

Capacitación Desarrollador

INTRODUCCION A VISUAL FOXPRO6

INDICE

1. Entorno de desarrollo

- Descargar software
- Interfaz de usuario
- Conceptos básicos
- Comentarios en un programa
- Abreviatura de comandos

2. Tipos de datos

- Contenedores de datos
- Operadores

3. Funciones

4. Variables

- Notación de Variables
- Flexibilidad de Variables

INDICE

5. Ciclos

- For
- For each
- Do while sencillo
- Do while/loop/exit
- Scan

6. Funciones y variables

- Variables publicas
- Variables locales
- Variables privadas

7. Parámetros y arreglos

- Parámetros por valor
- Parámetros por referencia
- Arreglos por referencia

8. Clases y objetos

Entorno de desarrollo

INTRODUCCION A VISUAL FOXPRO6

SOFTWARE VFP6

Descargar Visual FoxPro 6 con el siguiente enlace.

<http://sait.com.mx/download/vfp6.zip>

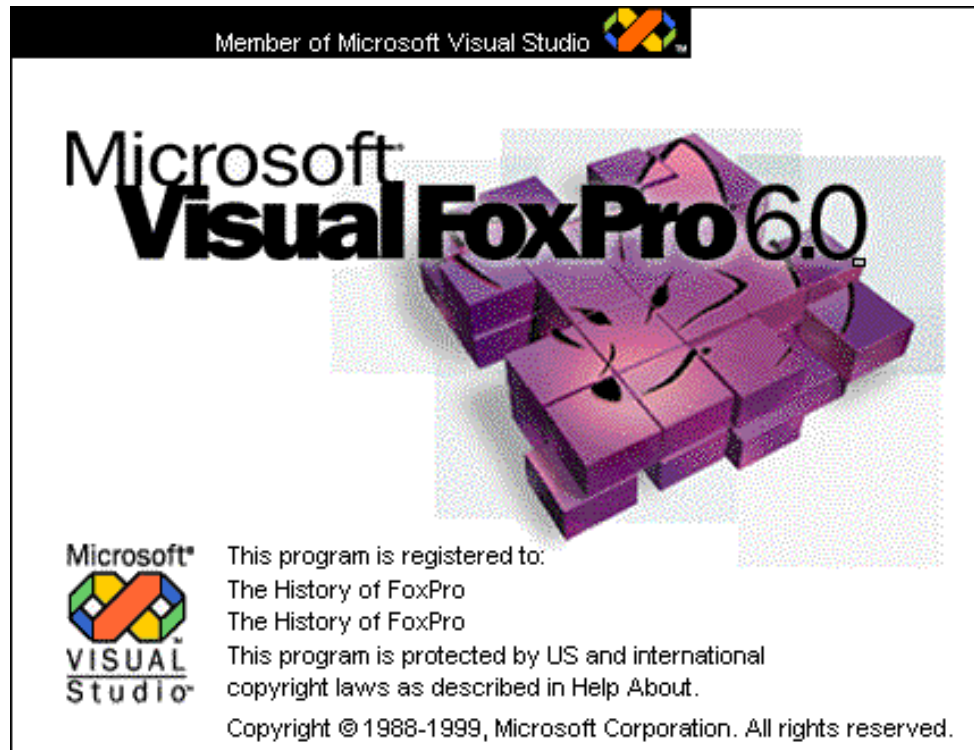
Para su instalación descomprima el archivo en el directorio C:\vfp6

CAPACITACION

DESARROLLADOR

INTRODUCCION A VISUAL FOXPRO6

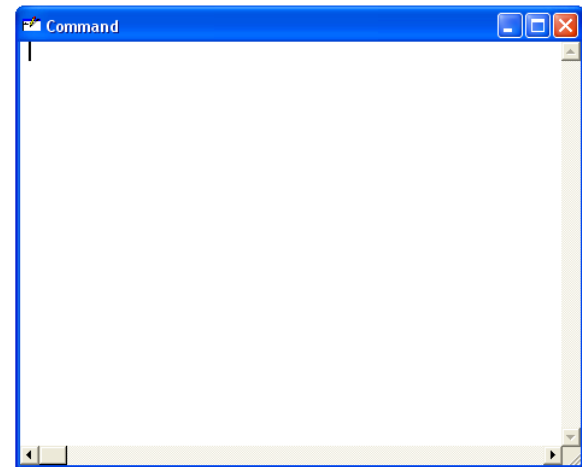
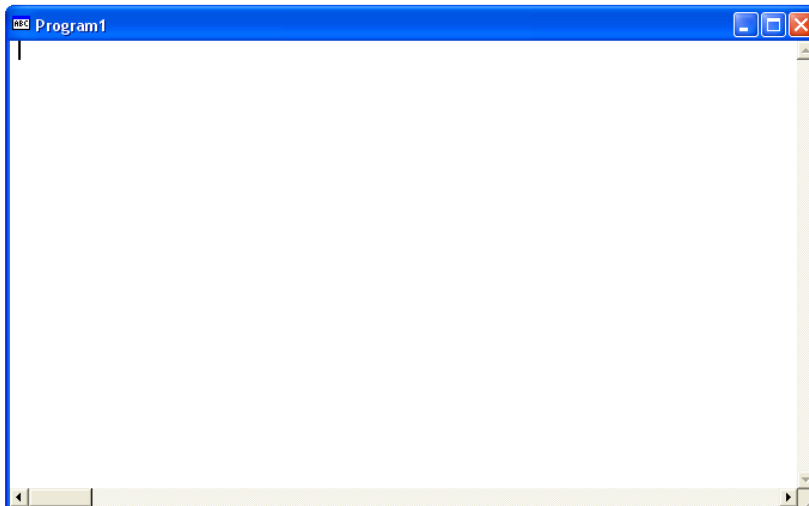
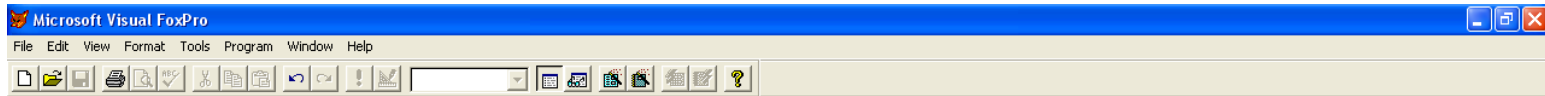
INTERFAZ DE USUARIO



CAPACITACION DESARROLLADOR

INTRODUCCION A VISUAL FOXPRO6

INTERFAZ DE USUARIO



CONCEPTOS BASICOS

Programa helloworld

1. Para crear o editar un programa utilice el siguiente comando.
? 'Hola Mundo'.prg
2. Se abrirá una ventana en la cual escribiremos el código del programa
? 'Hola Mundo'

? 'Hola'

?? ' Mundo'

*? Símbolo de interrogación para imprimir el resultado en una nueva línea.
?? Imprimir resultado en la misma línea.*

Ejemplo	Resultado
? 10	10
? 'Ignacio'	Ignacio
nTotal = 1500 ?nTotal	1500

CONCEPTOS BASICOS

Programa helloworld

3. Para comprobar si no existen errores en el código se utiliza la compilación.

Program / Compile

Archivo con extensión .FXP es el resultado de la compilación de un programa en Visual FoxPro.

NOTA:

- El archivo .FXP puede ser ejecutado sin que exista el programa base .PRG
- .FXP no permite la edición del mismo, solo su ejecución.

4. Para ejecutar el programa utilice el siguiente comando.

1. helloworld()
2. do helloworld

*Se puede utilizar cualquier de las opciones anteriores sin embargo por similitud con otros lenguajes se utiliza la opción 1 **helloworld()***

COMENTARIOS EN UN PROGRAMA

Utilizados para especificar una instrucción o realizar anotaciones.

1. **símbolo asterisco *** al inicio
2. **símbolo ampersand &&** después de una línea de código.

Los comentarios se identifican por cambiar su color a verde.

Ejemplo

*	*Este es un comentario
&&	? 'Hola Mundo' &&Este es un comentario

ABREVIATURA DE COMANDOS

Visual FoxPro Identifica comandos en abreviaturas de 4 letras.

Ejemplo

modify command helloworld

modi comm helloworld

? round (19.5,0)

? roun(19.5,0)

Tipos de Datos

INTRODUCCION A VISUAL FOXPRO6

TIPOS DE DATOS

Tipo	Ejemplo
NUMERIC	3 3.14
CHARACTER	"Hola" 'Hola'
LOGICAL	.t. (Verdadero) .f. (Falso)
DATE	{^1970/12/03}
DATETIME	{ {1970/12/03 14:35:59} }

CONTENEDORES DE DATOS

Tipo	Ejemplo
VARIABLES	nTotal = 10 m.Nombre = 'Ignacio'
ARREGLOS NOTA: INICIAN EN 1 HASTA N	dimension nombres[3] nombres[1] = 'Hugo' nombres[2] = 'Paco' nombres[3] = 'Luis' ? nombres [1]

OPERADORES

Ejemplo	
==	if nTotal == 0
<>	if nTotal <> 0
+	X = X +1
-	Y = Y -1
/	nFactor = nValor / 1-1
*	nImporte = nTotal * nTC

Funciones

INTRODUCCION A VISUAL FOXPRO6

FUNCIONES

Ejemplo	Resultado
dHoy = date() ?dHoy	03/21/14
? time ()	08:48:30
? dow(date())	6
?cdow (date())	Friday
? week (date())	12
? cmonth(date())	March
set century on ? date()	03/21/2014

FUNCIONES

Ejemplo	Resultado
set century off ? date()	03/21/14
n1 = '7' n2= '10' ? val(n1) + val(n2)	17.00
? Int (12.55)	12
? round(199.99,00)	200
? abs(-50)	50
? mod(50,4)	2
? max (16,28,45)	45

FUNCIONES

Ejemplo	Resultado
? min (16,28,12)	12
? rand()	0.85
? sqrt (400)	20
? ceiling(140.56)	141
cMensaje='Hola' ? empty(cMensaje)	.F.
? type('(12 * 3) + 4')	N (numeric)
? lower('HOLA MUNDO')	hola mundo
? upper('hola mundo')	HOLA MUNDO

FUNCIONES

Ejemplo	Resultado
? alltrim (' hola mundo ')	hola mundo
cTitulo= 'Tutorial Visual FoxPro' ? substr (cTitulo, 1, 8) ? substr (cTitulo, 9)	Tutorial Visual FoxPro
cTitulo= 'Tutorial Visual FoxPro ' ? len (cTitulo)	22
? chr(67)	C
? left('Mexico',3)	Mex
? right('Tijuana',5)	juana

FUNCIONES

Ejemplo	Resultado
? at ('a','Hola')	4
gcString='Gato' ? strtran(gcString, 'G', 'P')	Pato

Variables

INTRODUCCION A VISUAL FOXPRO6

NOTACION DE VARIABLES

CAMEL CASE

Variables, propiedades, clases y funciones se identifican por la primera letra indicando el tipo de dato.

Ejemplo variable numérica

nTotal

Ejemplo variable date (fecha)

dFecha

Ejemplo variable caracter

cNombre

NOTACION DE VARIABLES

CAMEL CASE

Variables, propiedades, clases y funciones se identifican por la primera letra indicando el tipo de dato.

Ejemplo variable logica

lExiste

Ejemplo variable arreglo/array/matriz

aClientes

Ejemplo variable objeto

oForma

FLEXIBILIDAD DE TIPOS EN VARIABLES

Las variables pueden almacenar cualquier valor numérico, carácter, etc. por ello no es necesario declarar el tipo de dato.

Ejemplo variable Total

Total = 100

Total = 'Cien'

Total = date()

Total = .t.

Total = .f.

Total = null

Ciclos

INTRODUCCION A VISUAL FOXPRO6

CICLO FOR

Ejemplo uno.prg

```
for i=1 to 100
    ? i
next
```

CICLO FOR EACH

Ejemplo dos.prg

```
dimension aNombres[3]
aNombres[1] = 'Hugo'
aNombres[2] = 'Paco'
aNombres[3] = 'Luis'

for each cNombre in aNombres
    ? cNombre
Next
```

DO WHILE SENCILLO

Ejemplo tres.prg

```
nRenglon = 1
```

```
do while nRenglon <=50  
    ? 'Este es el renglon:', nRenglon  
    nRenglon=nRenglon+1  
Enddo
```

DO WHILE / LOOP / EXIT

Ejemplo cuatro.prg

```
nRenglon = 1
do while nRenglon <= 50
    *Saltar renglones pares
    nRenglon = nRenglon+1
    if mod(nRenglon,2)==0
        loop
    endif
    ? 'Este es el renglon:', nRenglon
    *Si nRenglon es mayor a 40, sale del ciclo
    If nRenglon>40
        exit
    endif
Enddo
```

CICLO SCAN DATABASE

Ejemplo cinco.prg

***Crear una tabla y agregar 3 registros**

```
create table clientes ( nombre c(30), ciudad c(30))  
insert into clientes values ('Juan Perez','Monterrey')  
insert into clientes values ('Jose Lopez','Mexicali')  
insert into clientes values ('Luis Ramos','Torreón')
```

***Recorrer la tabla e imprimir los datos**

```
scan  
? clientes.nombre, clientes.ciudad  
endscan
```

Funciones y Variables

INTRODUCCION A VISUAL FOXPRO6

FUNCIONES

Utilizadas para calcular y devolver un valor.

Ejemplo holamundo.prg

```
function holamundo  
    ? 'Esta es una función'  
return
```

Para ejecutar una función debe ser llamada con el nombre con el cual se definió en la ventana de comandos.

holamundo()

Las funciones deben ser llamadas como el archivo en que se encuentran.

holamundo.prg

function holamundo

VARIABLES PUBLICAS

Disponibles en cualquier parte del programa, se identifican por tener un _ al inicio del nombre.

Ejemplo

```
public _version  
_version = 6.30
```

```
public _dirInicial  
_dirInicial = 'C:\SAIT'
```

VARIABLES LOCALES

Disponibles solo dentro de la función en la que se definen.

Ejemplo areaTriangulo.prg

```
function areaTriangulo (nBase, nAltura)
    local nArea
        nArea=nBase*nAltura/2
        ? nArea
    return nArea
```

VARIABLES LOCALES

Disponibles solo dentro de la función en la que se definen.

Ejemplo factorial.prg

```
function factorial(nvalor)
local i,nTotal
    nTotal=1
    for i=1 to nvalor
        nTotal = nTotal * i
    next
?nTotal
return nTotal
```

VARIABLES PRIVADAS

No utilizarlas, disponibles debajo de cualquier función que se mande llamar.

Ejemplo procesouno.prg

```
function procesouno
  private nTotal
    nTotal =10
    procesodos()
  return nTotal
function procesodos
  nTotal = nTotal * 3.1416
  ?nTotal
Return
```

Parámetros y arreglos

INTRODUCCION A VISUAL FOXPRO6

PARAMETROS POR VALOR

Por default los parámetros son pasados por valor, es decir se crea una copia.

Ejemplo

nValor = 7

? factorial(nValor)

PARAMETROS POR REFERENCIA

Un parámetro puede ser modificado cuando se coloca el símbolo @ al inicio.

Ejemplo facturacion.prg

```
function facturacion
    cFactura = 'A199'
    dFecha = { }
    if existeFactura(cFactura, @dFecha)
        ? 'La factura ', cFactura, ' fue elaborada el día', dFecha
    else
        ? 'No existe la factura ', cFactura
    endif
return
function existeFactura (cFact, dFec)
    *Simulacion de buscar factura
    dFec = date()
    *Regresa True en caso de encontrarse
return
```


ARREGLO POR REFERENCIA

Un arreglo debe de pasarse siempre por referencia utilizando el @, de lo contrario solo enviara el primer elemento.

Ejemplo seis.prg

```
dimension aArray[3]
registro (@aArray)
?aArray[1]
function registro(aAr)
    dimension aAr[3]
    aAr[1] = "HUGO"
    aAr[2] = "PACO"
    aAr[3] = "LUIS"
return
endFunc
```

Clases y Objetos

INTRODUCCION A VISUAL FOXPRO6

OBJETOS

Creación de un objeto, asignándole valor, llamar método.

Command	
Crear Objeto	<code>oForma = newObject('form')</code>
Llamar/Invocar el método show() para mostrar la forma	<code>oForma.show()</code>
Modificar propiedades	<code>oForma.caption = 'MiApp Personal'</code>
Propiedad ancho	<code>oForma.width=500</code>
Propiedad altura	<code>oForma.height=200</code>

OBJETOS

Creación de un objeto, asignándole valor, llamar método.

Command	
Llamar métodos	<code>oForma.hide()</code> <code>oForma.show()</code>
Destruir variable y objeto	<code>release oForma</code>

CLASES

Ejemplo librería.prg

```
define class miClase as custom
    function saludar()
        ? 'Hola, estoy ejecutando el metodo saludar'
    return

    function sumar(num1,num2)
        ? 'Hola voy a sumar ',num1,num2
        suma = num1+num2
        ?suma
    return suma
enddefine
```

CLASES

Comandos para ejecutar programa librería.prg

Para ejecutar una clase es necesario crear el objeto con el siguiente comando.

oObjeto1 = newObject('miClase','libreria.prg')

Llamar la función saludar.

oObjeto1.saludar()

Llamar la función sumar especificando parámetros num1, num2.

oObjeto1.sumar(30,20)

CLASES, OBJETOS Y FUNCIONES

EJEMPLO

CLASES clientesb.prg

PARTE1 CREAR CLASE

Definición de clase

```
define class clientes as custom
    id = 0
    nombre = ""
    telefono = ""
    ciudad = ""
```

Las clases deben de provenir de un objeto en este caso Custom, el cual contiene propiedades, eventos y métodos sin representación visual.

CLASES clientesb.prg

PARTE2 FUNCION CREAR TABLA ALMACENAMIENTO DE LA INFORMACION

Crear tabla clientes

```
function crearTabla()  
    create table clientes (id n(5), nombre c(50), telefono c(30), ciudad c(50))  
    select clientes  
    index on id tag id  
  
return
```

CLASES clientesb.prg

PARTE3 FUNCION ABRIR TABLA

Abrir tabla clientes

```
function abrirTabla()  
    if not file('clientes.dbf') && SI LA TABLA NO EXISTE LA CREA  
        this.crearTabla()  
        Return  
    endif  
    use clientes in 0 order id  
return
```

CLASES clientesb.prg

PARTE4 FUNCION AGREGAR CLIENTE

Agregar un cliente nuevo

```
function agregar()  
    *Obtener el ultimo id existente en la tabla  
    select max(id) as maxid from clientes into cursor temp  
    nLastId = temp.maxid + 1  
    *Agregar un registro a la tabla  
    insert into clientes (id, nombre, telefono, ciudad) values ;  
    (nLastId, this.nombre, this.telefono, this.ciudad)  
return
```

CLASES clientesb.prg

PARTE5 FUNCION ACTUALIZAR

Ejemplo clientesb.prg

```
function actualizar()  
    select clientes  
    update clientes set ;  
        nombre = this.nombre,;  
        telef   = this.telef,;  
        ciudad  = this.ciudad ;  
    where id == this.id  
return
```

CLASES clientesb.prg

PARTE6 FUNCION ELIMINAR

Eliminar un cliente

```
function eliminar()  
    delete from clientes where id==this.id  
return
```

La instrucción delete marca como desactivado el registro.

CLASES clientesb.prg

PARTE7 FUNCION CARGAR

Cargar un cliente

```
function cargar(nId)
    this.initValues()
    select * from clientes where id==nId into cursor temp
    if reccount('temp')>0
        this.id          = temp.id
        this.nombre      = temp.nombre
        this.telef = temp.telef
        this.ciudad = temp.ciudad
    endif
return
```

CLASES clientesb.prg

PARTE8 FUNCION INICIALIZACION DE VARIABLES

Inicializacion de las variables en 0

```
function initValues()  
    this.id = 0  
    this.nombre = "  
    this.telefono = "  
    this.ciudad = "  
  
return
```

CLASES clientesb.prg

PARTE9 FUNCION LISTAR TODOS

Lista los clientes de la tabla

```
function listarTodos()  
    select clientes  
    scan  
        ? clientes.id, clientes.nombre, clientes.telefono, clientes.ciudad  
    endscan  
return
```


CLASES clientesb.prg

PARTE10 FUNCION LISTAR ALGUNOS

Lista los clientes especificados en los parámetros

```
function listarAlgunos(nldIni, nldFin)
    select * from clientes where between (clientes.id, nldIni, nldFin) into cursor temp
    selec temp
    scan
        ? temp.id, temp.nombre, temp.telefono, temp.ciudad
    endscan
return
```

CAPACITACION DESARROLLADOR

INTRODUCCION A VISUAL FOXPRO6

COMANDOS

EJEMPLO	
FORMULARIO	create form compras
BASE DE DATOS	create database productos
CLASE	create class Controles
BIBLIOTECA	create classlib nue_lib
CLASE A BIBLIOTECA	create class controles of nue_bib as custom

COMANDOS

EJEMPLO	
LIMPIAR	Clear

CONTROLES FORMULARIO

EJEMPLO	
CHECKBOX	<code>MyForm.checkbox.value=.T.</code>
COMBOBOX	<code>MyForm.List1.AddItem("\-")</code>
COMMAND BUTTON	<code>frmMyForm.cmdCmndBtn1.Visible =.T.</code>
LISTBOX	
TEXTBOX	
TIMER	

CAPACITACION DESARROLLADOR

INTRODUCCION A VISUAL FOXPRO6