```

Aclaraciones:

- Los comentarios inician con el carácter arroba @

- Cada sentencia en HP50PL debe también encerrarse, contenerse o delimitarse entre un par de caracteres `…` (que es el carácter para acento grave, ASCII # 96), se puede obviar los delimitadores `…` cuando es un test y o si solo es una orden que no posee argumentos entre paréntesis

Ejemplo: `CLLCD`; es igual a CLLCD; esta orden proviene del acrónimo de las palabras (Clear LCD)

Mientras que en la siguiente sentencia se requiere los delimitadores `…` por que posee argumentos.

Ejemplo: `DISP("Hola Mundo",1)`;

- Si se desea que cada sentencia u objeto NO quede en el historial de niveles se debe finalizar con punto y coma, similar a como lo hace MatLab.

- El nombre del programa va después del cuerpo.

La orden STO proviene de STORAGE (almacenamiento), como el HP50PL se basa en un lenguaje de pila o niveles (puede combinar notación RPN con HP50PL)

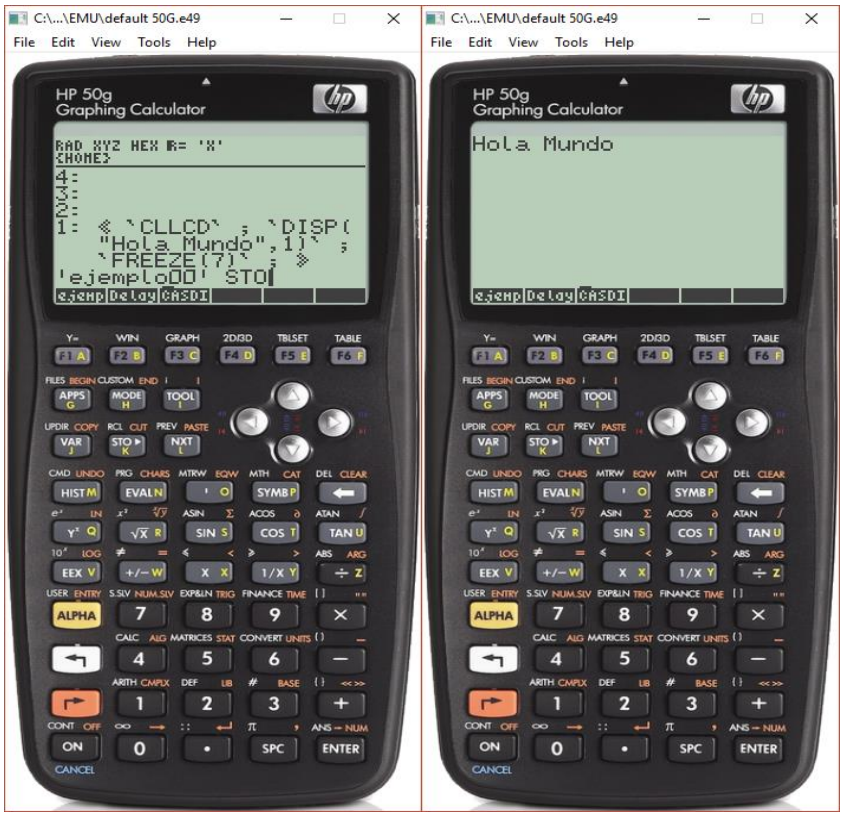
programa 'nombreDelPrograma' STO

finalmente, el nombre del programa tiene que estar entre un par de comillas simples 'ejemplo01' (que es el carácter para apóstrofo ASCII # 39), esto debido a que las comillas evitan la evaluación o ejecución del programa mientras se almacena en memoria.

Ejemplo #01a (completo)

SEUDOCÓDIGO	
Proceso ejemplo00() // esto es un comentario Limpiar Pantalla; Escribir "Hola Mundo"; Fin_Proceso	
HP50PL (código del cuerpo del programa equivalente)	
« `CLLCD`; `DISP("Hola Mundo",1)`; `FREEZE(7)`; » 'ejemplo01' STO	@ inicio del cuerpo del programa @ ESTO ES UN COMENTARIO, usa el símbolo arroba @ limpia la pantalla @ imprime en la primera línea del LCD una cadena de caracteres @ retiene o congela las 7 líneas del LCD @ fin del cuerpo del programa @ almacena en memoria

```
«
                              @ ESTO ES UN COMENTARIO, usa el símbolo arroba
  `CLLCD`;                    @ limpia la pantalla
  `DISP("Hola Mundo",1)`;     @ imprime en la primera línea del LCD una cadena
  `FREEZE(7)`;                @ retiene o congela las 7 líneas del LCD
»
'ejemplo01' STO
```



3: TIPOS DE DATOS

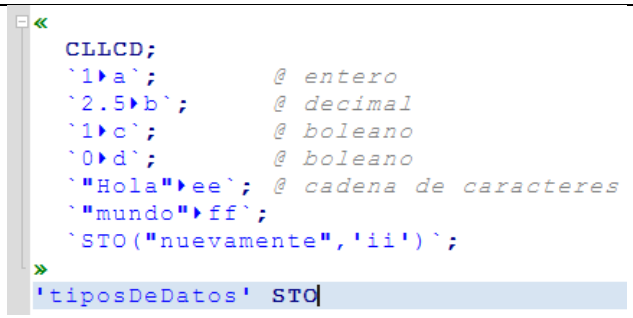
Ejemplo #02

seudocódigo

```
Proceso tipos_de_datos()
        Limpiar Pantalla;
        definir a como entero;
        a<-1;                   entero
        b<-2.5;                 decimal
        c<-verdadero;           boleano
        d<-Falso;               boleano
        e<-"Hola";              cadena de caracteres
        f<-"Hola Mundo";        cadena de caracteres
        i<-"Hola Mundo";        cadena de caracteres
FinProceso
```

HP50PL

```
«
      CLLCD;
      `1▸a`;        @ entero
      `2.5▸b`;      @ decimal
      `1▸c`;        @ boleano
      `0▸d`;        @ boleano
      `"Hola"▸ee`; @ cadena de caracteres
      `"mundo"▸ff`;
      `STO("nuevamente",'ii')`;
»
'tiposDeDatos' STO
```



Aclaraciones:

- El almacenamiento de objetos también se puede usar la orden o símbolo ▸ de la siguiente forma

`objeto▸nombreDelObjetoOvariable`

- Los nombres identificadores (IDs) no pueden llevar sub_guiones, salvo si contienen una unidad.

- Los IDs (e, i) son palabras reservadas para la función EXPONENCIAL y la UNIDAD IMAGINARIA, por esta razón no se pueden usar como nombres de variables ni de programas.

- el valor booleano para verdadero es un número ≥ 1

- el valor booleano para falso es el número 0

- un identificador (ID) por ejemplo var1 es diferente de Var1 y de VAR1

Ejemplo #03, ciclos

seudocódigo

```
Proceso ciclos()
        c<-0;
        Mientras c<=10 Hacer
                c<-c+1;
                Escribir c;                    // IMPRIME DE 1 A 11
        FinMientras

        c<-0;
        Repetir
                c<-c+1;
                Escribir c;
        Hasta que c=10

        c<-0;
        Repetir
                c<-c+1;
                Escribir c;
        Mientras que c<10

        si c=10 Entonces
                Escribir "Si";
        FinSi

        si c=10 Entonces
                Escribir "Si";
        Sino
                Escribir "No";
        FinSi

Fin_Proceso
```

HP50PL

```
«
  CLLCD;
  `0▶c`;
  WHILE `c≤10`
  REPEAT `c+1▶c`;
    `DISP(c,1)`;
    `FREEZE(2)`;
    HALT;
  END;
  @...
»
'ciclos' STO
```

Redactando, … espere por favor más actualizaciones y ampliaciones de este documento

Enlace del gaakEmu (emulador HP50G & otros)

http://hpcalc.gaak.org/?id=98765