

ANEXOS

ANEXO 1

DICCIONARIO DE DATOS

ANEXO 1 – Diccionario de Datos de Sistema LMP

Nombre de Tabla: accion

Campo	Tipo	Nulo	Key	Default	Extra
accion_id	int(10) unsigned	NO	Primaria		Auto Incremental
accion_nombre	varchar(45)	SI			

Nombre de Tabla: aplica_campana

Campo	Tipo	Nulo	Key	Default	Extra
acampana_id	int(10) unsigned	NO	Primaria		Auto Incremental
acampana_ref	varchar(21)	SI			
campana_id	int(10) unsigned	NO	Foránea		
cliente_id	int(10) unsigned	NO	Foránea		
sucursal_id	int(10) unsigned	SI	Foránea	0	
acampana_puntos	int(10) unsigned	NO			
acampana_consumo	decimal(14,2) unsigned	NO			
acampana_ticket	varchar(25)	SI			
acampana_obs	varchar(300)	SI			
user_id	int(10) unsigned	NO	Foránea		
acampana_fh_alta	Datetime	NO			
acampana_fh_mod	Datetime	SI			
acampana_estatus	varchar(10)	NO		aplicada	
cancelacion_id	int(10) unsigned	SI	Foránea		

Nombre de Tabla: aplica_redencion

Campo	Tipo	Nulo	Key	Default	Extra
aredencion_id	int(10) unsigned	NO	Primaria		Auto Incremental
aredencion_ref	varchar(21)	NO			
redencion_id	int(10) unsigned	NO	Foránea		
cliente_id	int(10) unsigned	NO	Foránea		
sucursal_id	int(10) unsigned	NO	Foránea		
aredencion_puntos	int(10) unsigned	NO			
aredencion_valor	decimal(14,2) unsigned	NO			
aredencion_tipo	int(10) unsigned	NO			
material_id	int(10) unsigned	SI	Foránea		
beneficiario_id	int(10) unsigned	SI	Foránea		
aredencion_referencia	varchar(20)	SI			
aredencion_autorizacion	varchar(20)	SI			
user_id	int(10) unsigned	NO	Foránea		
aredencion_fh_alta	Datetime	NO			
aredencion_fh_mod	Datetime	NO			
aredencion_estatus	varchar(10)	SI		aplicada	
cancelacion_id	int(11) unsigned	NO	Foránea		

Nombre de Tabla: baja_tarjeta

Campo	Tipo	Nulo	Key	Default	Extra
btarjeta_id	int(10) unsigned	NO	Primaria		Auto Incremental
cliente_id	int(10) unsigned	NO	Foránea		
cliente_bec	varchar(16)	NO			
btarjeta_motivo	int(2) unsigned zerofill	NO			
btarjeta_costo	decimal(18,2) unsigned	NO		0.00	
btarjeta_obs	varchar(300)	SI			
btarjeta_fh_mod	datetime	NO			
user_id	int(10) unsigned	NO	Foránea		
cliente_pnombre	varchar(50)	SI			
cliente_snombre	varchar(50)	SI			
cliente_apaterno	varchar(50)	SI			
cliente_amaterno	varchar(50)	SI			
btarjeta_estatus	tinyint(1)	SI		0	

Nombre de Tabla: balance_puntos

Campo	Tipo	Nulo	Key	Default	Extra
bpuntos_id	int(25) unsigned	NO	Primaria		Auto Incremental
cliente_id	int(10) unsigned	NO	Foránea		
bpuntos_generados	int(10) unsigned	NO		0	
bpuntos_fh_generados	Datetime	SI			
acampana_id	int(25) unsigned	SI	Foránea		
bpuntos_transferidos	int(10) unsigned	NO		0	
bpuntos_fh_transferidos	Datetime	SI			
transferencia_id	int(10) unsigned	SI			
bpuntos_redimidos	int(10) unsigned	NO		0	
bpuntos_fh_redimidos	Datetime	SI			
aredencion_id	int(25) unsigned	SI	Foránea		
bpuntos_cancelados	int(10)	SI		0	
bpuntos_fh_cancelacion	Datetime	SI			
cancelacion_id	int(10) unsigned	SI	Foránea		
bpuntos_saldo	int(10) unsigned	NO			
bpuntos_estatus	varchar(10)	NO		normal	

Nombre de Tabla: bandeja

Campo	Tipo	Nulo	Key	Default	Extra
bandeja_id	int(10) unsigned	NO	Primaria		Auto Incremental
bandeja_nombre	varchar(100)	NO			
bandeja_bec	varchar(16)	NO	Foránea		
bandeja_b1	varchar(4)	NO			
bandeja_b2	varchar(4)	NO			
bandeja_b3	varchar(4)	NO			
bandeja_b4	varchar(4)	NO			
bandeja_cid	varchar(15)	NO			
bandeja_fecha	varchar(5)	NO			
bandeja_msr	tinyint(1)	NO		0	

bandeja_ean	tinyint(1)	NO	0
bandeja_fh_alta	datetime	NO	
bandeja_fh_mod	datetime	NO	

Nombre de Tabla: bec_config

Campo	Tipo	Nulo	Key	Default	Extra
bconf_id	int(10) unsigned	NO	Primaria		Auto Incremental
bconf_cuenta_seed	int(10) unsigned	NO		1000	
bconf_card_seed	int(11)	NO		10000	
bconf_card_prefijo	int(6) unsigned	NO		5105	
bconf_card_consecutivo	int(10) unsigned	NO			

Nombre de Tabla: bitacora

Campo	Tipo	Nulo	Key	Default	Extra
bitacora_id	int(10) unsigned	NO	Primaria		Auto Incremental
accion_id	int(10) unsigned	SI	Foránea		
user_id	int(10) unsigned	SI	Foránea		
bitacora_ref	varchar(45)	SI		NoRef	
bitacora_fh_alta	datetime	SI			
bitacora_fh_mod	datetime	SI			
bitacora_ip	varchar(15)	SI			
bitacora_idref	varchar(45)	SI		0	
sucursal_id	int(10) unsigned	SI	Foránea	0	

Nombre de Tabla: campana

Campo	Tipo	Nulo	Key	Default	Extra
campana_id	int(10) unsigned	NO	Primaria		Auto Incremental
campana_nombre	varchar(40)	NO			
campana_descripcion	varchar(300)	NO			
campana_estatus	varchar(10)	NO		activa	
campana_fecha_inicio	date	NO			
campana_fecha_termino	date	NO			
campana_tipo	varchar(10)	NO			
tipoventa_id	int(10) unsigned	SI	Foránea	0	
campana_tipo_beneficio	varchar(10)	NO			
campana_puntos_beneficio	float unsigned	SI			
campana_porcentaje_beneficio	float unsigned	SI			
campana_consumo_minimo	decimal(16,2) unsigned	SI			
campana_consumo_maximo	decimal(16,2) unsigned	SI			
campana_frecuencia_consumo	int(10) unsigned	NO		1	
campana_periodo_consumo	int(10) unsigned	NO		1	
campana_dias_validacion	varchar(3)	SI		0	
campana_mailing	tinyint(1)	NO		0	

sucursal_id	int(10) unsigned	NO	Foránea	0
user_id	int(10) unsigned	NO		
campana_fh_alta	datetime	NO		
campana_fh_mod	datetime	SI		
tipoboleta_id	int(10) unsigned	SI	Foránea	
categoria_id	int(11)	SI	Foránea	
campana_tiplicador_puntos	int(11)	SI		1
campana_dias_activa	int(11)	SI		1
campana_tiplicador_tipo	varchar(10)	SI		ninguno

Nombre de Tabla: cancelacion

Campo	Tipo	Nulo	Key	Default	Extra
cancelacion_id	int(10) unsigned	NO	Primaria		Auto Incremental
cliente_id	int(10) unsigned	NO	Foránea		
cancelacion_ref	varchar(21)	NO			
cancelacion_operacion	int(1) unsigned	NO			
cancelacion_motivo	varchar(300)	NO			
cancelacion_fh_alta	Datetime	NO			
cancelacion_fh_mod	Datetime	NO			
acampana_id	int(10) unsigned	SI	Foránea		
aredencion_id	int(10) unsigned	SI	Foránea		
cancelacion_puntos	int(11)	NO			
cancelacion_monto	decimal(18,2) unsigned	NO			
user_id	int(10) unsigned	NO	Foránea		

Nombre de Tabla: categoria

Campo	Tipo	Nulo	Key	Default	Extra
categoria_id	int(10) unsigned	NO	Primaria		
categoria_nombre	varchar(50)	NO			
categoria_desc	varchar(50)	SI			
categoria_estatus	tinyint(1) unsigned	NO		1	
user_id	int(10) unsigned	NO	Foránea		
categoria_fh_alta	datetime	NO			
categoria_fh_mod	datetime	NO			
depocar_id	int(10) unsigned	SI	Foránea	0	

Nombre de Tabla: cliente

Campo	Tipo	Nulo	Key	Default	Extra
cliente_id	int(10) unsigned	NO	Primaria		Auto Incremental
cliente_pnombre	varchar(25)	NO			
cliente_snombre	varchar(25)	SI			
cliente_apaterno	varchar(25)	NO			

cliente_amaterno	varchar(25)	SI		
cliente_domicilio_calle	varchar(50)	SI		
cliente_domicilio_numero	varchar(25)	SI		
cliente_domicilio_cp	int(5) unsigned zerofill	SI		
cliente_domicilio_colonia	int(9) unsigned zerofill	SI		
cliente_domicilio_municipio	varchar(50)	SI		
cliente_domicilio_estado	int(2) unsigned zerofill	SI		
cliente_domicilio_pais	varchar(3)	SI		
cliente_email	varchar(100)	NO		
cliente_sexo	varchar(1)	SI		
cliente_fnacimiento	date	SI		
cliente_tel_ccode	varchar(3)	NO		52
cliente_tel_acode	varchar(3)	SI		
cliente_tel_numero	varchar(10)	SI		
cliente_identificacion_tipo	int(2) unsigned zerofill	SI		
cliente_identificacion_numero	varchar(50)	SI		
cliente_origen	varchar(20)	SI		
cliente_cuenta	decimal(14,0) unsigned zerofill	SI		
cliente_bec	varchar(16)	SI		
cliente_cid	varchar(20)	SI		
cliente_ws_referencia	varchar(25)	SI		
sucursal_id	int(11)	SI	Foránea	
grupo_id	int(10) unsigned	SI	Foránea	1
cliente_saldo_puntos	decimal(20,0) unsigned	NO		0
cliente_saldo_cuenta	decimal(20,2) unsigned	NO		0.00
cliente_avisos	tinyint(4)	NO		1
cliente_estatus	int(1) unsigned	NO		1
cliente_fh_alta	datetime	NO		
cliente_fh_mod	datetime	SI		
user_id	int(10) unsigned	NO	Foránea	
grupomail_id	varchar(5)	SI	Foránea	1
cliente_saldo_puntos_vencidos	decimal(20,0)	NO		0
cliente_preferencia_apuesta	varchar(200)	SI		
cliente_preferencia_apuesta_otro	varchar(200)	SI		
cliente_preferencia_referencia	varchar(200)	SI		
cliente_preferencia_referencia_otro	varchar(200)	SI		
cliente_preferencia_medio	varchar(200)	SI		
cliente_preferencia_medio_otro	varchar(200)	SI		
cliente_email2	varchar(100)	SI		
cliente_tel_ccode_cel	varchar(3)	SI		
cliente_tel_acode_cel	varchar(3)	SI		
cliente_tel_numero_cel	varchar(10)	SI		
cliente_domicilio_colonia_otra	varchar(100)	SI		
cliente_operadora	varchar(10)	SI		
cliente_division	varchar(10)	SI		

cliente_preferencia_mediocontacto	varchar(200)	SI
cliente_preferencia_mediocontacto_otro	varchar(200)	SI
cliente_domicilio_municipio_otro	varchar(250)	SI

Nombre de Tabla: concepto_baja

Campo	Tipo	Nulo	Key	Default	Extra
cbaja_id	int(2) unsigned zerofill	NO	Primaria		Auto Incremental
cbaja_nombre	varchar(25)	NO			
user_id	int(10) unsigned	NO	Foránea		
concepto_baja_fh_alta	datetime	NO			
concepto_baja_fh_mod	datetime	SI			

Nombre de Tabla: consultor_queries

Campo	Tipo	Nulo	Key	Default	Extra
cquery_id	int(10) unsigned	NO	Primaria		Auto Incremental
user_id	int(10) unsigned	NO	Foránea		
cquery_value	varchar(500)	NO			
cquery_fh_alta	Datetime	NO			
cquery_fh_mod	Datetime	NO			
cquery_titulo	varchar(150)	NO			

Nombre de Tabla: cpmex_colonias

Campo	Tipo	Nulo	Key	Default	Extra
codeCol	varchar(9)	NO	Primaria		
nameCol	varchar(60)	NO			
codeMpio	varchar(5)	NO	Foránea		

Nombre de Tabla: cpmex_municipios

Campo	Tipo	Nulo	Key	Default	Extra
codeMpio	varchar(5)	NO	Primaria		
nameMpio	varchar(60)	NO			
codeEdo	varchar(2)	NO	Foránea		

Nombre de Tabla: depocar

Campo	Tipo	Nulo	Key	Default	Extra
depocar_id	int(10) unsigned	NO	Primaria		
depocar_nombre	varchar(45)	NO			
depocar_descripcion	varchar(200)	SI			

ANEXO 2

CONFIGURACIÓN BÁSICA DE ZEND FRAMEWORK

ANEXO 2- CONFIGURACIÓN BÁSICA DE ZEND FRAMEWORK

Zend Framework requiere por lo menos PHP 5.1.4 o superior, aunque Zend recomienda encarecidamente la versión 5.2.3 o superior, porque hay parches de seguridad y mejoras en el rendimiento entre estas dos versiones.

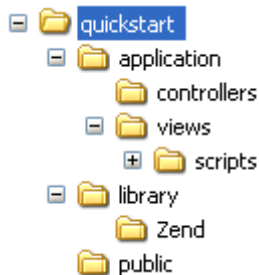
La instalación del Zend Framework es muy simple. Una vez que haya descargado y descomprimido el framework, deberá añadir la carpeta `/library` de la distribución al principio de su "include path". También puede mover la carpeta "library" a cualquier otra posición (compartida o no) de su sistema de archivos.

- Descargar la última versión estable. [<http://framework.zend.com/download>] Esta versión está disponible en formatos.zip. .tar.gz, es una buena opción para aquellos que comienzan o son nuevos en Zend Framework.

Ya que los componentes de Zend Framework están débilmente conectados, tiene la opción de usar cualquier combinación de ellos en sus aplicaciones. Los siguientes capítulos presentan una referencia exhaustiva de Zend Framework, componente a componente.

Estructura de la Aplicación en Zend Framework

Primero creamos una básica estructura MVC de nuestra aplicación bajo el directorio quickstart:



El directorio *public* deberá tener todos los archivos que serán directamente accesibles por nuestro servidor web, tales como el bootstrap (`index.php`), imágenes, css, js etc. y deberá incluir su respectivo archivo `.htaccess`.

Creando nuestro archivo .htaccess

Zend Framework MVC está diseñado implementando el patrón Front Controller. Esto significa que se debe reescribir todas las peticiones a un único script PHP que inicializa el FrontController.

Para el servidor web Apache, se debe crear el archivo `quickstart/public/.htaccess` con el siguiente contenido:

PLAIN TEXT

CODE :

```

1. RewriteEngine on
2. RewriteRule !\.(js|ico|gif|jpg|png|css)$ index.php
  
```

Esta regla de re escritura se despacharan todas las peticiones de URLs que no terminen en .js, .ico, .gif, .jpg, .png, or .css usando index.php. El archivo index.php es la puerta de entrada a nuestra aplicación, equivaldría el método main de Java o C.

Creando nuestra puerta de entrada al sistema (index.php)

Tenemos que crear nuestro quickstart/public/index.php con es siguiente contenido:

PLAIN TEXT

PHP :

```
1. error_reporting(E_ALL | E_STRICT);
2.
3. $rootPath = dirname(dirname(__FILE__));
4. set_include_path(get_include_path() . PATH_SEPARATOR .
5. $rootPath . '/library' . PATH_SEPARATOR);
6.
7. /** Incluimos Zend_Controller_Front */
8. require_once 'Zend/Controller/Front.php';
9.
10. Zend_Controller_Front::run($rootPath . '/application/controllers');
```

Primero nosotros agregamos nuestro directorio quickstart/library/ al include path de PHP. Ahora que tenemos todos los archivos de Zend Framework en el include path, podemos comenzar a hacer peticiones mediante el método run() de la clase Zend_Controller_Front y así ejecutar la Acción correspondiente (Action controllers).

Creando un Action Controller

Todas las clases controladoras de nuestra aplicación tienen que extender de la clase abstracta Zend_Controller_Action. En Zend Framework MVC el controlador (IndexController) por defecto es index y la acción (método indexAction()) por defecto también es index. Nuestro IndexController quedaría de la siguiente forma:

quickstart/application/controllers/IndexController.php

PLAIN TEXT

PHP :

```
4. /** @see Zend_Controller_Action */
5. require_once 'Zend/Controller/Action.php';
6. class IndexController extends Zend_Controller_Action
7. {
8.     public function indexAction()
9.     {
10.         $this->view->bienvenido = 'Zend Framework MVC!';
11.     }
12. }
```

Arriba en el IndexController está definido la acción indexAction(), contiene la declaración de una variable que será asignada a la vista index.phpml.

Ahora nos quedaría crear nuestra Vista que será el archivo index.phtml, la extensión de la vista (phtml), es otra más de las mejores prácticas que recomienda ZF, para

distinguir las vistas (view template) de los otros archivos PHP de nuestra aplicación, en su gran mayoría clases.

Contenido de nuestro quickstart\application\views\scripts\index\index.phtml:

PLAIN TEXT

HTML:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN";
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4.   <meta http-equiv="Content-Type" content="text/html; charset=utf-
   8" />
5.   <title>Zend Framework Quick Start</title>
6. </head>
7. <body>
8.   <h1><?php echo $this->bienvenido; ?></h1>
9. </body>
10. </html>
```

Para ver en acción nuestra aplicación abrimos el navegador y escribimos la siguiente url:
<http://localhost/quickstart/public/>

Como mencioné más arriba por defecto se despachará la petición hacia el controlador `indexController` y la acción `indexAction`, razón por la cual también podremos ver nuestra aplicación con la siguiente url: <http://localhost/quickstart/public/index> o con <http://localhost/quickstart/public/index/index>

ANEXO 3

ESTÁNDARES DE DESARROLLO

ANEXO 3- Estándares de Desarrollo

Comentarios Generales

Se podrán utilizar `/* */` para comentarios de más de una línea y `//` para comentarios de una sola línea.

Sólo se deberán insertar comentarios realmente útiles para la comprensión del programa. Evitar comentarios cuando la sentencia de código sea clara por sí misma. Como regla general, los comentarios que se insertarán serán aquellos que expliquen situaciones especiales o complejas que serían difíciles de recordar tiempo después por el mismo programador.

Clases

Los nombres de las clases deberán empezar con una letra mayúscula. Si el nombre contiene más de una palabra, entonces cada palabra deberá empezar con una letra mayúscula sin anteponer guiones bajos y otro tipo de caracteres especiales.

Ejemplos:

```
class BaseDeDatos
```

```
class Usuario
```

Funciones, Parámetros y Variables

Los nombres de las funciones y variables en general deberán estar escritos en minúsculas. Si el nombre contiene más de una palabra deberán separarse con guión bajo.

Las variables que estén relacionadas con campos de una base de datos, el nombre deberá ser exactamente igual que el del campo.

Para el resto de las variables se podrá utilizar la notación húngara, según la cual se inicia el nombre de la variable con un identificador del tipo de dato.

TIPO DE DATO	PREFIJO
Entero	int
Flotante, Real, Money	flt
Carácter	chr
Cadena	str
Booleano	bln
Fecha, Hora	dat
Arreglo	arr

En el caso de las funciones se deberá utilizar el verbo en infinitivo que denote la acción o propósito de la misma. (Ej.: calcular, obtener, graficar, generar, validar, etc)

Ejemplos:

```
$nombre_completo
```

```
$arr_mensajes
```

```
calcular_promedio()
```

Indentación

Un tabulador deberá ser utilizado como unidad de indentación en el código. Se recomienda configurar el tabulador con 4 espacios.

Nota: No se deberán utilizar los espacios, sino el tabulador.

Política de Llaves {}

Se utilizará el estilo BSD para la colocación de llaves en estructuras de control, funciones y clases. Con este estilo, las llaves son colocadas en líneas separadas, alineadas con el inicio de la palabra reservada:

Ejemplos:

```
while (condición)
```

```
{
```

```
    sentencias;
```

```
}
```

ANEXO 4

HERRAMIENTAS DE TESTEO

ANEXO 4- Herramientas de Testeo.

La realización de unos testeos de calidad, bien preparados y diseñados, es un factor clave para el óptimo desarrollo de una aplicación informática. Sin unos testeos bien detallados y planificados, una aplicación puede llegar al cliente con multitud de fallos, algo que es inaceptable en un proyecto empresarial.

Por ello, es importante identificar una serie de herramientas que ayuden a realizar esta labor tan importante. A continuación se detallarán algunas de estas herramientas, explicando las características fundamentales de cada una de ellas.

A la hora de realizar testeos, se han identificado 5 áreas importantes a tener en cuenta (se han tenido en cuenta también aplicaciones web):

1. Herramientas de carga y rendimiento (Load and Performance Test Tools)
2. Web Functional/Regression Test Tools (Java Test Tools)
3. Validadores de HTML (HTML Validators)
4. Comprobadores de Links (Link Checkers)
5. Herramientas de comprobación de seguridad (Web Site Security Test Tools)

Tipo	Nombre	Descripción
Carga y rendimiento	JMeter	Jmeter es una herramienta Open Source realizada en java (es un proyecto de Apache) que se utiliza para realizar tests de rendimiento, normalmente contra aplicaciones web. Jmeter nos permite realizar simulaciones de gran carga en el servidor, red o aplicación para comprobar su “fuerza” y para analizar el rendimiento ante diferentes tipos de sobrecarga.
Carga y rendimiento	JCrawler	Aplicación Opensource para realizar test de estrés a aplicaciones web. Le pasas una URL y puedes realizar una navegación. Admite redirecciones HTTP y cookies. Es independiente de la plataforma, posee un modo consola y es sencillo de configurar.
Funcionamiento Web	JUnit	JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado. Si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba. En caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

		JUnit también permite controlar pruebas de regresión, es decir, detectar nuevos bugs que puedan surgir a consecuencia de haber cambiado el código fuente inicial (por ejemplo, añadiendo nuevas funcionalidades). De esta manera, JUnit nos permitirá comprobar que la aplicación cumple con los requisitos y que no se ha alterado su funcionalidad inicial.
Funcionamiento Web	HttpUnit	HttpUnit es una librería java Open Source para realizar testeos de caja negra en aplicaciones web HTTP y servlets. La librería de testeo tiene las siguientes características: Autenticación Javascript Redirección automática HttpUnit se integra de forma sencilla con JUnit para realizar testeos de aplicaciones web. HttpUnit nos ofrece métodos para examinar y testear el HTML devuelto como texto, formularios, links,, botones, etc.
Validadores HTML	W3C Validator	Es uno de los validadores más conocidos en la actualidad. Es muy exigente y en ocasiones es difícil cumplirlo al 100%. Aun así, es recomendable seguir sus directrices.
Validadores HTML	HTML Validator	Plugin para firefox mediante el que podremos validar el HTML de nuestra web.
Compradores de Link	W3C Link Checker	Al igual que el “W3C Validator” para comprobar que la web cumple los estándares del W3C, W3C también nos permiten analizar una web en busca de enlaces rotos.
Compradores de Link	LinkChecker	Plugin para firefox mediante el que podremos validar si los links de una web funcionan correctamente.
Pruebas de seguridad	Powerfuzzer	Powerfuzzer es totalmente gratuito y permite a los administradores crear tests personalizados para sus aplicaciones Web con la intención de detectar agujeros de seguridad. En esencia, se trata de un simple escáner de aplicaciones. Algunos de los tests que ofrece son los siguientes: <ul style="list-style-type: none"> ∞ Código de páginas cruzadas (XSS) ∞ Inyecciones (SQL, LDAP, código, comandos, y XPATH) ∞ CRLF ∞ Estados HTTP 500
Pruebas de	Zero Day Scan	Zero Day Scan es un servicio online gratuito que

seguridad		<p>detecta las vulnerabilidades de una Web, como por ejemplo el Cross-Site Scripting (XSS), inyección SQL, y otras. A partir de este análisis, genera un resumen de resultados. Requiere la confirmación del propietario de la página.</p> <p>Características:</p> <ul style="list-style-type: none">∞ No requiere instalación, es un servicio gratuito.∞ Detecta ataques de Cross Site Scripting (XSS)∞ Detecta directorios ocultos y archives de backup.∞ Comprueba vulnerabilidades de seguridad conocidas.∞ Busca vulnerabilidades de SQL injections.∞ Detecta automáticamente zero day bugs∞ Realiza Website Fingerprinting
------------------	--	--