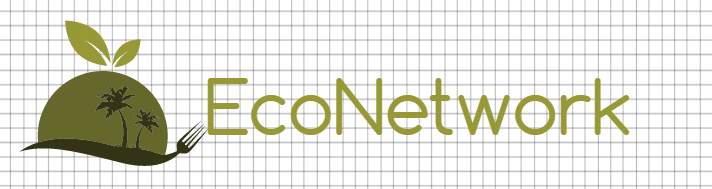
Proyecto final de grado

Introducción

**IDEA PRINCIPAL**

Desarrollo de una aplicación web para la plataforma EcoNetwork, se basará una tienda online, cuyo frontend será programado en JavaScript y PHP; mientras que su backend será implementado usando gvHIDRA, un framework de PHP de código libre desarrollado por trabajadores de la empresa donde realicé las FCT.

Presentación



EcoNetwork es una plataforma desarrollada para facilitar la integración de los distintos servicios de la compañía AjaniCorp en el mundo de las TIC. El objetivo principal de la plataforma será dotar a usuarios, empleados y cargos de la compañía de un servicio online para la gestión de la tienda de recetas ecológico-vegetarianas de la compañía. Este objetivo se logrará por medio de la creación de una aplicación web.

El objetivo de este documento es transmitir una idea general tanto del funcionamiento de la plataforma como sus detalles de análisis, diseño e implementación, tanto desde el punto de vista del usuario, como del desarrollador o la persona o equipo encargado del mantenimiento del proyecto.

La plataforma se conforma de una aplicación web que se divide en dos grandes partes. Por un lado tenemos la parte de *frontend,* que se corresponde con la parte de la aplicación interactuable por usuarios normales y clientes; por otro lado está la parte del *backend*, que se corresponde con la parte de administración y mantenimiento de la aplicación, operada por empleados o cargos de la compañía.

Valoración personal

SCRUM es una metodología maravillosa. Esto no quita que tanto yo como gran parte de mis compañeros hayamos creado una relación amor/odio con esta metodología y con las consecuencias que conllevaba su uso. En primer lugar destacar la organización del trabajo, que era impecable (lástima que solo fuera en teoría): las tareas estaban perfectamente repartidas y el trabajo era clasificado en función del coste de desarrollo que conllevaba cada tarea. Creo que aplicando SCRUM en un entorno serio y de manera óptima (sin haber ningún problema) puede potenciar al grupo de desarrollo y a la empresa a hacer entregas más rápidas y flexibles, orientando y asesorando lo máximo y mejor posible al cliente.

Ahora bien, no todo es de color de rosas en SCRUM. Como es obvio, siempre hay desarrolladores que se implican menos que otros, ya sea por desmotivación o falta de interés, o simplemente porque nos hallábamos en una “demostración” de SCRUM que aunque estaba muy bien enfocada para el grupo de clase que éramos, no llegó a concienciarnos de que nos teníamos que tomar en serio aspectos como la planificación temporal de cada tarea o el repartimiento de estas.

Por estos motivos considero SCRUM como una metodología idónea para trabajar y mantener contento al cliente, pero hay que tener en cuenta que si hay deficiencias por parte de los desarrolladores, el producto se verá ampliamente influido por estas deficiencias.

Por la parte que respecta al trabajo en grupo realizado, he de decir que no he tenido ningún problema con mi grupo a la hora de realizar el proyecto; lógicamente hay gente que ha trabajado más que otra pero aún así hemos sido capaces de realizar las entregas a tiempo y con calidad. Sin duda volvería a trabajar con mis compañeros.

Como propuesta de mejora para el próximo curso, me gustaría que se diese más hincapié en lo que a la planificación temporal se refiere, evitando el error que hemos cometido los tres grupos este año que ha sido realizar una planificación temporal arbitraria, y, por tanto, inexacta. Para llevar a cabo esto propongo acortar las entregas ligeramente y recalcar que se debe hacer bien la planificación y el reparto de tareas para que la entrega se lleve a cabo lo mejor posible.

Con respecto al proyecto realizado, me gustaría destacar el uso del framework gvHIDRA para la capa del backend de la tienda (valga la redundancia). El hecho de contar con este framework aligera mucho la carga de trabajo en lo que a mantenimientos CRUD se refiere: solo hay que preocuparse de instalar correctamente el entorno, configurar la capa de origen de datos y seleccionar las tablas que quieres mantener, asociarles una clase manejadora y parametrizar los campos a mostrar, con opciones como la visibilidad de elementos HTML, el uso de datepickers o calendarios integrados ya en el framework, soporte para listas de datos estáticas y dinámicas y la generación de reportes con Jasper.

¿Significa esto que usaría gvHIDRA constantemente como herramienta para desarrollar todos mis proyectos web? **No.**

gvHIDRA es un framework creado en 2006, por lo que muchas de sus funcionalidades aún no están actualizadas (por ejemplo, la interfaz web se ha actualizado a Bootstrap en 2015, mientras tanto usaban CSS2; es incompatible con forks de SGBD como MariaDB, etc.). Por otra parte, el lenguaje de programación con el que está hecho, PHP, no es un lenguaje de programación que me despierte mucho interés, principalmente por su incómodo sistema de dependencias, tipado débil y por la dificultad que conlleva a la hora de depurar código. Es cierto que PHP tiene características fantásticas, como su gran cantidad de librerías que incluye, su bajo coste a la hora de crear un entorno y su rapidez para construir aplicaciones, pero sus deficiencias me llevan a no querer utilizar este lenguaje, y por consecuente, este framework para mis proyectos. En su lugar me gustaría destacar una herramienta análoga a gvHIDRA, perteneciente al mismo proyecto que este, gvPontis y que está programada en Java, [gvNIX](http://www.gvnix.org/).

Tecnologías utilizadas

**Herramientas de desarrollo utilizadas**

|  |  |  |
| --- | --- | --- |
| **Tipo de tecnología** | **Tecnología utilizada** | **Descripción** |
| Entorno de Desarrollo Integrado (IDE) | NetBeans 8.1 | NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java, aunque también admite muchos otros como ha sido en este caso **PHP**. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. |
| Sistema Gestor de Bases de Datos (SGDB) | MariaDB | MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Está desarrollado por Michael (Monty) Widenius (fundador de MySQL) y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria -que reemplaza con ventajas a MyISAM- y otro llamado XtraDB -en sustitución de InnoDB. |
| Editor de texto | Sublime Text 3 (con licencia) | Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia. |
| Navegador web | Mozilla Firefox  Google Chrome  Internet Explorer | Los tres navegadores web por excelencia, sin ellos sería imposible visualizar nuestro trabajo. |
| Control de Versiones | Git | Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. |
| Servidor web | UniServer Zero XII | UniServer es una solución ligera para servidores web que corran bajo el S.O Windows. Con menos de 10 MB, incluye las últimas versiones de PHP, MySQL, Apache y Perl. |
| Automatización de tareas | Apache Ant  Jenkins | Herramientas para la automatización de todo tipo de tareas, que se sirve de scripts de configuración en formato XML en caso de Ant y Bash en caso de Jenkins.  Han sido creados con el propósito de automatizar las operaciones de despliegue. |
| Hosting | Openshift | OpenShift es un producto de computación en la nube de plataforma como servicio de Red Hat. |

Tecnologías utilizadas en el cliente

|  |  |  |
| --- | --- | --- |
| **Tipo de tecnología** | **Tecnología utilizada** | **Descripción** |
| Lenguajes de programación | JavaScript | Lenguaje de programación interpretado, que se ejecuta en el lado del cliente, específicamente en los navegadores web. Proporciona a las aplicaciones un gran dinamismo y su tipado débil facilita la curva de aprendizaje. |
| HTML 5 | Pilar central de los lenguajes de programación en el ámbito web. Es utilizado para mostrar los diferentes elementos que podemos encontrar en la web. |
| Frameworks y librerías | jQuery 1.11 | jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. |
| jqGrid | jqGrid es un componente de la librería de interfaz de usuario jQuery UI que permite mostrar datos en forma tabular y manipularlos. Incluye un paginador y una herramienta de filtrado. |
| SweetAlert | SweetAlert es una librería de JavaScript que renueva el viejo mensaje de alerta de JavaScript e incluye nuevos mensajes de alerta con opciones interesantes como prompts, animaciones y transiciones CSS. |
| Herramientas para el desarrollo de interfaces | CSS 3 | Lenguaje de programación que se encarga de dar un formato de presentación a los elementos HTML. |
| Less | Framework de CSS desarrollado en JavaScript que incluye características como variables CSS y mixins. |
| Bootstrap 3 | Framework de CSS desarrollado por Twitter, Inc que dota al diseñador de un abanico infinito de posibilidades para utilizar en sus interfaces web, así como una facilidad innata para el funcionamiento responsivo. |

Tecnologías utilizadas en el servidor

|  |  |  |
| --- | --- | --- |
| **Tipo de tecnología** | **Tecnología utilizada** | **Descripción** |
| Lenguajes de programación | PHP 5.5 | PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos |
| Librerías | cURL | cURL es una librería de PHP que permite la comunicación entre dos aplicaciones utilizando gran variedad de protocolos como HTTP, FTP, SMTP o POP3. |
| fPDF | fPDF es una librería de PHP que permite generar informes o reportes de manera similar a Jasper en Java, puediendo extraer datos de una base de datos. |

Estructura de paquetes

Al tratarse el proyecto de la tienda de una aplicación construida en PHP, no tenemos una estructura de paquetes definida y estricta, sin embargo, se ha intentado empaquetar todos los módulos de la aplicación en paquetes según su ámbito de funcionalidad. En la carpeta raíz del proyecto podemos ver los siguientes paquetes:



|  |  |
| --- | --- |
| **Nombre de paquete** | **Descripción** |
| .git | Carpeta con toda la información relativa al repositorio del SCV. |
| Ant | Carpeta con el automatizador Ant, herramienta que facilita el despliegue |
| Backend | Módulo correspondiente al backend de la aplicación. **Hecho con gvHIDRA.** |
| Config | Carpeta que contiene el archivo config.cfg, que almacena la configuración del SGDB al que debe conectarse la aplicación. |
| Css | Carpeta que contiene los archivos relativos a estilos e iconos de la aplicación. |
| Fonts | Carpeta que contiene las fuentes personalizadas de Bootstrap y de la tienda. |
| Img | Carpeta que contiene todas las imágenes utilizadas en la aplicación. |
| Jqgrid | Carpeta que contiene los archivos .php necesarios para mostrar la información en los diversos jqGrid que utiliza la aplicación. **Al usar el módulo de backend de gvHIDRA su uso es opcional, ya que el desarrollo del backend se realiza de manera independiente al contar con gvHIDRA.** |
| JS | Carpeta donde se encuentran los archivos .js que utiliza la aplicación. |
| Migration | Carpeta donde se encuentran los ficheros .sql que contienen toda la información que se cargará en el SGBD. |
| Server | Carpeta donde se encuentran todos los paquetes que se ejecutan en el lado del servidor. |

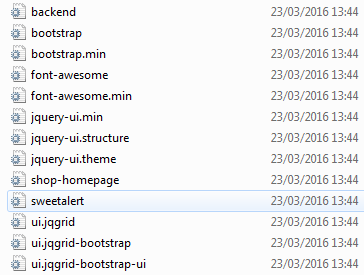
Para dejar más claro el funcionamiento de la aplicación, se hará un especial hincapié en los paquetes que la conforman, especialmente en los paquetes JS, Server y CSS, al igual que el backend realizado en gvHIDRA.

* **Paquete JS:**



El paquete JS incluye una carpeta esencial para la aplicación, la carpeta “lib” que contiene todas las librerías JavaScript que necesita la aplicación para funcionar. Además, contiene los cuatro scripts principales de la tienda, los dos del backend (no gvHIDRA), el script que contiene las clases de dominio de la tienda y el script principal de la tienda, tienda.js.

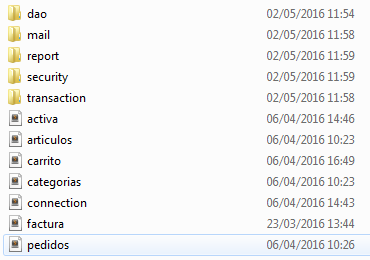
* **Paquete CSS:**

****

El paquete CSS incluye todas las hojas de estilos de la tienda. En primer lugar podemos encontrar los CSS del framework Bootstrap, el CSS del plugin de jQuery jqGrid y el CSS de la librería SweetAlert.

A parte de los archivos .css de las librerías, también podemos encontrar las hojas de estilos del backend (no gvHIDRA) y la hoja de estilo más importante, shop-homepage.css.

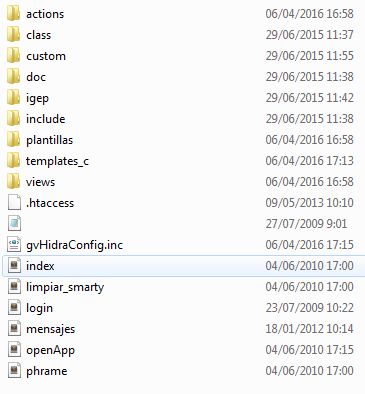
- **Paquete server:**

****

El paquete server es uno de los paquetes más importantes de la aplicación. Toda la lógica de la aplicación que corre por el lado del servidor está contenida en este paquete, que contiene los siguientes directorios:

|  |  |
| --- | --- |
| **Nombre del paquete** | **Descripción** |
| Dao | Paquete que contiene los objetos de acceso a datos, es decir, los objetos que se encargarán de acceder al SGBD y realizar las operaciones pertinentes. |
| Mail | Paquete que contiene la clase PHPMailer, que permite enviar y recibir correos electrónicos utilizando PHP. Se utiliza para proveer un servicio de correo electrónico a la aplicación. |
| Report | Paquete que contiene las clases encargadas de generar informes relativos a facturas y derivados documentos utilizados por la aplicación. |
| Security | Paquete que contiene la lógica de la aplicación relativa a la seguridad: autorización, autenticación, sesiones, acceso a cuentas y registro son algunas de sus funciones más representativas. |
| Transaction | Paquete que contiene los ficheros encargados de realizar una transacción con el banco central. |

* **Backend gvHIDRA**

****

|  |  |
| --- | --- |
| **Nombre del paquete** | **Descripción** |
| Actions | Paquete que contiene todas las clases .php que gvHIDRA, tras haberlas mapeado contra un SGBD utiliza para intercambiar información con él. **Es un análogo a la carpeta DAO del backend clásico.** |
| Custom | Paquete que contiene estilos y funcionalidades personalizados/as por el desarrollador. |
| Doc | Documentación de gvHIDRA. |
| Igep | Core o núcleo de gvHIDRA. Toda su funcionalidad está contenida en este paquete. |
| Include | Paquete donde podremos incluir módulos para que sean importados por gvHIDRA. |
| Plantillas | Paquete que contiene las plantillas que utiliza gvHIDRA, basadas en Smarty TPL. |

Requisitos iniciales

Los requisitos iniciales de la aplicación se pueden dividir en tres ámbitos principales: requisitos de interfaz, requisitos de seguridad y requisitos de organización.

* **Requisitos de interfaz:**

La interfaz está dirigida principalmente por criterios que se basan en el perfil del usuario que va a visitar potencialmente el sitio web. Es por esto, que se debe atender a este en todo momento, fijándonos en criterios que son esenciales, tales como su diseño (buena adaptación a distintos dispositivos, colores y formas dependientes de la temática, elementos web interactuables) como su manejo (usabilidad, accesibilidad, facilidad de uso). Un déficit en esta parte provocaría una disminución en la calidad del producto pues podría afectar negativamente la experiencia del usuario, ya bien dificultando su navegación o siéndole poco atrayente. Es por esto que una interfaz sólida, funcional y usable resulta crucial para el proyecto.

* **Requisitos de seguridad:**

En la seguridad relativa a una aplicación web podemos diferenciar dos procedimientos principales: autenticación y autorización. La autenticación es el proceso mediante el cual un usuario se identifica en la aplicación, y la autorización es el proceso mediante el cual se otorgan permisos de acceso a un usuario identificado.

En EcoRecipes podemos encontrar varios casos donde será necesaria la autenticación:

|  |  |  |
| --- | --- | --- |
| **Acciones** | **Rol del usuario** | **Nivel de seguridad** |
| Compra de productos | Usuario estándar  Administrador | 1 |
| Tareas de administración | Administrador | 2 |

Una vez autenticado, el usuario puede realizar las siguientes operaciones dependiendo del nivel de seguridad al que tenga acceso:

|  |  |  |
| --- | --- | --- |
| **Nivel de seguridad** | **Rol del usuario** | **Acciones permitidas** |
| 1 | Usuario estándar | * Navegación total * Compra de productos * Modificación de su información personal * Acceso a sus últimas facturas |
| 2 | Empleado EcoRecipes | * Todas las anteriores * Administración del sitio web |

* **Requisitos de organización general**

Los requisitos de organización presentan tres panoramas que atienden al rol del usuario y al tipo de uso que se quiera hacer con la aplicación. Podemos diferenciar pues tres tipos de navegación:

|  |  |  |
| --- | --- | --- |
| Tipo de navegación | Descripción | Nivel de seguridad requerido |
| Navegación normal | Navegación susceptible de ser realizada por cualquier usuario de Internet. | 0 |
| Navegación normal y compra | Al logearse se activa la función de compra, que permite adquirir productos del sitio. | 1 |
| Navegación normal, compra y administración | Nivel de navegación solamente alcanzable por empleados de EcoRecipes, permite un control total sobre el sitio. | 2 |

Puntos a destacar del proyecto

En primer lugar, me gustaría destacar el uso de gvHIDRA. gvHIDRA es una herramienta RAD, por lo que confiere al desarrollador rapidez y facilidad a la hora de implementar soluciones costosas como puede ser un CRUD; además, es un estándar que utiliza la Generalitat Valenciana para sus aplicaciones web de gestión. También destacar que el proyecto gvHIDRA es uno de los proyectos más importantes que se llevan a cabo en Alfatec Sistemas; por lo que el uso de este framework me puede servir para aprender y realizar un mejor trabajo en la empresa.

Uno de los apartados que mejor reflejan el trabajo realizado en el proyecto es el frontend, del cual me encantaría destacar la buena integración que tiene con el negocio, es decir, cumple de manera eficaz con todos los requisitos que se plantearon en el análisis y crea de una manera rotunda un símbolo de identidad de lo que al negocio respecta (el sitio web es una propia tienda de la marca EcoNetwork), y se ha conseguido esto a través de elegir los mejores colores, temática e ideario de la marca.

El último aspecto pero no menos importante que me gustaría destacar es el código fuente de la aplicación. A través del código fuente hemos podido definir unas bases y crear una estandarización que de otra manera no se podría conseguir. A pesar de ser una aplicación creada en PHP y no haber recibido pautas en cuanto a arquitectura de la aplicación, el equipo encargado del proyecto ha conseguido, mediante el uso de tecnologías estándares y prácticas de programación puestas en común crear una aplicación que satisface todos los requisitos iniciales con un código fuente homogéneo, formal y funcional.

Propuestas de mejora

Está claro que a nivel funcional debemos incluir mejoras que maximicen el rendimiento del negocio, sin embargo, a nivel de programación, me gustaría hacer dos grandes cambios: refactorización y reestructuración. El hecho de que sea una aplicación web programada en PHP tiene algunos inconvenientes en cuanto a estructura; y el hecho de que seamos estudiantes y hayamos desarrollado este proyecto en un ámbito estudiantil “en caliente”, desarrollando a medida que íbamos aprendiendo hace que al final del proyecto se encuentren errores o partes “mejor” programadas que otras.

* **Refactorización / Reestructuración.** Se incluyen las siguientes propuestas de mejora:
  + Creación de interfaces que hagan implementar a las clases los métodos (sólo para el backend clásico)
  + Uso de TypeScript en el frontend, para establecer un tipado estricto a la hora de programar. Esto proporcionará en cierta manera una estandarización al frontend, puesto que toda la aplicación compartiría la misma arquitectura.
  + Prescindir de usar la librería mysql de PHP. Está depreciada y es vulnerable a **inyección SQL.**
* **Propuestas de mejora a nivel funcional.** Se incluyen las siguientes propuestas de mejora.
  + Activación de cuenta (con captcha): esta medida de seguridad esencial en todo tipo de aplicaciones web prevendrá una creación masiva de cuentas y, por consiguiente, prevendrá la saturación de la base de datos.
  + Feedback: una tienda online sin feedback es como una multinacional sin publicidad. El captamiento de opiniones es esencial para un negocio, y más aún si es comercial.
  + Seguimiento de los pedidos: la aplicación carece de una manera de visualizar el estado del pedido, tanto en el panel de administración como para los clientes.
  + Cupones de descuento para clientes habituales: esta medida permitirá al negocio ganar clientes.