

## CAPITULO II

### ESTUDIO COMPARATIVO DE KYLIX Y PHP

La existencia de herramientas Kylix y PHP, que son utilizadas para el desarrollo Web en PETROECUADOR, ha conmovido saber cual de ellas presta mayor utilidad, y en que momento usarlas e implementarlas. Las herramientas tienen un mismo fin, el de construir aplicaciones Web, pero con distinta forma de funcionalidad, características y requerimientos para su desempeño.

#### 2.1 FUNCIONALIDAD POR HERRAMIENTA.

##### 2.1.1 Funcionamiento básico de Php.

El lenguaje PHP es un lenguaje de programación de estilo clásico, esto quiere decir, que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, clases y otros elementos.

##### 2.1.1.1 Sintaxis y estructura.

En un documento de php siempre empieza por una etiqueta `<?php` y después en el final cerrado por `?>` , y al finalizar cada instrucción se debe agregar el caracter `;`<sup>10</sup>

---

<sup>10</sup> <http://www.elguruprogramador.com.ar/zonas/ver.asp?cod=11> "Introducción y Sintaxis Básica del PHP" 5 de mayo del 2003.

```
<?php  
echo "Esto es un ejemplo";  
?>
```

Un documento de PHP también puede introducirse en una página html, como se puede observar.

```
<html>  
<head>  
  <title> Página HTML y PHP</title>  
</head>  
<body>  
<?  
echo "Esto es código PHP"; // Es un Script en HTML.  
?>  
</body>  
</html>
```

### 2.1.1.2 Variables.

Una variable es un contenedor de información, en el que podemos asignar números enteros, números decimales, caracteres; el contenido de las variables se puede leer y se puede cambiar durante la ejecución de una página PHP.

En PHP todas las variables comienzan con el símbolo del dólar \$ y no es necesario definir una variable antes de usarla, tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres, si se quiere forzar a que una variable tenga un tipo de datos específico se debe utilizar la función `settype()`.

```

<?php
    $a = 1;
    $b = 3.34;
    $c = "Es una prueba";
    echo $a $b $c; // 13.34Esto es una prueba
?>

```

En este ejemplo se define tres variables, \$a, \$b y \$c con diferentes tipos los cuales pueden variar durante su ejecución y dinamismo.

Los tipos de variables más principales para el manejo de documentos php son enteros, reales, alfanuméricas ver Tabla 2-1.

Tabla 2-1 (Tipo de datos de PHP)

<b>Numéricas</b> Almacenan cifras			
Enteros	\$entero=2002;	Números sin decimales	
Real	\$real=3.14159;	Números con o sin decimal	
<b>Alfanuméricas</b> Textos compuestos de números y/o cifras			
Cadenas	\$cadena="Hola número 9";		
<b>Tablas</b> Almacenan series de informaciones numéricas y/o alfanuméricas			
Arrays	Son las variables que se guardan las tablas	Unidimensional	Multidimensional
		\$vector[1]="A"; \$vector[2]=45;	\$matriz[1][2]="Pera"; \$matriz[2][“fruta”]="Piña"; \$matriz[3][“uno”][5]=5;
<b>Objetos</b>			
Se trata de conjuntos de variables y funciones asociadas. Presentan una complejidad mayor que las variables vistas hasta ahora pero su utilidad es más que interesante.			

Fuente: <http://www.desarrolloweb.com/articulos/308.php?manual=12>

Existen 2 tipos de variables, las variables *locales* que solo pueden ser usadas dentro de funciones y las variables *globales* que tienen su ámbito de uso fuera de las funciones, podemos acceder a una variable global desde una función con la instrucción `global nombre_variable`.<sup>11</sup>

### 2.1.1.3 Operadores.

Los operadores de PHP son muy parecidos a los de C y JavaScript. Estos son los operadores que se pueden aplicar a las variables y constantes, los siguientes operadores son:

#### 2.1.1.3.1 Operadores Matemáticos

Tabla 2-2 (Operadores Numéricos de PHP)

Operador	Nombre	Ejemplo	Descripción
+	Suma	5 + 6	Suma dos números
-	Resta	7 - 9	Resta dos números
*	Multiplicación	6 * 3	Multiplica dos números
/	División	4 / 8	Divide dos números
%	Módulo	7 % 2	Devuelve el resto de dividir ambos números, en este ejemplo el resultado es 1
++	Suma 1	\$a++	Suma 1 al contenido de una variable.
--	Resta 1	\$a--	Resta 1 al contenido de una variable.

Fuente: <http://www.desarrolloweb.com/articulos/310.php?manual=3>

#### 2.1.1.3.2 Operadores de Comparación.

Los operadores de comparación son usados para comparar valores y así poder tomar decisiones ver Tabla 2-3.

---

<sup>11</sup> <http://www.desarrolloweb.com/articulos/308.php?manual=12> "Variables en PHP" 5 de Mayo del 2003

Tabla 2-3 (Operadores de comparación de PHP)

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
==	Igual	\$a == \$b	\$a es igual \$b
!=	Distinto	\$a != \$b	\$a es distinto \$b
<	Menor que	\$a < \$b	\$a es menor que \$b
>	Mayor que	\$a > \$b	\$a es mayor que \$b
<=	Menor o igual	\$a <= \$b	\$a es menor o igual que \$b
>=	Mayor o igual	\$a >= \$b	\$a es mayor o igual que \$b

Fuente: <http://www.desarrolloweb.com/articulos/310.php?manual=3>

### 2.1.1.3.3 Operadores lógicos.

Tabla 2-4 (Operadores Lógicos de PHP)

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
&&	Y	(7>2) && (2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas.
and	Y	(7>2) and (2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas.
	O	(7>2)    (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera.
Or	O	(7>2) or (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera.
!	No	!(7>2)	Niega el valor de la expresión.

Fuente: <http://www.desarrolloweb.com/articulos/310.php?manual=3>

### 2.1.1.4 Sentencias.

Las sentencias más principales para el uso con variables y operadores con php son las siguientes como se muestra en la Tabla 2-5.

Tabla 2-5 (Lista de sentencias básicas de PHP)

<b>Condicionales</b>	
If..else...	If (condición){ Sentencias a ejecutar }else{ Sentencias a ejecutar }
Switch case	switch(opción) { case "opcion1": sentencias a ejecutar      break; case "opcion1": sentencias a ejecutar      break; case "opcion.n...": sentencias a ejecutar      break; }
<b>Bucles</b>	
While	while (condición) {    instrucciones a ejecutar.  }
For	for (inicial ; condición ; ejecutar en iteración) { instrucciones a ejecutar. }
<b>Conexión a base de datos</b>	
mysql_connect	mysql_connect(host,usuario,password);
odbc_connect	odbc_connect()
ibase_connect	ibase_connect(host,usuario,password);
odbc_connect	odbc_connect(Dns,usuario,password)
<b>Salidas</b>	
Printf	printf(cadena formato, variable1, variable2...);

Echo	echo (cadena y/o variables);
Print	print (cadena y/variables);
<b>Funciones</b>	
Function	function nombrefunción (parametros){ Instrucciones a ejecutar }

Fuente: <http://www.desarrolloweb.com/articulos/310.php?manual=3>

### 2.1.2 Funcionamiento de Kylix.

Kylix permite diseñar proyectos, cuyos elementos son diferentes, formularios, módulos y otros, los cuales cada uno de ellos tiene dos ficheros asociados, así por ejemplo un formulario o modulo (sus propiedades y los componentes que posea) se almacena en un fichero de formulario que por defecto es de tipo texto con extensión .xfrm. El código fuente que define el comportamiento del formulario se guarda en un unit de formulario, también de tipo texto, que tiene extensión .pas, y un proyecto tiene la capacidad de asociar a estos archivos mencionados, este proyecto, al igual es un archivo de texto con extensión .dpr asociado a dos archivos .conf y .kof, para la configuración de compilación del proyecto.<sup>12</sup>

#### 2.1.2.1 Estructura y sintaxis.

La estructura básicamente se basa en el archivo unit de formulario, que es un caso especial de un unit más general, que contiene cualquier cantidad de código reutilizable, tipos de datos, constante y variables. Resulta frecuente la modularización de una aplicación en un número (potencialmente grande) de units, algunas de las cuales serán units de formulario y otras no.

---

<sup>12</sup> LONG Brian [http://www.terra.es/personal6/ravenwizards/tutorial\\_kylix\\_blong/parte6/Tutorial\\_de\\_Kylix\\_parte6.html](http://www.terra.es/personal6/ravenwizards/tutorial_kylix_blong/parte6/Tutorial_de_Kylix_parte6.html) #Introducci%F3n "Creación de Objetos Dinámicos" 5 de mayo del 2003.

La estructura de un unit se diseña para que cumpla su misión. Exceptuando la línea del encabezado (denominada unidad de encabezado), cada unit contiene dos secciones obligatorias, el interfaz y la implementación. La sintaxis para algunas sentencias utilizan la palabra (*Begin*) y (*end;*), y al final de cada instrucción de línea esta precede del símbolo (;) a menos que sea el final del documento que termina con (*end.*).

```
Unit Unit1;  
interface  
implementation  
end.
```

Los archivos del proyecto cuentan con una cabecera que corresponde al nombre del proyecto, y los units de formulario ó units especiales, que puede utilizar dicho proyecto.

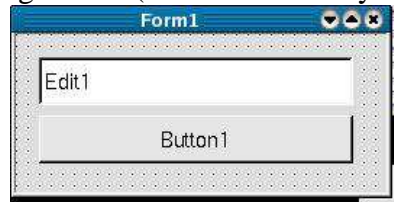
```
program Project1;  
uses  
  QForms,  
  Unit1 in 'Unit1.pas' {form1};  
{ $R *.res }  
begin  
  Application.Initialize;  
  Application.CreateForm(Tform1, form1);  
  Application.Run;  
end.
```



Ejemplo de un unit de formulario con elementos, (caja de texto y un botón), su interfase se puede apreciar en la Figura 2-1.

```
unit Unit1;  
interface  
uses  
  SysUtils, Types, Classes, Variants, QTypes, QGraphics, QControls, QForms,  
  QDialogs, QStdCtrls;  
type  
  TForm1 = class(TForm)  
    private  
      { Private declarations }  
    public  
      { Public declarations }  
  end;  
var  
  Form1: TForm1;  
implementation  
  {$R *.xfm}  
end.
```

Figura 2-1 (Formulario en Kylix)



Fuente: Investigación del Autor.

### 2.1.2.2 Procedimientos y Funciones.

Un procedimiento es una rutina del que se dispone en Pascal. Es similar a una función que no devuelve valores, ejemplo de un procedimiento:

```
procedure TForm1.Edit1Change(Sender: TObject);  
begin  
end;
```

Una función es una subrutina que se diseña para devolver un valor determinado, pudiéndose utilizar en cualquier lugar que se necesite.

Se puede ver que definir funciones (o procedimientos) es muy parecido a definir métodos. Sin embargo los métodos pertenecen a una clase y son compartidos por todos los objetos de ese tipo de clase. Las funciones y los procedimientos no son relativos a ningunas clases; simplemente son independientes de las subrutinas utilizadas.<sup>13</sup>

### 2.1.2.3 Manejadores de Eventos.

Los eventos, se definen exactamente del mismo modo que las propiedades; aunque las propiedades de un evento se definen como tipos procedimentales. Esto significa que se pueden referir a una rutina que concuerde con su tipo.

#### **Type**

```
TNotifyEvent = procedure (Sender: TObject) of object;  
property OnClick: TNotifyEvent;
```

### 2.1.2.4 Variables.

Las variables y constantes en Kylix necesariamente deben declararse ya sean estas locales o globales especificando su tipo.

Declaración de una variable tipo Integer.

```
Procedure Button1Click(Sender: TObject);  
var I: Integer; //variable integer  
begin  
I:=20;  
I:=I+20; end;
```

---

<sup>13</sup> LONG Brian: [http://www.terra.es/personal6/ravenwizards/tutorial\\_kylix\\_blong/parte6/Tutorial\\_de\\_Kylix\\_parte6.html](http://www.terra.es/personal6/ravenwizards/tutorial_kylix_blong/parte6/Tutorial_de_Kylix_parte6.html) "Creación de Objetos Dinámico" 5 de Mayodel 2003.

### 2.1.2.4.1 Tipos de Datos.

Se especificara los tipos de dato, básicos, a pesar que existen un sinnúmero de tipos que puede tomar una variable u constante Tabla 2-6

Tabla 2-6 (Tipo de datos de Kylix)

<b>INTEGER</b>		
<b>Tipo</b>	<b>Rango</b>	<b>Formato</b>
Shorint	-128...127	8-bit
Smallint	-32768...32767	16-bit
Longint	-2147483648...2147483647	32-bit
int64	$2^{63}..2^{63}-1$	64-bit
Byte	0..255	8-bit
Word	0..65535	16-bit
Longword	0..4294967295	32-bit
<b>REAL</b>		
<b>Tipo</b>	<b>Rango</b>	<b>Formato en bytes</b>
Real48	$2.9*10^{-39}.. 1.7*10^{38}$	6
Single	$1.5*10^{-45}.. 3.4*10^{38}..$	4
Double	$5.0*10^{-324}.. 1.7*10^{308}$	8
Extended	$3.6*10^{-4951}.. 1.7*10^{4932}$	10
Comp	$-2^{63}+1.. 2^{63}-1$	8
Currency	-92233720368577.5808.. - 92233720368577.5808..	8
<b>STRING</b>		
<b>Tipo</b>	<b>Máximo de caracteres</b>	<b>formato</b>
ShortString	255 caracteres	2 a 256 bytes
AnsiString	$\sim 2^{30}$	4 bytes a 2Gb
Widestrig	$\sim 2^{31}$	4 bytes a 2Gb
<b>Boleano</b>		
<b>Tipo</b>	<b>Formato</b>	
Bytbool	2 bytes	
Longbool	4 bytes	
Wordbool	8 bytes	
<b>ARRAY</b>		
<b>Unidimensional</b>	Array of [1..10] of (tipo de dato)	
<b>Multidimensional</b>	Array of [1..10] of array [1..10] of ( tipo de dato)	

Fuente: Investigación del Autor.

### 2.1.2.5 Operadores.

Los operadores utilizados en la programación de Kylix son similares a los de pascal, así los operadores para realizar las siguientes operaciones tenemos:

### 2.1.2.5.1 Matemáticos.

Tabla 2-7 (Operadores Matemáticos de Kylix)

Operador	Nombre	Ejemplo	Descripción
+	Suma	5 + 6	Suma dos números
-	Resta	7 - 9	Resta dos números
*	Multiplicación	6 * 3	Multiplifica dos números
/	División	4 / 8	Divide dos números

Fuente: Investigación del Autor

### 2.1.2.5.2 Comparación.

Tabla 2-8 (Operadores de comparación en Kylix)

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
=	Igual	a = b	a es igual b
<>	Diferente	a <>b	a es diferente b
<	Menor que	a < b	a es menor que b
>	Mayor que	a > b	a es mayor que b
<=	Menor o igual	a <= b	a es menor o igual que b
>=	Mayor o igual	a >= b	a es mayor o igual que b

Fuente: Investigación del Autor

### 2.1.2.5.3 Lógicos.

Como se observa en la Tabla 2-9 los operadores a manejarse en Kylix son:

Tabla 2-9 (Operadores lógicos de Kylix)

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
and	Y	(8>1) and (1<3)	Devuelve verdadero cuando ambas condiciones son verdaderas.
or	O	(7>1) or (1<3)	Devuelve verdadero cuando al menos una de las dos es verdadera.
not	No	not(2>1)	Niega el valor de la expresión.

Fuente: Investigación del Autor

### 2.1.2.6 Sentencias.

Las sentencias más principales para el uso con variables y operadores con Kylix son las siguientes, ver Tabla2-10.

Tabla 2-10 (Sentencias básicas de Kylix)

<b>Condicionales</b>	
If.. else	If (condición) then begin Sentencias a ejecutar; End; Else begin Sentencias a ejecutar End;
Case	Case (opción) of 0: (instrucción a ejecutarse); 1.....n: (instrucción a ejecutarse); end;
<b>Bucles</b>	
While	while (condición) do begin instrucciones a ejecutar; end;
For	for (inicial) to (numero de iteraciones) do begin instrucciones a ejecutar; end;
<b>Base de datos</b>	
Para el acceso a base de datos utiliza elementos de dbExpress(SQLconnection, SQLquery, SQLtable, SQLStoreproc, SQLDataSet, SQLclientDataSet)	
<b>Salidas</b>	
Salida de mensajes en formularios. <b>Showmessage</b>	Showmessage(cadena y/o variables);
Uso específico en aplicaciones web. request.Content	request.Content:= '(Texto HTML)';
Imprimir documentos en formas <b>Writeln</b>	Writeln(Archivo, cadena y/o variables);
<b>Funciones</b>	
<b>Funciones</b>	Function nombre(parametros):tipo de valor a devolver int string); instrucciones a ejecutar; begin

Fuente: Investigación del Autor

## 2.2 TABULACIÓN DE RECURSOS.

Los recursos que se tomarán en cuenta en la tabulación serán de hardware, para el funcionamiento de estas dos herramientas; y sus requerimientos de software de apoyo para ejecutar las aplicaciones web creadas en estas dos herramientas Kylix y PHP; sin olvidar sus propiedades básicas las cuales hacen la distinción en el mercado de herramientas para el desarrollo web.

### 2.2.1 Requerimientos para instalación de KYLIX y PHP.

Tabla 2-11 (Requerimientos para Instalar Kylix y PHP)

HARDWARE	KYLIX V 3.0		PHP V 4.3	
	Min.	Max.	Min.	Max.
Procesador	Pentium 233 MHz	Pentium IV 2.5GHz ó superior	386 66 Mhz	Pentium 200 Mhz o superior
Memoria RAM	64Mb	recomendado 256Mb	32 MB	64 MB
Disco duro	<ul style="list-style-type: none"> <li>520 MB libres en disco para instalación total.</li> </ul>		<ul style="list-style-type: none"> <li>20 MB libres en disco para instalación</li> </ul>	
Unidad de CD	<ul style="list-style-type: none"> <li>SI</li> </ul>		<ul style="list-style-type: none"> <li>SI</li> </ul>	
Teclado	<ul style="list-style-type: none"> <li>SI</li> </ul>		<ul style="list-style-type: none"> <li>SI</li> </ul>	
Mouse	<ul style="list-style-type: none"> <li>SI</li> </ul>		<ul style="list-style-type: none"> <li>SI</li> </ul>	
Monitor	<ul style="list-style-type: none"> <li>Resolución VGA o superior</li> </ul>		<ul style="list-style-type: none"> <li>Resolución VGA o superior</li> </ul>	

Fuente: Investigación del Autor.

Como se observa en la Tabla 2-11 para su instalación y funcionamiento Kylix utiliza mayor capacidad en disco y memoria RAM, para que su desempeño sea óptimo, si al trabajar con Kylix en tiempo de diseño, utiliza mucha memoria, por ende el PC debe tener memoria suficientemente; a diferencia de PHP utiliza menos espacio en disco y memoria.

### 2.2.2 Características básicas de KYLIX y PHP.

Tabla 2-12 (Características Básicas Kylix PHP)

CARACTERÍSTICAS	SOFTWARE			
	Soporta			
	KYLIX		PHP	
	Si	No	Si	No
Funcionamiento en un S.O Linux	X		X	
Entorno Visual	X			X
Acceso a bases de datos	X		X	
Compilador	X			X
Programación orientada objetos	X			X
Editor de Código	X			X
Detección de Errores	X		X	
Registro y costo de licencia	X			X
Ayuda incorporada	X			X
Aplicaciones de Servidor	X			X

Fuente: Investigación del Autor.

Como hace referencia la Tabla 2-12, el funcionamiento de estas dos herramientas puede ejecutarse en un sistema operativo Linux en sus versiones (7.0, 8.0, 9.0). La creación de aplicaciones Web en estas herramientas por lo general está orientada a un servidor Web Apache y para el funcionamiento de PHP, debe estar configurado como módulo en el servidor Web Apache, y Kylix deben ser configuradas todas las aplicaciones que se realicen como módulo o DSO´ CGI en el servidor u otro servidor Web.

El entorno visual (Figura 2-4) ayuda a Kylix a crear sus aplicaciones sin tener que utilizar otras herramientas para su desarrollo; mientras que PHP para la edición de código, puede editarse en cualquier editor de texto, a pesar que existen editores de código de PHP muy buenos, por ejemplo gPhpEdit 0.4.3 para Linux.

### 2.2.2.1 Trabajo con Bases de datos corporativas en Kylix.

El trabajo de *KYLIX* con base de datos integra en las aplicaciones que facilitan con los controladores de alta velocidad así dbExpress para Oracle, DB2, Interbase y MySQL y otras. La creación rápida y la optimización del rendimiento y la respuesta de sus aplicaciones de bases de datos con los componentes de comprobación y depuración SQL Monitor para el incremento de respuesta del servidor con las actualizaciones del caché.

La librería de componentes para desarrollo en plataformas cruzadas (CLX) proporciona una paleta mejorada de unos 165 componentes reutilizables, personalizables y ampliables para construir rápidamente aplicaciones Internet, centradas en bases de datos y el uso de interfaz gráfica (GUI).

La utilización de código fuente de los CLX permite construir los componentes personalizados o reutilizar y ampliar fácilmente los componentes base y así acelerar la velocidad de desarrollo.<sup>14</sup>

MyBase es un motor de base de datos local XML de mínima carga, que ofrece un rápido rendimiento de tablas "en memoria" e intercambio flexible de datos, y cuya instalación, configuración y distribución resultan muy simples. MyBase proporciona migración transparente para la aplicación hacia y desde cualquier sistema de gestión de bases de datos relacionales (SGBDR) soportado por Kylix. MyBase soporta relaciones de tablas maestro / detalle, los más populares tipos de datos (incluyendo Blobs), sintaxis ANSI 92 SQL para filtros, agregaciones dinámicas y restricción de datos.

---

<sup>14</sup> ABOX <http://www.abox.com/productos.asp?pid=228> "Kylix Desarrollo rápido de aplicaciones en Linux" 5 de Mayo del 2003



Kylix e InterBase, el desarrollo de alta productividad y rendimiento se combina con una base de datos rápida y de bajo mantenimiento y carga del sistema. InterBase es la base de datos de calidad comercial, que combina la facilidad de uso con un bajo coste de mantenimiento, portabilidad multiplataforma y potencia empresarial.

#### **2.2.2.2 Trabajo con Bases de datos en PHP.**

*PHP* tiene funciones que permite conectar nativamente a un sinnúmero de motores de base de datos a los que puede acceder.

- Adabas D
- dbm
- dBase
- filePro
- Hyperwave
- Informix
- InterBase
- LDAP
- Microsoft SQL server
- mSQL
- MySQL
- ODBC
- Oracle
- PostgreSQL
- Solid
- Sybase

Es posible también conectar a bases de datos a través de una interfaz ODBC, de tal forma que la conectividad con cualquier base de datos está asegurada. Las conexiones a Bases de Datos se hacen por medio de funciones contenidas en

módulos externos, pero también es posible usar módulos de abstracción a las bases de datos lo que hoy en día es un punto muy fuerte en las aplicaciones Web, además del acceso de datos proporciona funciones para el acceso a diferentes servidores como son:

- Acceso a servidores IMAP
- Envío de correo con SMTP
- Acceso a servidores de FTP
- Acceso a SNMP para gestión de redes y equipos
- Generación dinámica de gráficos y documentos PDF
- Análisis de documentos XML
- Corrector de ortografía
- Generación de datos en WDDX (Intercambio Web de Datos Distribuidos)

### **2.2.2.3 Tratamiento de imagen con PHP.**

*PHP* tiene la capacidad de generar imágenes en momento de ejecución gracias a un módulo que interactúa con las librerías GD (de Boutell). Puede crear imágenes en formato PNG, GIF y JPEG. Además de poder crear este tipo de imágenes es capaz de abrir una imagen existente, modificarla y luego enviarla al navegador.

Esta característica puede venir bien en servidores potentes y aplicaciones que requieran traducción completa, dado que normalmente los textos de los botones o imágenes de la Web no se traducen, PHP tiene la posibilidad de hacerlo.

PHP está diseñado de forma muy modular y ha sido sencillo según han ido surgiendo librerías, y utilizarlas desde PHP. Toda esta funcionalidad está basada en librerías que en su mayor parte no han sido desarrolladas.

#### 2.2.2.4 Tratamiento de imágenes en KYLIX.

Para la generación de imágenes en el Web, carece de herramientas y funciones propias en el diseño gráfico, esto hace necesario Instalar un componente TWebBitmap a la barra de herramientas de Internet. Esta herramienta permite manipular imágenes de formato BMP y generara imágenes GIF, JPEG, PNG, para implementar al desarrollo Web se puede ver en la Figura 2-2 y su instalación puede verse en el Anexo N.2.

Figura 2-2 (Componente TWebBitmap)



Fuente: Investigación del Autor.

#### 2.2.2.5 Tratamiento de cadenas con PHP.

PHP disponen de una gran variedad de funciones que trabajan con cadenas de texto. Permiten trocear una cadena en varias usando un caracter (o un conjunto de caracteres) como separador, unir varias cadenas en una, extraer subcadenas, encontrar la existencia y/o posición de un caracter (o cadena) dentro de una cadena, y una infinidad de ellas que hacen la vida más fácil al programador.

### **2.2.2.6 Trabajo con Sockets en Kylix.**

La creación de aplicaciones con sockets permite la comunicación con otros sistemas que utilizan TCP/IP u otros puertos de comunicación, existe diferentes objetos en la barra de componentes para uso en diferentes aplicaciones, divididos en dos paneles, componentes de cliente y componentes de servidor.

### **2.2.2.7 Escritura en Sockets con PHP.**

En la lectura y escritura en Sockets (UNIX y TCP/IP) se diferencian dos modos: Modo Cliente y Modo Servidor.

- **Modo Cliente**

La conexión a sockets se hace muy similar a trabajar con ficheros de texto, las funciones fgets, fputs y subcadenas son capaces de trabajar con sockets.

- **Modo Servidor**

La conexión a un socket para conseguir algo parecido a un servidor (o daemon) se hace un poco diferente al Modo Cliente, pero no deja de ser bastante más sencillo que en otros lenguajes, además dispone de fgets y fputs para leer y escribir al socket.

#### **2.2.2.8 Expresiones regulares PHP.**

Otra de las ventajas que ofrece PHP es el poder buscar y reemplazar patrones en cadenas de texto por medio de expresiones regulares, y tiene soporte tanto para expresiones regulares tipo Perl como tipo POSIX.

#### **2.2.2.9 XML en Kylix.**

Para el trabajo con archivos XML Kylix utiliza XML Data Binding wizard, que permite generar interfases, definición de clases que corresponde a la estructura de un documento XML, esto ayuda a la creación y manipulación de código XML.

#### **2.2.2.10 Trabajar con ficheros XML en PHP.**

PHP tiene dos formas de trabajar con ficheros XML, un analizador sintáctico basado en eventos (EXPAT) y un analizador sintáctico basado en DOM (libxml). Además provee una API para trabajar con ficheros WDDX.

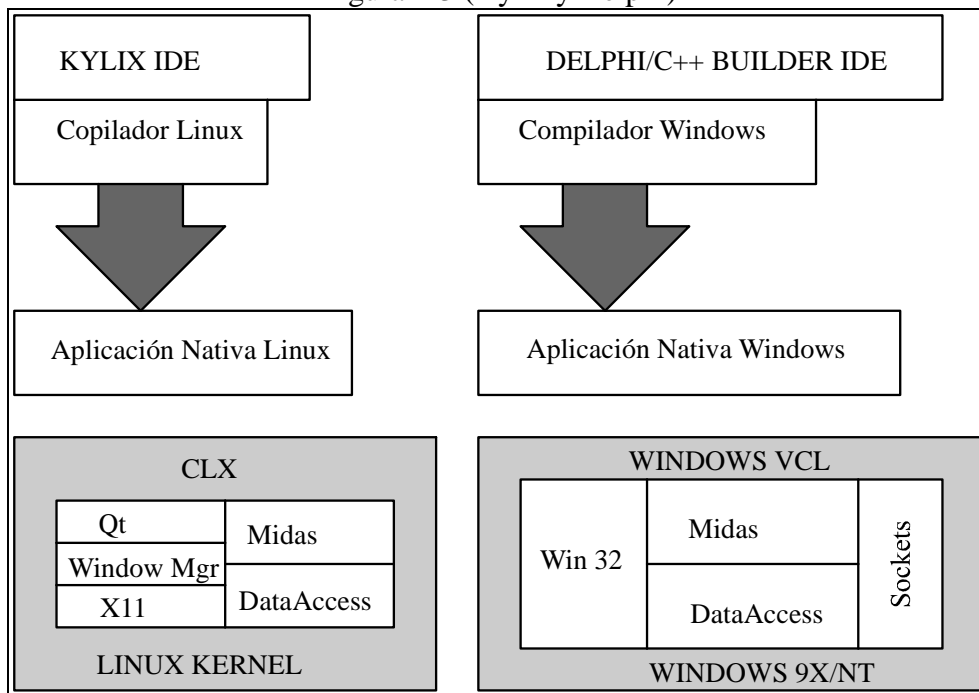
### **2.3 Compatibilidad y Portabilidad de Kylix en desarrollo Web.**

Kylix es un entorno de programación visual que soporta el Desarrollo de Aplicaciones Rápida (RAD) compatible con Borland Delphi. Está orientado a objetos y utiliza el lenguaje de programación *Object Pascal*. Es un entorno integrado (IDE).<sup>15</sup>

---

<sup>15</sup><http://www.tau.org.ar/mirrors/LuCAS/Presentaciones/200203jornadassalamanca/fcarrasco/kylix.html>  
"Borland Kylix" 20 de Mayo del 2003

Figura 2-3 (Kylix y Delphi)



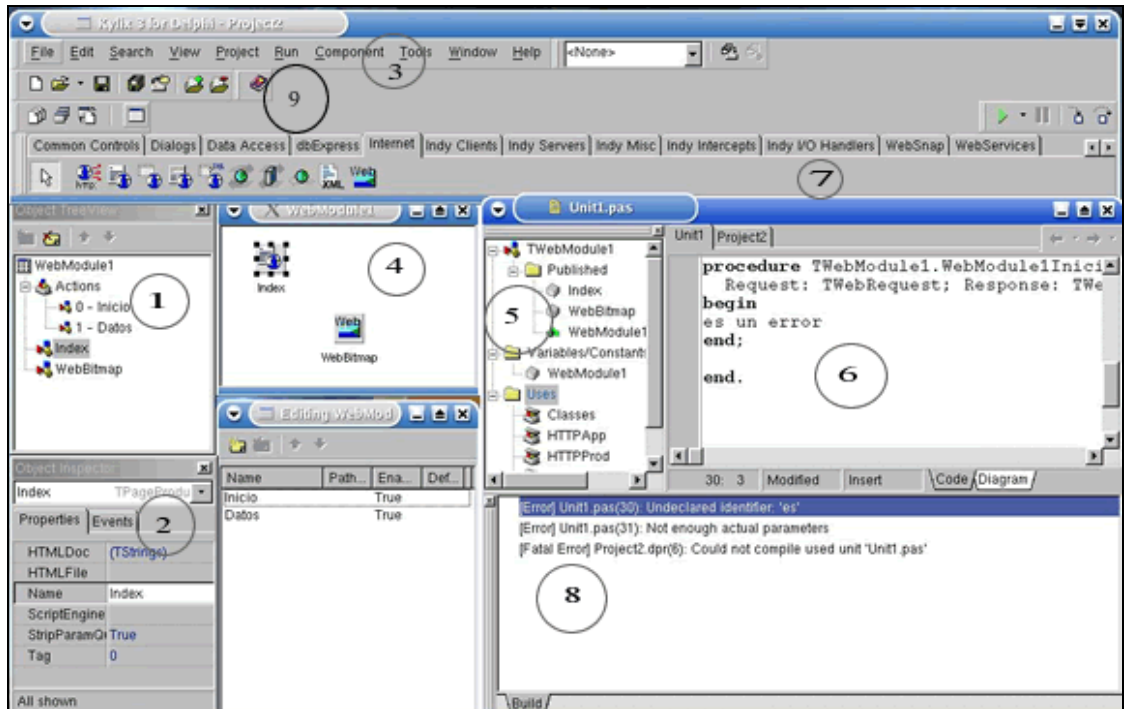
Fuente: (<http://www.evolucion-igital.net/imprimir.php?q=a&a=kylix>)

Los componentes y propiedades no están disponibles en Kylix como en Delphi Figura 2-3. Algunas propiedades desaparecen y otras nuevas están introducidas.

Además de CLX, la nueva librería de componentes de Kylix modelada y siguiendo las VCL de Delphi que permite que las aplicaciones construidas en Delphi sean portadas fácilmente a Linux con mínimos cambios, combinar Kylix con Delphi es utilizar un solo conjunto de código fuente para desarrollar aplicaciones cruzadas Linux/Windows.

El IDE de Kylix se compone de varias ventanas ver Figura 2-4.

Figura 2-4 (Entorno Kylix3)



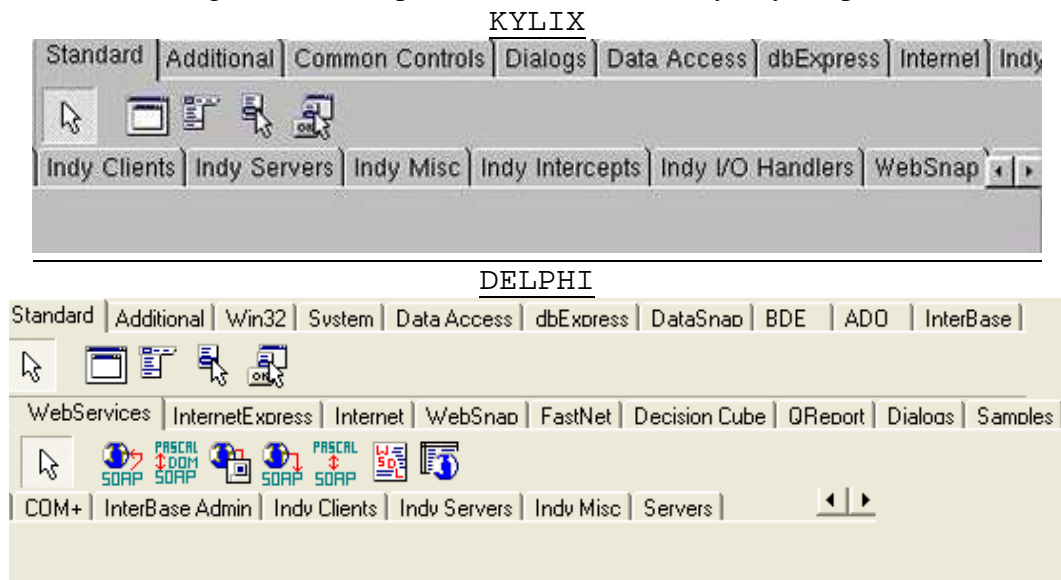
Fuente: Investigación del Autor

1. Visor de árbol de objetos (Object treeview)
2. Visor de propiedades de objetos (Object Inspector)
3. Menú
4. Modulo y/o Formulario
5. Explorador de código (Code Explorer)
6. Editor de código

7. Barra de componentes
8. Depurador de errores
9. Barra de Herramientas

Algunos componentes pueden estar presentes en Delphi como en Kylix, ver Figura 2-5.

Figura 2-5 (Componentes diferentes en Kylix y Delphi)



Fuente: Investigación del Autor

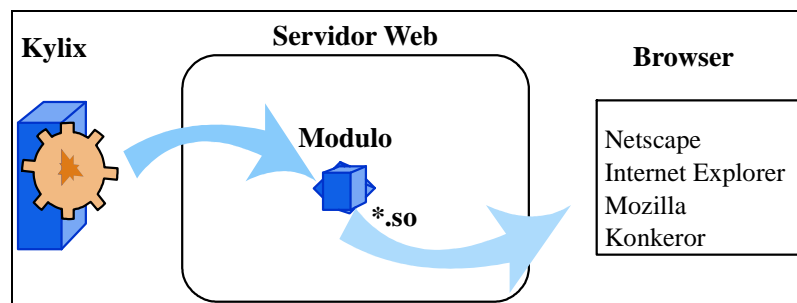
La integración de un entorno de desarrollo, hace que el depurador sea interactivo, un diseñador visual intuitivo y un conjunto de componentes mejorados que se combinan para ofrecer las herramientas que necesita para migrar las aplicaciones a Linux rápidamente. El compilador optimizado e integrado de Kylix produce aplicaciones de alto rendimiento y velocidad. La reutilización y personalización



de Kylix acelera los tiempos de producción y permite particularizar el entorno de desarrollo.<sup>16</sup>

Kylix produce directamente código ejecutable, de tal forma que las aplicaciones no dependen de otros programas o librerías, es el caso de aplicaciones Web, que son compiladas con una extensión \*.so para módulos o aplicaciones en formato CGI, ver Figura 2-6.

Figura 2-6 (Soporte de Aplicaciones Web trabajando con Kylix)



Fuente: Investigación del Autor.

## 2.4 Compatibilidad y Portabilidad de PHP en desarrollo Web.

PHP es un lenguaje de "Script" con una sintaxis híbrida entre el C y el Perl, utilizado para la generación dinámica de páginas Web, naturalmente es de programación de estilo clásico, esto quiere decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, no es un lenguaje de marcas como podría ser HTML, XML o WML. Está mas cercano a JavaScript, pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos, Mail, Impresoras. El

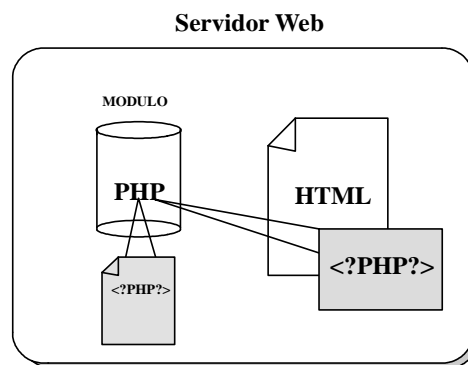
---

<sup>16</sup> ABOX <http://www.abox.com/productos.asp?pid=228> "KylixDesarrollo rápido de aplicaciones en Linux" 5 de Mayo del 2003

programa PHP es ejecutado en el servidor y el resultado enviado al navegador, el resultado es normalmente una página HTML pero igualmente podría ser una página WML, así también utiliza recursos del servidor.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador y del sistema operativo, pero sin embargo para que las páginas PHP funcionen, en el servidor Apache, IIS V3 V4, Xitami u otros debe soportar PHP, ver Figura 2-7.

Figura 2-7 (Servidor Web con soporte a PHP)



Fuente: Investigación del Autor

En cuanto a la compatibilidad y portabilidad en tiempo de desarrollo de aplicativos PHP, se pueden realizar en cualquier editor de HTML o de texto simple; el código PHP al igual que los demás Scripts se puede incrustar en la página HTML, HTM, XML; se observa en la Tabla 2-13 la compatibilidad de las dos herramientas.

Tabla 2-13 (Compatibilidad Kylix y PHP)

COMPATIBILIDAD	SOFTWARE			
	Soporta			
	KYLIX		PHP	
	Si	No	Si	No
Borland Delphi V (5, 6, 7)	X			X
Editores de Texto		X	X	
Navegadores (Mozilla, Internet Explorer, Netscape, Konqueror)	X		X	
Multipataforma en (Tiempo de Desarrollo)		X	X	
Multipataforma en (Tiempo de Ejecución)		X	X	

Fuente: Investigación del Autor.

## 2.5 Tiempo en desarrollo.

*Desarrollo Kylix para Linux*, como desarrollador Delphi, resulta fácil trabajar con Kylix. Utiliza la misma forma visual arrastrar y soltar, un entorno que es familiar a los demás lenguajes visuales, con el entorno y un mismo conjunto de herramientas, que incluyendo el Diseñador de Formularios y/o módulos, el Inspector de Objetos, el depurador y el compilador.

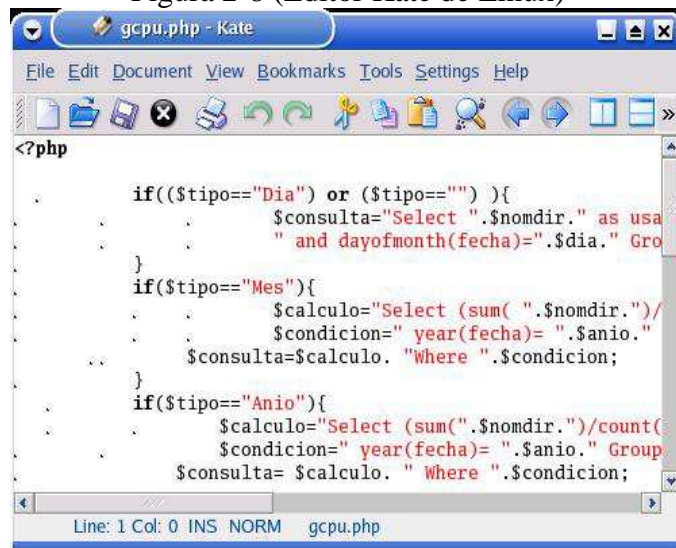
Configurar, depurar, y organizar, las mayores aplicaciones con múltiples ficheros fuentes dependientes. El Project Manager simplifica el control de contenidos, la configuración del compilador, y la construcción de cada módulo del proyecto. Manipulando fácilmente las propiedades de los componentes con el Debug Inspector, y con el Code Explorer se puede navegar por el código fuente lo que optimiza tiempo.

En *PHP* el desarrollo es poco pesado por que se tiene que editar el código en su mayoría, pero gracias a editores php se pueden simplificar el tiempo de trabajo, y

las funciones y/o clases, se pueden llamar de cualquier parte del proyecto Web en forma de requerimiento o incrustación a la página Web, esto permitiría optimizar código y tiempo.

Para la creación de código PHP se puede realizar en cualquier editor de texto u editor de código HTML, algunos editores a mencionar Vi, Kate (Figura 2-8), gPhpEdit/Linux otros, estos editores de textos ayudan a la creación de aplicaciones y proyectos PHP facilitando a los usuarios.

Figura 2-8 (Editor Kate de Linux)



Fuente: Investigación del Autor.

## 2.6 Curva de Aprendizaje.

La curva de aprendizaje de estas herramientas empieza desde su instalación y configuración, y el manejo de las mismas en diferentes niveles como son conexión a base de datos, diseño de páginas, manipulación de base de datos, diseño de páginas, control y depuración de errores. En la Tabla 2-14 se listan los niveles por semanas, que tomó en aprender y desarrollar una aplicación web.

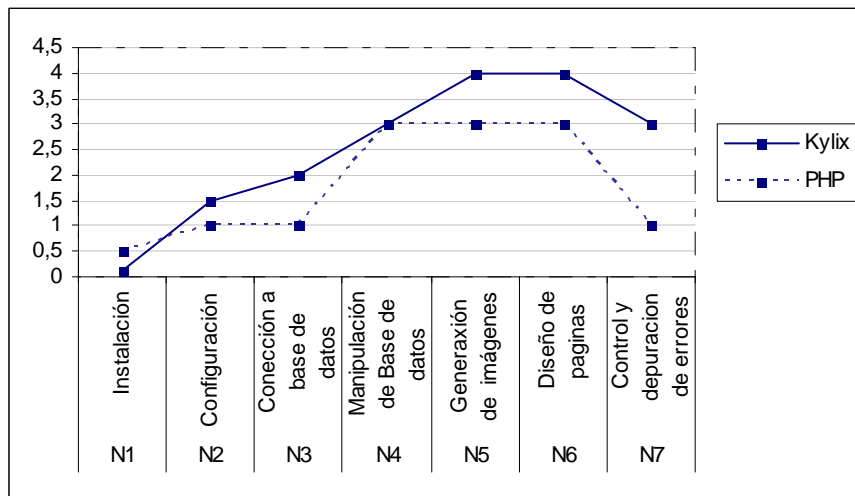
Tabla 2-14 (Niveles de aprendizaje)

Niveles	Descripción	Semanas	
		Kylix	PHP
N1	Instalación	0,1	0,5
N2	Configuración	1,5	1
N3	Conexión a base de datos	2	1
N4	Manipulación de Base de datos	3	3
N5	Generación de imágenes	4	3
N6	Diseño de paginas	4	3
N7	Control y depuración de errores	3	1

Fuente: Investigación del Autor.

Para mejor visualización del lector se observa en la siguiente Figura 2-9:

Figura 2-9 (Curva de aprendizaje)



Fuente: Investigación del Autor.

## 2.7 Característica de funcionamiento.

- Como se manifiesta anteriormente, todas las aplicaciones que se realicen en *Kylix* son compiladas de una forma nativa como modulo y/o CGI, que son alojadas a un directorio del servidor web.

Los siguientes casos son de un servidor web Apache en S.O Linux.

**Caso 1:** Alojamiento de la aplicación web como modulo de apache:

```
/opt/apache/libexec/aplicacionweb.so
```

**Caso 2:** Alojamiento de aplicación web como CGI de apache.

```
/opt/apache/cgi-bin/aplicacionwebcgi
```

En **PHP** todos sus archivos residen en la carpeta donde se ejecutan todas las páginas html, htm. El servidor carga el modulo de PHP para poder interpretar el código php y así ejecutar todas las paginas con extensión .php .php3 que se encuentran en el directorio Document Root del servidor.

## **2.8 Seguridades en Aplicaciones Web.**

En las aplicaciones Web, existen elementos o partes identificables, como: el entorno, los límites y fronteras, variables externas que interfieren o afectan las relaciones internas de los componentes del sistema, y directamente relacionados con la información existen elementos de entrada y salida, codificación, validación, procesamiento, monitoreo, control, entre otros.

Esto se ha venido convirtiendo en la columna vertebral de muchas empresas y negocios, en lo que se refiere a la seguridad, tanto para el desarrollo, ejecución y mantenimiento, de aplicaciones Web, el desarrollador debe observar, y analizar el grado de seguridad que puede implementar en su Web Site, por ejemplo en las grandes empresas que manejan transacciones, transferencia de dinero, deberá

tener un sinnúmero de controles que permitan al usuario mantener seguro sus intereses.<sup>17</sup>

Todo esto depende del software y la adecuada implementación que siempre esta buscando lograr la integración apropiada de todas las tareas relacionadas con el logro de los propios objetivos u objeto social.

En esto hay que tener en cuenta los recursos y su interacción entre sus elementos, que se pueden reunir como principales entre los siguientes cuatro componentes:

- Hardware
- Software
- Recurso Humano
- Base de Datos

Es también necesario pensar en los diferentes tipos de arquitecturas como: Cliente / servidor y actualmente con mayores tendencias hacia las aplicaciones Web, que funcionan a través de la Internet permitiendo el uso de programas desde servidores y la realización de tareas y transacciones con procesamiento en línea y en tiempo real, desde cualquier sitio, dentro y fuera de las instalaciones de las empresas.

---

<sup>17</sup> PHP <http://php.benscom.com/manual/es/security.php> "Seguridad" 5 de Mayo del 2003

## **2.8.1 Seguridades en PHP.**

PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que todo lo que se ejecute en el servidor Web sea inseguro por defecto.

PHP ha sido diseñado específicamente, para ser un lenguaje mas seguro para escribir programas CGI, que Perl o C y con la correcta selección de las opciones de configuración del tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita.

Existen diferentes modos de utilizar PHP, existen multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes usos, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

Con esto hay que actualizar las versiones que están en línea, el grupo PHP publicada versiones últimas para controlar situaciones que corrigen estos problemas de seguridad.

### **2.8.1.1 Binarios CGI.**

#### **2.8.1.1.1 Posibles ataques.**

Usando PHP como un binario CGI es una opción para instalaciones, pero por cualquier causa no se puede integrar a PHP como módulo en el software servidor por ejemplo Apache, usara PHP como clases de CGI.



Esta configuración implica generalmente al instalar el binario ejecutable de PHP en el directorio `cgi-bin` del servidor Web. El documento del CERTIFICADO recomienda no instalar intérpretes en `cgi-bin`. Aunque el binario PHP puede ser usado como intérprete independiente, PHP está diseñado para prevenir los ataques que esta configuración hace posible.

Accediendo a ficheros del sistema: `http://my.host/cgi-bin/php?/etc/passwd`

La información introducida después del signo de interrogación (?) es transferida como argumento de la línea de comando al intérprete por el interfaz del CGI. Normalmente los intérpretes abren y ejecutan el fichero especificado como el primer argumento en la línea de comando.

Cuando se ejecuta como un CGI Script, PHP rechaza interpretar los argumentos de la línea de comando.

Accediendo así a cualquier documento Web en el servidor:

`http://my.host/cgi-bin/php/secret/doc.html`

La información con el camino (Path) de la URL después del nombre del binario PHP, `/secret/doc.html` es usada convencionalmente para especificar el nombre del fichero que será abierto e interpretado por el programa CGI. Normalmente, algunas directivas del servidor Web (Apache:Action) son usadas para redireccionar peticiones de documentos como `http://my.host/secret/script.php3` al intérprete PHP. Con esta configuración, el servidor Web comprueba primero los permisos de acceso al directorio `/secret`, y después crea la petición redireccionada `http://my.host/cgi-bin/php/secret/script.php3`. Desafortunadamente, si la petición es hecha de esta forma en un principio, el servidor Web no comprueba

los permisos de acceso del fichero `/secret/script.php3`, sino solamente del fichero `/cgi-bin/php`. De esta manera cualquier usuario que pueda acceder `/cgi-bin/php` también puede acceder a cualquier documento protegido en el servidor Web.

En PHP, a la hora de compilar, la opción de configuración `--enable-force-cgi-redirect` y las directivas de configuración a la hora de ejecutar `doc_root` y `user_dir` pueden ser usadas para prevenir este ataque, si el árbol de documentos del servidor tiene cualquier directorio con acceso restringido.

#### **2.8.1.1.2 Caso 1: Solamente se sirven ficheros públicos.**

Si el servidor no contiene información que esta protegida con clave o acceso de control de IPs, no se necesitan estas opciones de configuración. Si el servidor Web no permite realizar redireccionamientos, o el servidor no tiene modo de comunicar al binario PHP que la petición es una petición segura redireccionada, se puede especificar la opción `--disable-force-cgi-redirect` en el Script de configuración. De todas maneras, se tiene que asegurar que los Scripts PHP no confíen en la manera al llamar al Script, ni de forma directa `http://my.host/cgi-bin/php/dir/script.php3` o por redirección `http://my.host/dir/script.php3`.

#### **2.8.1.1.3 Caso 2: Usando `--enable-force-cgi-redirect`.**

Esta opción a la hora de compilar previene que alguien llame a PHP directamente con una URL como la siguiente `http://my.host/cgi-bin/php/secret_dir/script.php3`. PHP solamente analizará en este modo si ha pasado por una regla de redireccionamiento en el servidor.

Normalmente la redirección en la configuración de Apache es hecha con las siguientes directivas:

Esta opción ha sido solo comprobada con el servidor Web Apache, y depende de Apache para fijar la variable de entorno CGI no estándar `REDIRECT_STATUS` en las peticiones de redireccionamiento. Si el servidor Web no soporta ningún modo para informar si una petición es directa o redireccionada, no se usa esta opción y se debe usar alguno de los otros modos de ejecución de la versión CGI documentados aquí.

#### **2.8.1.1.4 Caso 3: Usando `doc_root` o `user_dir`.**

Incluir contenidos activos, como Script y ejecutables, en el directorio de documentos del servidor Web, es algunas veces considerada una práctica insegura. Si por algún fallo de configuración, los Scripts no son ejecutados pero mostrados como documentos HTML, cualquiera podrá conseguir código registrado o información de seguridad, como por ejemplo: claves de acceso. Por ello, muchos administradores prefieren utilizar otra estructura de directorios que contenga solamente los Scripts, los cuales serán solamente accesibles vía PHP CGI, y por ello siempre serán interpretados y no mostrados.

Se puede definir el Script PHP "document root" con la directiva de configuración `doc_root` en el fichero de configuración, o definir la variable de entorno `PHP_DOCUMENT_ROOT`. Si esta definida, la versión CGI de PHP siempre obtendrá el nombre del fichero a abrir con `doc_root` y el camino (path) utilizado en la petición, así estará seguro que ningún Script sea ejecutado fuera de este directorio (excepto para `user_dir`)

Otra opción que se puede usar aquí es `user_dir`. Cuando `user_dir` no esta definido, lo único que controla la apertura del fichero es `doc_root`. Si intentamos abrir una

URL tal como esta `http://my.host/~user/doc.php3` no se abrirá un fichero en el directorio de usuarios, en su lugar se abrirá un fichero llamado `~user/doc.php3` en el directorio `doc_root`. (Si, un directorio que empieza por tilde [`~`]).

Si `user_dir` esta definido por ejemplo como `public_php`, una petición tal como `http://my.host/~user/doc.php3`, abrirá un fichero llamado `doc.php3` en el directorio llamado `public_php` del directorio "home" del usuario. Si el directorio del usuario es `/home/user`, el fichero ejecutado será `/home/user/public_php/doc.php3`.

La expansión de `user_dir` ocurre sin tener en cuenta la configuración de `doc_root`, de este modo se puede controlar los accesos al directorio principal (document root) y al directorio de usuario separadamente.

#### **2.8.1.1.5 Caso 4: Analizador PHP fuera del árbol Web.**

Otra opción más segura es poner el analizador binario PHP, en algún lugar fuera del árbol de ficheros Web. Por ejemplo en `/usr/local/bin`. Lo único de esta opción es de editar nuestro camino (path) a:

```
#!/usr/local/bin/php
```

Como primera línea en cualquier fichero que contenga código PHP. También será necesario asignar al fichero permisos de ejecución, de esta manera, es tratado de la misma manera que cualquier otro CGI Script escrito en Perl o sh u otro lenguaje utilizado para Scripts y que utilicen el mecanismo `#!` para ejecutarse.

Para conseguir que PHP maneje correctamente con esta configuración, la información de `PATH_INFO` y `PATH_TRANSLATED`, el analizador PHP debería ser compilado con la opción de configuración `--enable-discard-path`.

#### **2.8.1.1.6 Capa de seguridad en PHP.**

Otra forma de seguridad es cuando se crea archivos PHP el módulo de seguridad, incluido al principio, realiza las comprobaciones oportunas y actúa permitiendo ver el archivo o denegando su visualización dependiendo de dichas comprobaciones. Dependiendo del nivel de seguridad que deseemos implementar, la creación de este archivo puede ser más o menos complicada.

Lo único que hace, recuperar la variable de sesión donde guardo si ese usuario ha sido autenticado o no. Luego se comprueba esa variable para saber si se ha autenticado el usuario o no, realizando estas acciones.

Si no se autentifica, y redirige al navegador a la página que tiene el formulario de autenticación inicial. Además, el Script PHP, y la página deja de ejecutarse y el resto no se observará, sólo manda al navegador la redirección.

#### **2.8.2 Seguridad en Kylix.**

Kylix al utilizar componentes NetCLX, y combinan tecnologías de desarrollo, esto permite utilizar componentes de navegador, servidor y bases de datos para entregar rápidamente aplicaciones Web escalables que soporten un gran número de usuarios y un gran volumen de datos, esto mantiene la seguridad y confiabilidad de las aplicaciones Web; para realizar estos proyectos se puede utilizar funciones Internet como HTTP, FTP, SMTP y NNTP dentro de sus aplicaciones con Neurona. El nivel de seguridad también influye gracias a la utilización del compilador de código nativo altamente optimizado de Kylix para

crear aplicaciones que se ejecuten a alta velocidad. Kylix ayuda a crear dinámicamente paquetes cargados para una máxima eficacia de la aplicación y para crear librerías de objetos compartidos reutilizables y ejecutables independientes y de libre distribución.

## 2.9 Costo/Ventajas y desventajas de Kylix.

### 2.9.1 Costo de Kylix.

En la siguiente Tabla 2-15, se observa el costo de las licencias de Kylix en diferentes versiones en el mercado, esto implica a que las empresas o instituciones que quieren trabajar con esta herramienta deben pagar su licencia para uso.

Tabla 2-15 (Lista de costo Kylix 2.0 y 3.0)

Versiones de Kylix	Precios	Actualización	Licencias
Kylix Professional Edition 2.0	\$ 249.00		
Kylix Enterprise Edition 2.0	\$ 1,999.00		
Kylix Professional Edition 3.0	\$ 245.00	\$ 125.00	\$ 195.00
Kylix Enterprise Edition 3.0	\$ 1,995.00	\$ 995.00	\$ 1,595.00

Fuente: <http://www.siga.com.mx/siga/sigalista.html> "Precios"

### 2.9.2 Ventajas.

- Además de ser un desarrollador de aplicaciones Web, permite desarrollar aplicaciones convencionales usando formas en Linux.

- Compatible con Delphi, las aplicaciones CLX permiten recuperarse en otro S.O Windows.
- Su entorno es visual.
- Conexión a un sinnúmero de base de datos.
- Todas sus aplicaciones son ejecutables.
- Herramientas Wizard para aplicaciones determinadas.
- Portabilidad de aplicaciones a Delphi.

### **2.9.3 Desventajas.**

- Tiene un costo su licencia de acuerdo a su versión para utilizar en una entidad empresarial.
- Para mantenimiento de aplicaciones web, necesariamente se requiere de un equipo instalado Kylix con sus respectivas configuraciones.
- No se ejecuta en una terminal Linux.
- Utiliza mucha memoria en tiempo de diseño.
- Su nivel de aprendizaje es alto a menos que el usuario tenga conocimiento de Delphi o pascal.
- Los módulos generados, sólo se ejecuta en Linux.

## **2.10 Costos / Ventajas y desventajas de PHP.**

### **2.10.1 Costo de PHP.**

Prácticamente PHP no tiene costo, en su totalidad es gratis; y sin embargo su utilidad es poderosa para el desarrollo Web, esta herramienta por lo general trabaja con Mysql que es un motor de Base de datos, y el Servidor Apache estos y otros elementos se encuentran en la distribución Linux en cualquiera de sus versiones.

### **2.10.2 Ventajas.**

- No tiene costo, es de libre distribución.
- Existen editores específicos para PHP (gPHP Edit V1.0.9).
- El uso de variables y otros elementos en programación no necesariamente requieren ser declarados.
- Acceso nativo a base de datos.
- Sus aplicaciones son livianas en un servidor web.
- No es un Script que se ejecuta en el cliente si no en el servidor.
- El nivel de aprendizaje es rápido por ser muy similar a C++.
- Para el mantenimiento de un aplicativo web o un documento PHP, simplemente se puede utilizar una terminal de Linux.



- PHP se ejecuta en cualquier plataforma siempre y cuando lo soporte el Servidor Web.

### 2.10.3 Desventajas.

- Vulnerabilidad en sus versiones y ocasiona inseguridad.
- Posibles ataques, por ser ejecutado en modalidad CGI.
- No contiene un editor visual propio.

### 2.11 RESUMEN DE TABLAS COMPARATIVAS.

En resume podemos apreciar en la Tabla 2-16, los requerimientos, característica, compatibilidad, nivel de aprendizaje y costos.

Tabla 2-16 (Resumen de tablas comparativas)

REQUERIMIENTOS				
HARDWARE	KYLIX V 3.0		PHP V 4.3	
	Min.	Max.	Min.	Max.
Procesador	Pentium 233 Mhz.	Pentium IV 2.5Ghz ó superior	386 66 Mhz	Pentium 200 Mhz o superior
Memoria RAM	64Mb	recomendado 256Mb	32 MB	64 MB
Disco duro	<ul style="list-style-type: none"> <li>• 520 MB libres en disco para instalación total.</li> </ul>		<ul style="list-style-type: none"> <li>• 20 MB libres en disco para instalación</li> </ul>	
Unidad de CD	<ul style="list-style-type: none"> <li>• SI</li> </ul>		<ul style="list-style-type: none"> <li>• SI</li> </ul>	

Teclado	• SI	• SI		
Mouse	• SI	• SI		
Monitor	• Resolución VGA o superior	• Resolución VGA o superior		
<b>CARACTERÍSTICAS</b>				
<b>Descripción</b>	<b>SOFTWARE</b>			
	<b>Soporta</b>			
	<b>KYLIX</b>		<b>PHP</b>	
	<b>Si</b>	<b>No</b>	<b>Si</b>	<b>No</b>
Funcionamiento en un S.O Linux	X		X	
Entorno Visual	X			X
Acceso a bases de datos	X		X	
Compilador	X			X
Programación orientada objetos	X		X	
Editor de Código	X			X
Detección de Errores	X		X	
Registro y costo de licencia	X			X
Ayuda incorporada	X			X
Aplicaciones de Servidor	X			X
<b>COMPATIBILIDAD</b>				
<b>Descripción</b>	<b>SOFTWARE</b>			
	<b>Soporta</b>			
	<b>KYLIX</b>		<b>PHP</b>	
	<b>Si</b>	<b>No</b>	<b>Si</b>	<b>No</b>
Borland Delphi V (5, 6, 7)	X			X
Editores de Texto		X	X	
Navegadores (Mozilla, Internet Explorer, Netscape, Konqueror)	X		X	
Multiplataforma en (Tiempo de Desarrollo)		X	X	
Multiplataforma en (Tiempo de Ejecución)		X	X	

<b>NIVEL DE APRENDIZAJE</b>				
<b>Niveles</b>	<b>Descripción</b>	<b>Semanas</b>		
		<b>KYLIX</b>	<b>PHP</b>	
N1	Instalación	0,1	0,5	
N2	Configuración	1,5	1	
N3	Conexión a base de datos	2	1	
N4	Manipulación de Base de datos	3	3	
N5	Generación de imágenes	4	3	
N6	Diseño de paginas	4	3	
N7	Control y depuración de errores	3	1	
<b>COSTO</b>				
	<b>Versiones de Kylix</b>	<b>Precios</b>	<b>Actualización</b>	<b>Licencias</b>
<b>KYLIX</b>	Kylix Professional Edition 2.0	\$ 249.00		
	Kylix Enterprise Edition 2.0	\$ 1,999.00		
	Kylix Professional Edition 3.0	\$ 245.00	\$ 125.00	\$ 195.00
	Kylix Enterprise Edition 3.0	\$ 1,995.00	\$ 995.00	\$ 1,595.00
<b>PHP</b>	PHP V(3-4)	\$ 0.00	\$ 0.00	\$ 0.00

Fuente: Investigación del Autor.