



I E S A L B A R R E G A S

TÉCNICO SUPERIOR EN DESARROLLO DE
APLICACIONES WEB

Departamento de Informática

PROYECTO

QUILLQUEST

Manual Técnico



Índice

1. Introducción	3
2. Arquitectura de la aplicación	4
2.1. Frontend	5
2.1.1. Tecnologías usadas	5
2.1.2. Entorno de desarrollo	5
2.2. Backend	6
2.2.1. Tecnologías usadas	6
2.2.2. Entorno de desarrollo	6
3. Propuesta de mejoras	7

1. Introducción

QuillQuest es una aplicación inspirada en los clásicos libros de “Crea tu propia aventura”. Estas aventuras funcionaban de forma que proporcionan un párrafo seguido de dos opciones “Si quieres hacer X, ve a la página Y. Si quieres hacer A, ve a la página B”.

De esta forma, consiste en una aplicación que brinda a los usuarios de una plataforma que les proporciona historias interactivas y la capacidad de crearlas tratando de imitar este formato de historias.



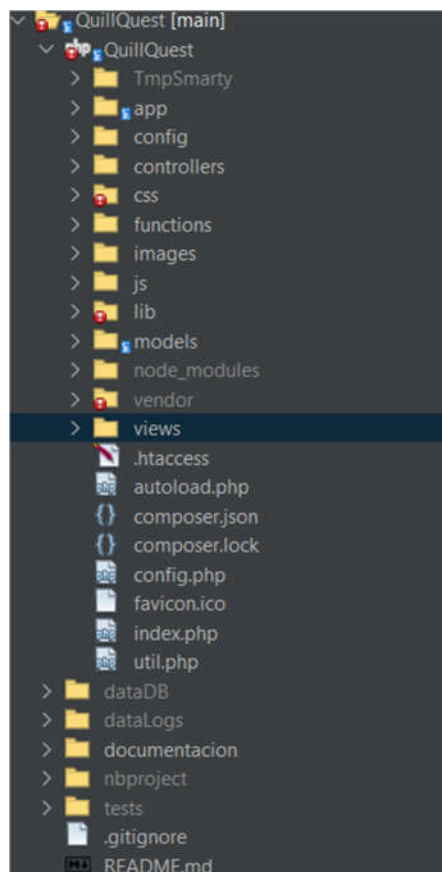
Estas historias interactivas funcionan de tal forma que el autor de una historia nos presenta un cuento con título, sinopsis, portada y género. Las páginas que componen este cuento tienen un contenido que trae la historia que se va reproduciendo, seguido de una o dos opciones que el lector deberá elegir para proceder con la historia.

Cada opción llevará a un desenlace único, y cada desenlace vendrá precedido de una opción, por lo que las historias se podrían ramificar de cientos de formas hasta llegar a un texto final sin opciones.

2. Arquitectura de la aplicación

Las tecnologías utilizadas son:

- Sistema operativo: Windows 11 Pro
- Lenguajes de programación usados:
 - HTML, JavaScript
 - PHP, SQL
- Frameworks:
 - Bootstrap
 - JQuery
 - Smarty (Plantillas para PHP)
- Entorno de desarrollo:
 - Apache Netbeans IDE 19
 - PHPMyAdmin y MySQL Workbench
 - WAMPP Bundle
- Gestor de dependencias: Composer (para PHP) <https://getcomposer.org/>



2.1. Frontend

2.1.1. Tecnologías usadas

- **HTML:**
HTML (HyperText Markup Language) es el lenguaje de marcado usado para estructurar contenido en la web mediante etiquetas.
- **JavaScript:**
JavaScript a las páginas web, ejecutándose en el navegador del usuario.
- **Bootstrap**
Bootstrap es un framework de CSS gratuito y de código abierto diseñado para facilitar el desarrollo de sitios web responsivos y móviles. Ofrece una colección de estilos y componentes predefinidos, como formularios, botones y menús de navegación.
- **jQuery**
jQuery es una biblioteca de JavaScript rápida y ligera que simplifica la manipulación del DOM, el manejo de eventos y la realización de animaciones. También facilita la realización de solicitudes AJAX.
- **Smarty (Plantillas para PHP)**
Smarty es un motor de plantillas para PHP que separa la lógica de presentación del código de aplicación. Permite a los desarrolladores y diseñadores trabajar de manera más eficiente mediante el uso de plantillas predefinidas para generar contenido dinámico.

2.1.2. Entorno de desarrollo

- **Apache NetBeans IDE 19**
Es un entorno de desarrollo integrado (IDE) gratuito y de código abierto para desarrollar aplicaciones en varios lenguajes, principalmente Java. Ofrece un editor avanzado, herramientas de depuración y pruebas, y soporte para múltiples lenguajes y tecnologías web.

2.2. Backend

2.2.1. Tecnologías usadas

- **PHP**
PHP (Hypertext Preprocessor) es un lenguaje de scripting del lado del servidor ampliamente utilizado para el desarrollo web. Permite generar contenido dinámico, interactuar con bases de datos y manejar formularios de manera eficiente.
- **SQL**
SQL (Structured Query Language) es un lenguaje estándar para gestionar y manipular bases de datos relacionales. Se utiliza para realizar consultas, actualizar datos, y administrar la estructura de las bases de datos.
- **Composer**
Es un gestor de dependencias para PHP que facilita la instalación y actualización de bibliotecas y paquetes. Permite declarar las bibliotecas de las que depende tu proyecto y se encarga de instalarlas y actualizarlas automáticamente.

2.2.2. Entorno de desarrollo

- **phpMyAdmin:**
Herramienta web gratuita y de código abierto para administrar bases de datos MySQL a través de una interfaz gráfica basada en navegador.
- **MySQL Workbench:**
Herramienta visual de diseño y administración de bases de datos MySQL, con funcionalidades para modelado de datos, desarrollo SQL y administración del servidor.

3. Proceso de despliegue

Será necesario/recomendable utilizar un Bundle de despliegue como Xampp o Wamp

htaccess

Cambiar Access-Control-Allow-Origin

Configurar entorno

1. En config.php cambiar las siguientes variables según dónde se ejecute el proyecto

_ENVIRONMENT
_URL_ENVIRONMENT
_URL_LOGS
_URL_MAIL
_KEY_CAPTCHA

Para entorno de producción habilitar 'production', para desarrollo 'develop' y para local, poner 'localhost' o cualquier que se necesite.

2. En js/config.js necesitas poner las siguientes urls

rutaAjax
principalUrl
urlLogin

3. Crear las carpetas dataDB, dataLogs y tests si no están creadas al mismo nivel que el proyecto.

4. Crear la carpeta TmpSmarty dentro del proyecto QuillQuest si no está creada.

Configurar base de datos

En primer lugar se necesita crear la base de datos e importar las tablas. Hay que crear también un usuario con privilegios.

En el directorio dataDB hay que guardar los datos de acceso a la base de datos (user, database, password y server) en formato json y después cifrado base64.

- Si tienes un entorno de producción los datos se tienen que guardar en el fichero db_prod.config.
- Si tienes un entorno local, se guardan en db_local.config.

Ejemplo json:

```
{  
  "server": "name_server",  
  "user": "name_user",  
  "password": "password_user",  
  "name": "name_db"  
}
```

Después tienes que codificar el json en base64 y ponerlo en el fichero.

Captcha key secret

Si se utiliza la función de captcha de google aquí se almacena la key secret sin condicionar. Cambiar por la que corresponda.

Configurar servidor de correo

Se necesita configurar el servidor de correo con los datos propios. La clase se encuentra en config/ImMailer.php

Logs

Necesitas crear el directorio dataLogs en la raíz, mismo nivel que dataDB. También hay que crear los subdirectorios que se necesiten.

dataLogs/

- panel
- web
- mails

Composer

Ejecutar composer para que se instalen todas las librerías.

Eliminar la carpeta de phpunits en producción -> muy importante

Privilegios

En el caso de linux

```
# chown apache:apache -R "directory"
# chcon -t httpd_sys_content_t "directory" -R
# chcon -t httpd_sys_rw_content_t "directory" -R
```

Se necesitan dar privilegios a estos directorios:

- dataLogs/
- QuillQuest/TmpSmarty
- QuillQuest/images/
- tests/

4. Propuestas de mejoras

Hay varias propuestas de mejoras presentes:

- La primera y la más urgente es la edición de las páginas de las historias, tanto a la hora de crear una historia nueva como a la hora de editar una historia con páginas ya existentes
- Las lecturas de las historias pueden quedar más elegantes e incluir la portada
- Al finalizar una historia, sería interesante llevar a una valoración de la historia y una sección de comentarios
- Los autores no tiene mucho crédito en la página más allá de la mención en la presentación de la historia. Merece la pena hacer ranking de usuarios y darles mayor visibilidad a aquellos que mejores valoraciones reciban
- Realizar un balance en el panel de control. Los usuarios podrían estar interesados en ver los resultados de sus historias.