

Barcelona Web

Barcelona's Web Service for tourism

Class: Desarrollo de Aplicaciones Web 2 (DAW2)

Name: Edgar Muñoz Manjón

Thesis Advisors: Carlos Arrebola, Jordi Egea, Anna Rovira

Due Date: 13/05/2023

El proyecto que voy a realizar es una página web sobre Barcelona. Esta web es una SSR(Server Site Rendering), por lo cual, todas las vistas (páginas) se ejecutan en el servidor, aunque el comportamiento será de una SPA (Single Page Application), por lo tanto, se verá como una única página. **NO** se ha necesitado contratar un servidor externo (Ejemplo: 000WebHost), debido a que **me he montado mi propio servidor/framework**. Está realizada de la siguiente forma:

- **BackEnd:** el lenguaje de programación que utiliza es Java.
- **FrontEnd:** son las páginas web. Estas páginas estarán creadas mediante Web Components (inyectar código en la web), debido a que es más sencillo organizar cada código.
- **Base de Datos:** la base de datos será MySQL. Esta base de datos, contendrá las siguientes tablas:
 - ◆ Tabla Usuarios:
 - Identificador (id) del usuario -> integer
 - Nombre -> String
 - Apellido -> String
 - Edad -> DATE
 - Sexo -> Char
 - Dirección -> String
 - Teléfono -> String
 - Contraseña (cifrada) -> String
 - ◆ Roles:
 - Identificador (id) de los roles -> integer
 - Nombre del rol -> String
 - Tipo de rol -> integer

Hay que tener en cuenta este sistema de validación por parte del cliente:

- El **correo** debe de ser correcto, por lo tanto, debe cumplir esta expresión regular: `/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,4}+$/`
- El **nombre y el apellido** del usuario, la **inicial** debe de ser en mayúscula y el resto en minúscula. **Mínimo 3 letras y máximo 20 letras. No tiene que haber espacios. No puede haber ningún carácter.**
- El apartado **sexo**, tiene que poner **si es hombre, mujer...**
- La **edad** debe de ser una fecha
- El **teléfono** debe de tener entre 3 y 9 números
- La **dirección** debe de **ser correcta**

Por parte del servidor, hay que cumplir **TODOS** los tests para comprobar que todos los datos sean válidos, y evitar los daños a la Base de Datos y al programa general.

Otra de las cosas que hay que tener en cuenta, hay que evitar la inyección SQL, puesto que puede perjudicar al sistema de BackEnd y a la propia Base de Datos.

Casos de Uso:

- **Administrador:** solo puede haber un **único** administrador, que es el que se encarga de gestionar **TODO** el programa, incluyendo la Base de Datos.
- **Registrado:** este rol **NO** tiene permisos para acceder a la Base de Datos, tampoco puede modificar los datos de la Base de Datos, ni el programa (BackEnd) general.
!!!Para actualizar los datos, previamente se debe haber completado el formulario de Registro!!!
- **NO Registrado:** este rol es el que **viene por defecto**, ya que NO estas registrado. La funcionalidad que puedes conseguir es navegar por la página web, pero **NO** puedes acceder a algunas secciones, puesto que **tienes que registrarte** para obtener los permisos de dicho rol.

Diagrama de clases:

El diagrama de clases es la manera de representar la funcionalidad del proyecto, ya que te indica la vinculación de los datos. Este proyecto está organizado de la siguiente manera y su funcionalidad:

- **MODELO:**
 - **Configuración:** esta clase almacena la configuración de la Base de Datos.
- **VISTA:** En este apartado, debo de explicar cómo se relaciona los componentes (FrontEnd) con el BackEnd.
- **CONTROLADOR:**
 - **AES:** esta clase tiene como funcionalidad encriptar la contraseña y almacenarla en la Base de Datos. Estas son sus propiedades y métodos:
 - Propiedades:
 - ◆ **Algorithm->String:** definimos que tipo de algoritmo es (AES, Blowfish...)
 - Métodos:
 - ◆ **encryptPassword(password String):** este método recibe la contraseña del usuario introducido en el formulario y lo que hace es generar el cifrado,

Las css que voy a implementar van a ser las siguientes:

- h1-> font-family: 'Quattrocento', serif;
- label-> font-family: 'Rokkitt', serif; font-size: xx-large;

La web va a tener estas características:

- Los componentes cuya extensión es **.global**, significa que son componentes **estáticos**, es decir, **todas** las páginas (**vistas**) tendrán el mismo aspecto.
- Los componentes cuya extensión es **.web**, son los componentes de esa vista, por lo cual, es **independiente** del resto de páginas.
Cada componente tendrá su css y su lógica de negocio (JavaScript), aunque el aspecto puede que no se note.

Recursos Web

Tipografías de Google Fonts: [Browse Fonts - Google Fonts](#)

Búsqueda de información:

1. Métodos de referencia para obtener ruta del fichero: <https://puntocomnoesunlenguaje.blogspot.com/2013/05/clase-file-java.html>
2. Evitar la inyección SQL: [Evitar SQL Injection con Java \(openai.com\)](#)
3. Para crear mi propio servidor HTTP: [Java HTTP Web Server \(openai.com\)](#)
4. Encriptación de Contraseña: [Encrypting Passwords in MySQL with Java \(openai.com\)](#)
5. Rest Client: [Create Java REST Client \(openai.com\)](#)

Herramientas Utilizadas

La herramienta **ChatGPT** de **OpenAI** me ha servido para obtener la idea de gestionar el programa y saber como organizar el cliente, servidor y base de datos.