	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT		gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA		Versión: 1.0
		Fecha: 12/03/2010		Pág.: 1/5
DOSSIER: ¿QUÉ ES GVHIDRA?				

¿Qué es gvHIDRA?

Es el acrónimo de gvHIDRA: **g**eneralitat **v**alenciana **H**erramienta **I**ntegral de **D**esarrollo **R**ápido de **A**plicaciones.

Si hay que dar una explicación rápida de que es este proyecto, podemos decir que se trata de un entorno de trabajo (Framework) para el desarrollo de aplicaciones de gestión en entornos web con PHP siguiendo una guía de estilo (una guía para unificar los criterios de aspecto y usabilidad en el proceso de desarrollo de aplicaciones).

No sabemos si esto resolverá todas las preguntas que ha suscitado su consulta a este documento; pero entendemos, que no. Cuando tratamos de tecnología y más concretamente de Software es complicado dar una definición clara y concreta de un proyecto. Por esa razón hemos optado por intentar citar los objetivos que cubre.

Un poco de historia

El Servicio de Organización e Informática (SOI) de la CIT ha trabajado tradicionalmente con la premisa de aumentar la productividad en base a entornos de trabajo que facilitan el desarrollo de aplicaciones. Concretamente, una de sus líneas de desarrollo utilizaban unas plantillas en PowerBuilder que, en una arquitectura cliente servidor, reducían los tiempos de desarrollos resolviendo la parte general del problema.

Dentro de este marco, la CIT emprendió el proyecto **gvPONTIS** cuyo objetivo principal era migrar todos los sistemas a sistemas de código abierto. Entre otras facetas afectadas se encontraban los lenguajes de programación (se seleccionaron PHP y Java), los SGBD (se decidió fomentar el uso de Postgresql), la arquitectura (pasar al desarrollo web/tres capas)...


Con todo ello, se decidió crear un proyecto en PHP que, basándose en la guía de estilo de las aplicaciones de la CIT, aportara las mismas ventajas que las anteriores plantillas de PowerBuilder: aumentar la productiva de nuestros desarrollos.

Pero claro, a todos esos requerimientos, teníamos que añadir las dificultades que generaba el nuevo ámbito de trabajo: el entorno web (HTML, Javacript,...), el enfoque OpenSource, ... Por tanto se decidió incorporar como requerimiento la simplificación del entorno de trabajo para un desarrollador.

Con todo ello se creó un proyecto (igep: **I**mplementación de la **G**uía de **E**stilo en **P**hp) cuya primera versión estable salió el 16-11-2004. Este proyecto, al ser liberado con licencia GPL tomó la denominación gvHIDRA.

Pero, ¿Qué es gvHIDRA?

- **Patrones:** para facilitar el desarrollo se han definido una serie de patrones de diseño. Estos patrones definen la forma de representación de la información (formato tabular, registro,...) y la forma con la que interacciona el usuario con dicha información. Esto nos lleva a que con la selección de un patrón estamos marcando el diseño global de la ventana.
- **Componentes complejos:** la experiencia acumulada nos dice que todas las aplicaciones necesitan de una serie de componentes "complejos". Ventanas de selección (en PB listas de valores), listas enlazadas, acciones de interfaz (en WEB tec. AJAX),

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 12/03/2010	Pág.: 2/5
DOSSIER: ¿QUÉ ES GVHIDRA?			

mensajes de información,... El FW genera estos componentes simplificando su utilización en las aplicaciones.


- **Operaciones preprogramadas y parametrizables:** al igual que con los componentes, hemos advertido que cierta problemática se repite en todas las aplicaciones que desarrollamos. Por esa razón, la hemos generalizado y resuelto en el FW, siendo incorporada a las aplicaciones de forma transparente. Algunos de los casos más típicos: control de concurrencia, búsquedas inteligentes, CRUD, validaciones, transacciones, ...
- **Soporte a diferentes SGBD:** a través del proyecto PEAR::MDB2, el FW permite trabajar con diversos SGBD. Concretamente hemos probado Oracle, Postgresql y Mysql.
- **Listados:** uno de los mayores retos que encontramos en el ámbito del proyecto fue generar informes de forma tan versátil como los datawindows de PoweBuilder. Gracias al proyecto jasperreports lo hemos conseguido.
- **Arquitectura MVC:** el FW garantiza la arquitectura MVC forzando la separación de la lógica de negocio de la presentación mediante una distribución física de los ficheros fija.
- **Control de la vista:** el FW intenta simplificar la labor del programador, por ello no se es necesario introducir ninguna línea de código HTML o Javascript en la aplicación. Simplemente tiene que centrar sus esfuerzos en la parte de negocio (PHP). Para la parte de presentación tiene una serie de plantillas (una por patrón) en las únicamente tiene que introducir los componentes a utilizar (plugins).
- **Customs o temas:** para poder definir las particularidades propias de la organización, el FW se distribuye en una arquitectura App/Custom/core que permite modificar tanto el aspecto (CSS, imágenes, ...) como definir comportamiento propio para la organización.
- **Testing:** el FW incorpora la herramienta PHPUnit para que se puedan realizar tests sobre el código generado.
- **Autenticación:** Para poder incorporarse en diferentes organismos el FW incorpora un mecanismo de validación extensible siendo capaz de acoplarse a cualquier sistema de validación a través de PHP.
- **Depuración y Auditoría:** Tanto para facilitar la depuración durante el desarrollo como para realizar tareas de auditoría en explotación el FW incorpora un mecanismo sencillo mediante BD.

Todo esto es lo que es gvHIDRA: *generalitat valenciana Herramienta Integral de Desarrollo de Aplicaciones*.

Filosofía

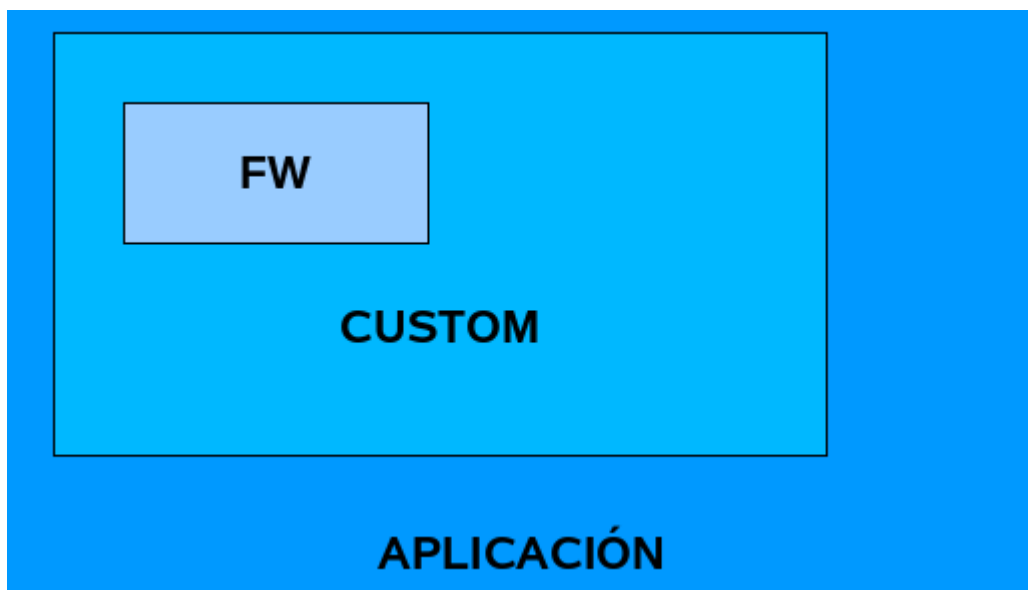
Una vez claro que es gvHIDRA, vamos a destacar un aspecto de diseño con nos revela la filosofía del proyecto. Evidentemente, se trata de un proyecto OpenSource y que aunque ha sido concebido en la CIT, ha sido diseñado para poder ser utilizado en todo tipo de organizaciones.

Esto lo podemos comprobar simplemente al observar la arquitectura de configuraciones y la

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 12/03/2010	Pág.: 3/5
DOSSIER: ¿QUÉ ES GVHIDRA?			

carga de las mismas. Para permitir la reutilización, las características de configuración se cargan en forma de herencia en el siguiente orden:

1. FW: carga las características de configuración por defecto del FW.
2. Custom: permite modificar las características de configuración para toda la organización. De este modo, no será necesario añadir dicha modificación en todas las aplicaciones.
3. App: permite modificar las características de configuración a nivel de la aplicación actual.

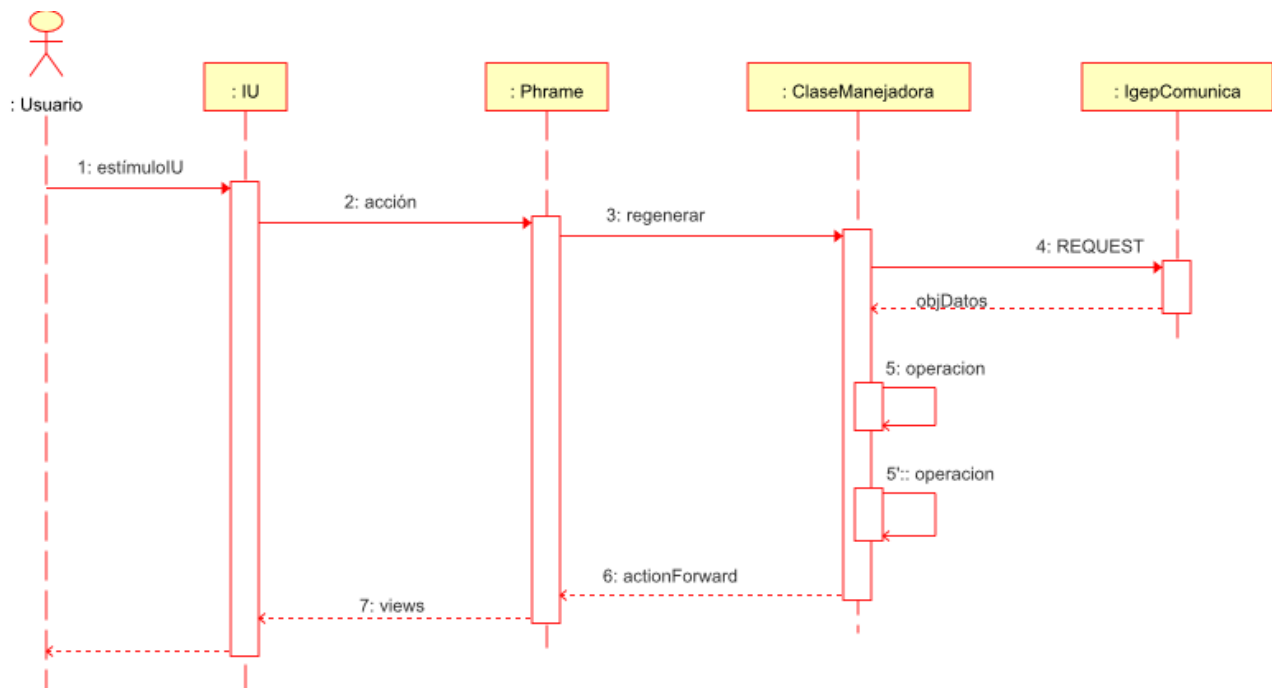


Ha esta arquitectura de carga, tenemos que añadir que el FW utiliza carga a dos niveles (estática y dinámica), de forma que permite adaptarse a requisitos evaluables en tiempo de ejecución.

Arquitectura

Para los lectores conocedores de la arquitectura web, en este apartado vamos a adentrarnos, de forma breve, en la arquitectura interna de gvHidra. Básicamente con esto pretendemos que el lector pueda entender la arquitectura del FW a través de un ejemplo. Veremos como un estímulo de pantalla atraviesa toda la arquitectura del FW y como este responde.

Teniendo en cuenta estos objetivos vamos a presentar un diagrama de secuencia donde aparecen diferentes actores (clases) que intervienen en la ejecución de una acción del framework. Este diagrama no pretende ser una referencia precisa de todos los procesos internos de gvHidra, pero será muy útil para explicar nuestros objetivos.



Seguindo la secuencia temporal marcada por la vertical de arriba hacia abajo tenemos los siguientes mensajes:


1.- el usuario lanza un estímulo que recoge la interfaz. Estos estímulos pueden ser por ejemplo:

- Pulsar un botón, bien sea un botontooltip (los de la barra superior o los de una VS) o un botón de la barra inferior.
- Cuando un campo pierde el foco.
- Cuando una lista desplegable cambia el valor.

2.- el estímulo recibido gvHIDRA lo transforma en una llamada a una acción a partir de los datos de la pantalla. Estas acciones pueden ser genéricas (tiene comportamiento preprogramado) o acciones particulares y se le encargan a una clase en concreto (clase manejadora). Phrame comprueba si la clase tiene alguna instancia en Session en cuyo caso la revive, la reconecta con la base de datos (esto garantiza la persistencia). En caso contrario crea una nueva instancia.

3.- La clase manejadora ya con el control, recibe la acción a realizar y prepara los datos.

4.- Para preparar los datos que provienen de la interfaz (ver anexo 5.2), la clase manejadora necesita que alguien los estructure y le dote de unos mecanismos adecuados para trabajar con ellos cómodamente (agrupando los datos en tuplas). Esta labor la realiza IgepComunicación, que genera los "cursores" que posteriormente se utilizarán para manejar la información. Es importante destacar que el FW evita el problema de tipos de datos en PHP ya que realiza una conversión previa (al programado los datos siempre le llegan en formato PHP).

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 12/03/2010	Pág.: 5/5
DOSSIER: ¿QUÉ ES GVHIDRA?			

5.- Las acciones se pueden componer de varias operaciones o métodos internos del framework (generalmente, dos). Estos métodos tienen puntos concretos donde el programador puede modificar su comportamiento (métodos pre y post).

6.- A partir de las posibles modificaciones del comportamiento y con los resultados de las operaciones, el Framework obtiene un `actionForward` de retorno. Tras grabar todos los datos en la `SESSION`, devuelve el control a `phrame` indicándole el destino (ver los posibles retornos de las acciones genéricas ver el anexo 5.1).

7.- `Phrame` obtiene la URL del `actionForward` y carga dicha URL. En este caso se puede dar el caso de que el programador o el framework quiera que no se recargue la página, con lo que `phrame` no cambiará la URL pero lanzará cualquier posible mensaje.

Historial de versiones

<i>Versión</i>	<i>Fecha</i>	<i>Autor</i>	<i>Comentarios</i>
1.0	12-03-2010	Toni Félix	