

MeteOwn

CFGS DAW

CURSO 2021/2022

Pablo Cruz Rodríguez

COLEGIO CALASANZ SALAMANCA



| | |
|---|-----------|
| 1. Introducción | 3 |
| 2. Descripción de la aplicación | 4 |
| 3. Tecnologías escogidas y justificación | 18 |
| 4. Diseño de la aplicación | 20 |
| 4.1. Diagramas y definición de casos de uso..... | 20 |
| 4.2. Diagramas de clases..... | 21 |
| 4.3. Modelo entidad-relación | 31 |
| 5. Arquitectura de la aplicación | 32 |
| 5.1. Estructura del proyecto | 32 |
| 5.2. Librerías externas utilizadas..... | 40 |
| 6. Manual de despliegue..... | 41 |



1. Introducción

El proyecto “MeteOwn” consiste en una aplicación meteorológica con elementos personalizables. En ella podrás consultar el tiempo en tu ciudad, con una previsión de un par de días y podrás acceder a un mapa global con distintos gráficos climatológicos. Además, cada usuario dispone de un avatar totalmente personalizable, al que podrás vestir con diferentes prendas, cada una de ellas cambiando la previsión meteorológica a una más precisa para dicho usuario con dichas prendas. Esto es, si eliges un anorak, la web te informará de que en un día de invierno estarás más abrigado que si eliges una sudadera. Todo personalizable y actualizable, permitiendo comprobar rápidamente con los datos climatológicos del mismo día que combinación de prendas necesitas exactamente, para no abrigarte ni más ni menos.

2. Descripción de la aplicación

La aplicación web de MeteOwn cuenta con distintas páginas, entre las que destacaremos las más importantes, “Home”, “Modify Clothes”, “Weather Me” y “Sign Up”. A lo largo del documento iremos exponiendo y describiendo en detalle cada una de ellas, el prototipado original y el resultado final.

Home (Prototipo).

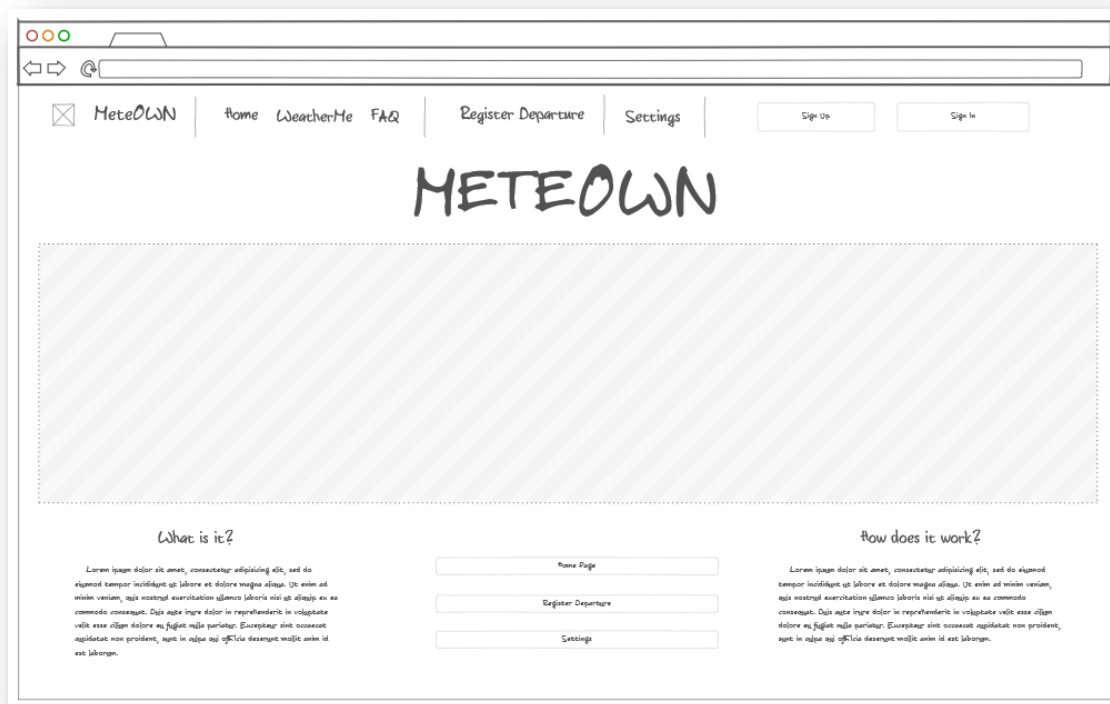


Imagen de un prototipado de la imagen de la página de Home.



En ella se puede apreciar un banner superior que permanecerá siempre a la vista y que contendrá, empezando por la izquierda:

- Logo.
- Título, con acceso a la página de Home.
- Inicio, con acceso a la página de Home.
- Weather Me, con acceso a la página principal del usuario.
(Si está autenticado, sino saltará el PopUp de login).
- FAQ, con acceso a una página simple con respuestas a preguntas sencillas sobre el funcionamiento de la web.

(En caso de estar autenticado)

- Register Departure, con acceso a una página desde la que se puede modificar de manera manual la sensación térmica de distintas prendas y distintos conjuntos de prendas.
- Setting, con acceso a un menú con distintas opciones para cambiar ajustes de usuario.

(En caso de no estar autenticado)

- Sign up, redirigirá a la página de registro.
- Sign in, aparecerá el PopUp de login.

Debajo del banner, en la zona central, aparecerá el nombre de la aplicación y a continuación, ocupando todo el ancho, un carrusel con imágenes con desplazamiento automático y manual.



Debajo, dividido en tres zonas tenemos:

- Descripción de la página web.
- Botones con acceso a las partes más importantes de la web.
- Descripción del funcionamiento de la web.

Home (Resultado).

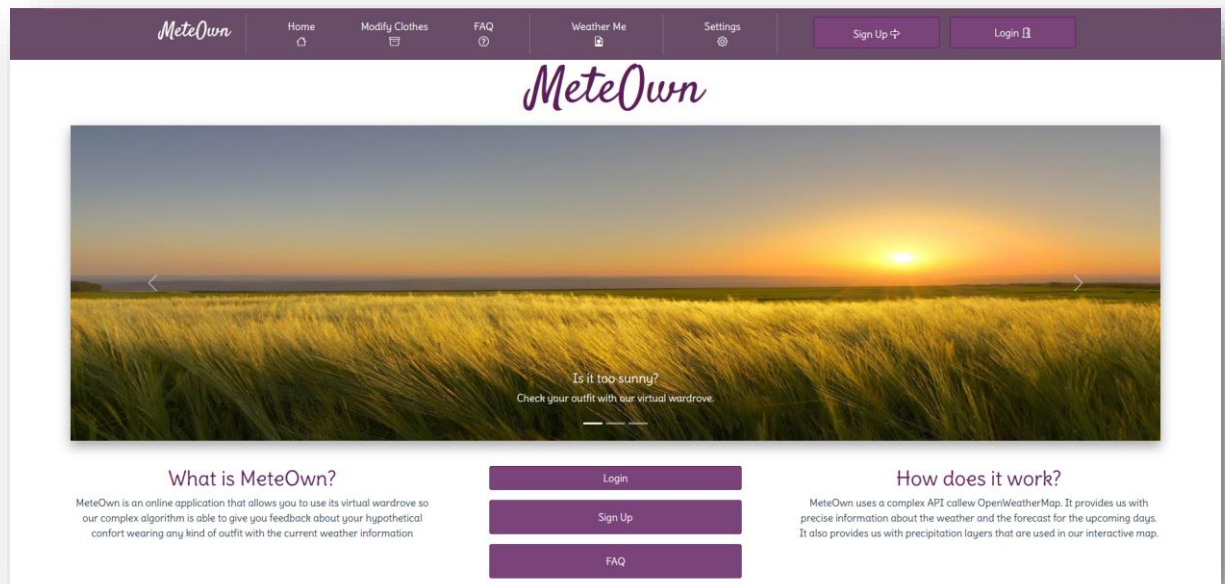


Imagen real de la página Home.

El resultado final es fiel al prototipo. El orden en el menú superior ha sido alterado, así como algún detalle menor de diseño. En caso de no estar autenticado e intentar acceder a una funcionalidad restringida se abrirá la ventana de Login. Además, en caso de estar autenticado se mostrará el Nickname del usuario en lugar del botón de Sign Up, tal como se muestra en la imagen.

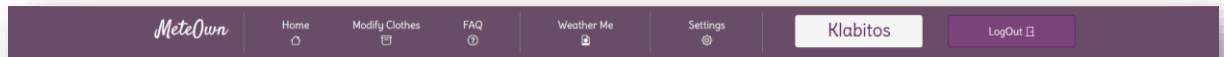


Imagen real del menú en caso de estar autenticado.

Sign Up (Prototipo).

Imagen de un prototipado de la página de Sign Up.

Con el menú superior sin mostrar las funciones reservadas a los usuarios autenticados, se divide el resto de la pantalla en dos secciones, la zona de la izquierda reservada a algún tipo de imagen promocional de la web y a la derecha todos los datos necesarios para el registro en la web.

Estos datos son: Email, contraseña, nombre completo, nickname, ciudad de residencia y fecha de nacimiento.

Se comprobará la validez del formato de mail y la contraseña, que deberá ser escrita dos veces.

Si todo está en orden saltará un popUp con un texto informativo como este:

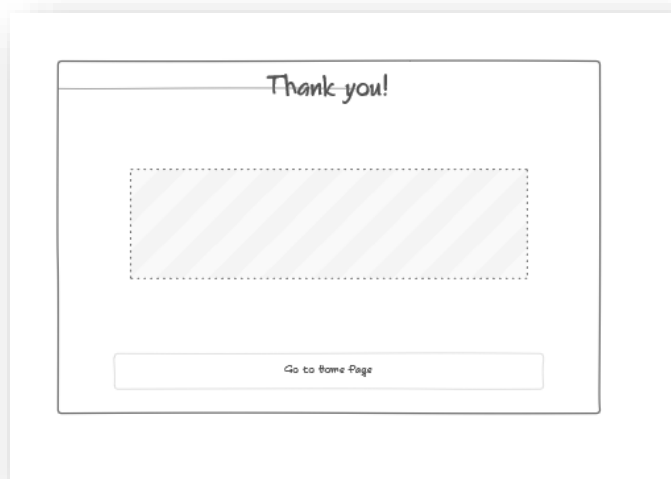


Imagen del prototipado del mensaje de confirmación de usuario registrado.



Sign Up (Resultado).

The screenshot shows the 'Sign Up' page of the 'MeteOwn' application. The navigation bar at the top is purple and contains the following links: Home, Modify Clothes, FAQ, Weather Me, Settings, Sign Up, and Login. The main content area is white. On the left, there is a large image of four friends sitting in a field. On the right, the 'Sign Up' form is displayed. The form has the following fields: Email, Password, Repeat Password, Name, Surname, Username, Nickname, Birth date, and City. The 'Send' button is at the bottom of the form.

Imagen real de la página de Sign Up.

La página de Sign Up también es una fiel representación del prototipado presentado, con la única diferencia de la imagen utilizada, que hace referencia a un grupo de amigos tomando el sol. También se ha cambiado el mensaje de confirmación por uno más minimalista.

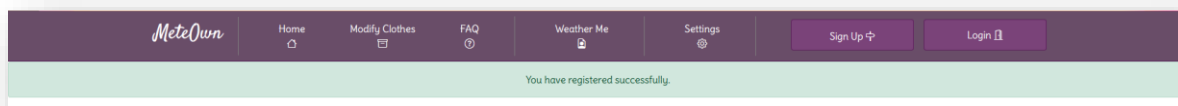


Imagen real del mensaje de confirmación.

Sign In / Login (Prototipo).

Se mostrará un PopUp desde el lugar donde se haya clicado el Sign In. Deberás introducir tus datos y podrás acceder a toda la funcionalidad.

(Si se pulsa la opción de contraseña olvidada te mostrará un mensaje predefinido.)

Imagen del prototipado del PopUp de Login.



Sign In / Login (Resultado).

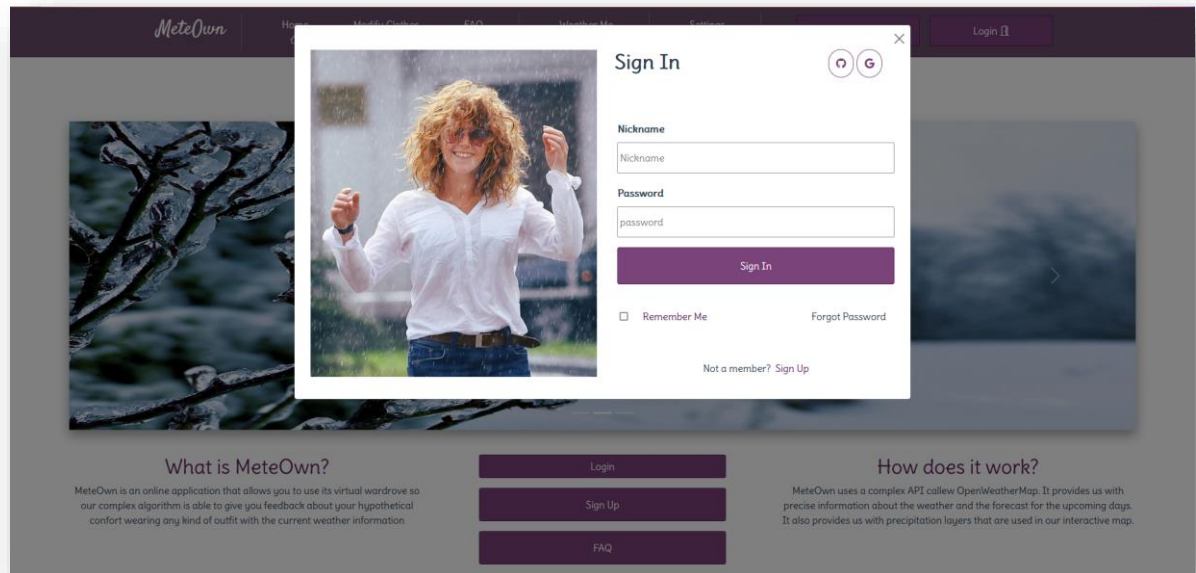


Imagen real del PopUp de Login.

El login es una representación fiel del prototipo, añadiendo una imagen para alegrar la interfaz. El PopUp saltará independientemente de la ventana siempre que se intente acceder a una funcionalidad restringida para usuarios.

Weather Me (Prototipo).

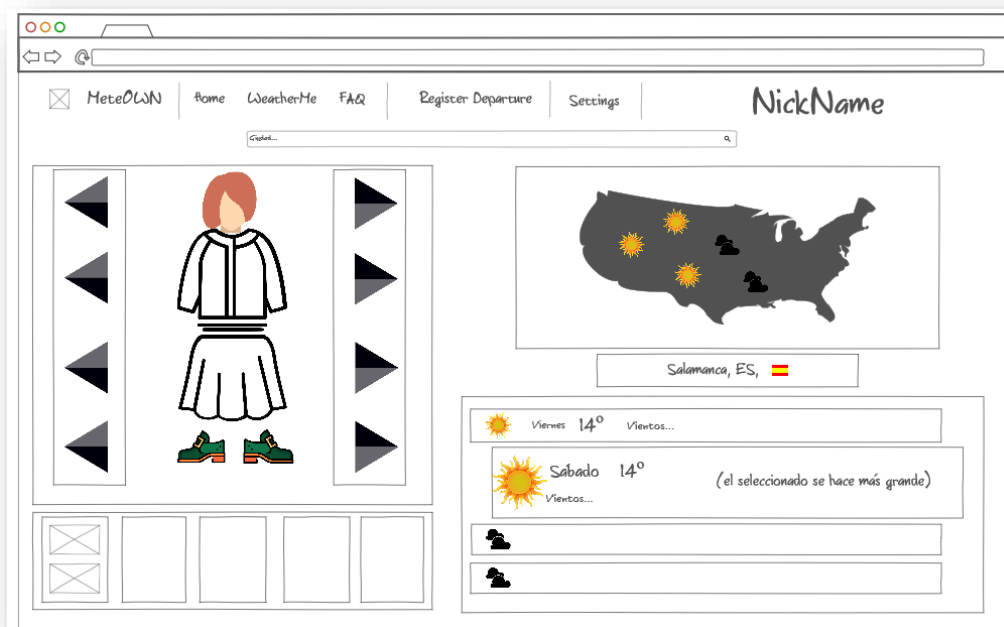


Imagen del prototipado de la página de WeatherMe.

Página principal del usuario, con la inmensa mayoría de la funcionalidad.

Debajo del menú con todas las opciones se encuentra un input de tipo texto en el que al podremos definir la ciudad para la que obtendremos la temperatura, por defecto vendrá la temperatura y el tiempo de la ciudad que hayamos puesto al registrarnos.

Debajo, a la izquierda, nos encontramos con nuestro avatar, al que podremos cambiar las prendas de ropa al hacer clic en las flechas de la izquierda y de la derecha. Cada vez que se cambie la ropa el resultado cambiará y se mostrará en el cuadrado inmediatamente debajo, en el que se mostrará una actividad y una imagen representando el confort de la temperatura que tendremos. Por ejemplo, un símbolo que indique caminar y un símbolo que indique frío.

En la zona lateral derecha, en primer lugar, tendremos un mapa que mostrará distintos estados climatológicos, como incidencia de lluvia y de nubes. Debajo, el nombre de la ciudad y su país en forma de bandera.

Y, por último, se mostrará el tiempo de la ciudad elegida de una manera “tradicional” con la temperatura y la humedad de la ciudad en cada día con una previsión de un par de días. Cuando se seleccione un día, el tamaño de este apartado crecerá.

Weather Me (Resultado).

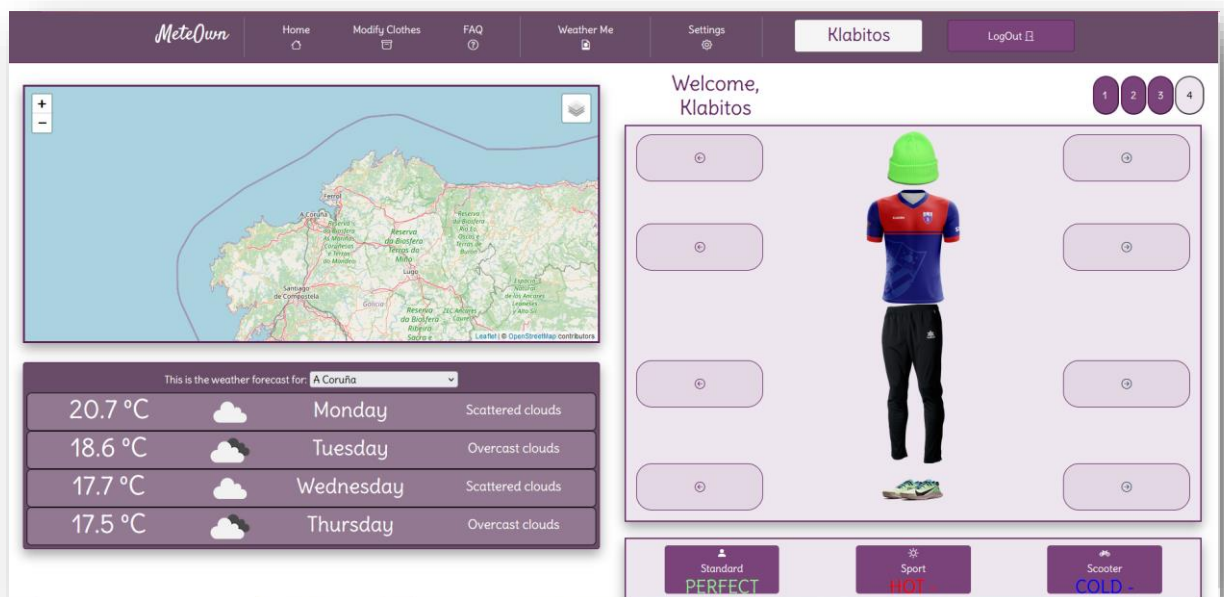


Imagen real de la página de WeatherMe.

En la página de WeatherMe se han realizado numerosos cambios estéticos, el más llamativo el cambio en el orden de los componentes por razones estéticas. Además, se ha cambiado el input de tipo texto por un select con todas las ciudades de España. A parte de esos detalles, se ha añadido la funcionalidad de los Outfits, conjuntos con nombre y prendas predeterminadas.



Settings (Prototipo).

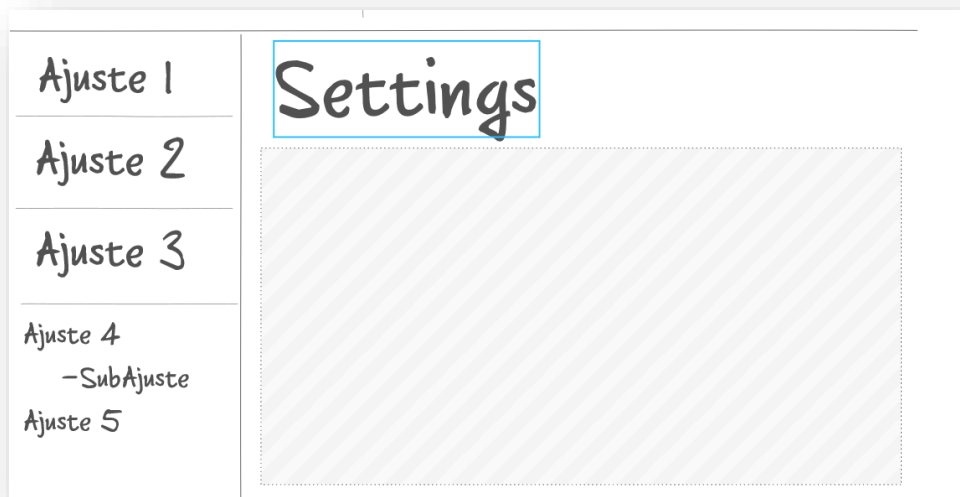


Imagen del prototipado de la página de Settings.

Página con un pequeño menú con ajustes para cambiar los parámetros generales de la cuenta y la mayoría de los datos que se encuentran en la base de datos.

Se podrá hacer clic en cualquier apartado del menú vertical que se encuentra a la derecha y desaparecerá la información anterior y aparecerá la del nuevo apartado. Todas con un botón final para aceptar los cambios realizados en los ajustes.



Settings (Resultado).

| User Information | Form Fields |
|------------------|----------------------------|
| Change city | Username: Klabitos |
| ADMIN | Name: Pablo |
| | Surname: Cruz |
| | E-mail: klabitos@gmail.com |
| | Birth Date: 12 / 06 / 1998 |

Imagen real de la página de Settings.

Register Departure (Prototipo).

Página desde la que podrás cambiar de manera visual y manual los valores de frío y calor de las diferentes prendas, a la vez que podrás registrar tu sensación térmica después de salir a la calle con un conjunto de prendas.

En el apartado de la derecha se encontrará un cuadro desplegable en el que podrás elegir si la prenda es de cabeza, tronco, piernas o pies. Después te saldrá esa zona de tu avatar y podrás cambiar con las flechas hasta seleccionar la prenda elegida. Cada prenda tendrá un input “range” que indicará la resistencia al frío de dicha prenda, y que podremos cambiar y guardar.

En la zona de la izquierda podremos registrar la salida al exterior. Elegiremos las prendas que llevábamos y seleccionaremos en un input de tipo rango el confort con dichas prendas. Cuando pulsemos aceptar la base de datos almacenará los datos y cambiará de manera proporcional los valores de dichas prendas para dichas actividades.



Imagen del prototipado de la página de Register Departure.

Modify Clothes (Antes: Register Departure) (Resultado).

La página de Register Departure ha cambiado a llamarse Modify Clothes, donde por razones estéticas se ha modificado la vista a una zona dividida de modificación de ropas y otra de creación de outfits, en vez de mostrar las dos a la vez.

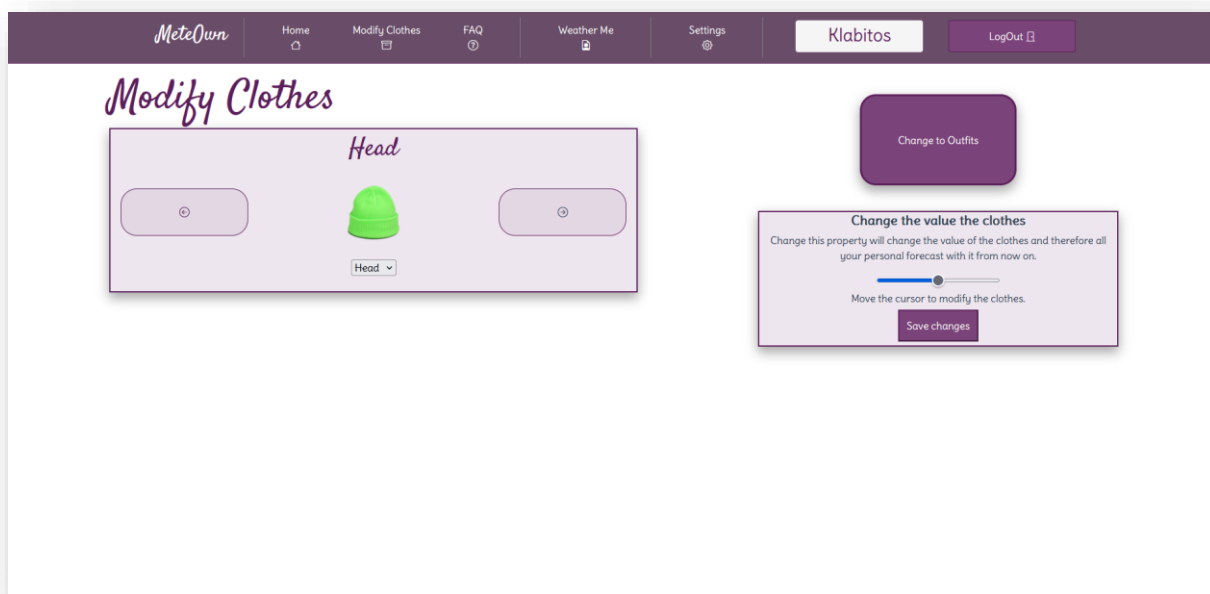


Imagen real de la página de Modify Clothes, zona de modificación de prenda.

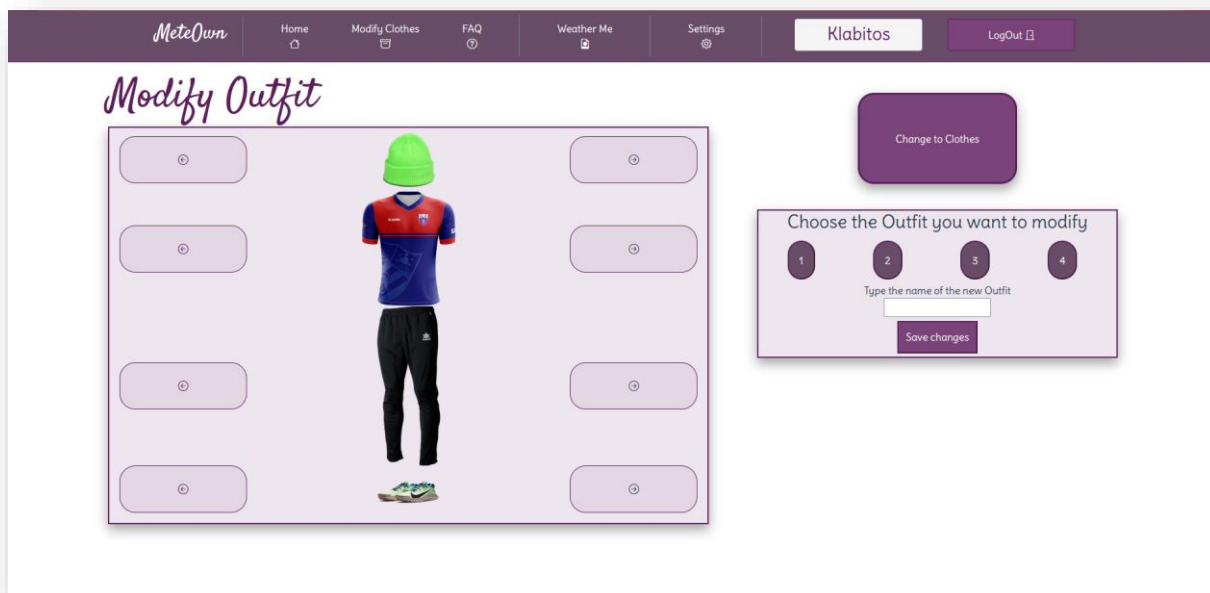


Imagen real de la página de Modify Clothes, zona de creación de Outfits.



Tecnologías escogidas y justificación

Para la creación de Bootstrap se han utilizado las siguientes tecnologías:

- **Vue**

Por la facilidad a la hora de la creación de componentes, con su modelo de single page que permite reunir en un solo archivo todo el CSS, JS y HTML que necesites. Además, su complejidad es mucho menor que algunos frameworks hermanos, como Angular o React.

- **Bootstrap**

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Su facilidad a la hora de utilizarla ha sido uno de los mayores motivos para su implementación.

- **Leaflet**

Leaflet es una biblioteca JavaScript de código abierto que se utiliza para crear aplicaciones de mapas web. Ha sido utilizada en la zona personal de los usuarios y se ha implementado para mostrar distintos tipos de mapas, en este caso en particular el mapa de precipitaciones, de viento y de nubes. Es una biblioteca muy sencilla de implementar y totalmente personalizable. Además, está preparada a la perfección para ser usada con nuestra API meteorológica.

- **OpenWeatherMap API**

OpenWeatherMap es una API con funciones gratuitas que permite obtener el tiempo y las condiciones meteorológicas actuales en cualquier parte del mundo, con una precisión aceptable. En el caso de MeteOwn se ha hecho uso de una licencia de desarrollador, argumentando que se va a utilizar con fines educativos. Esta licencia nos proporciona mapas informativos (que usamos con Leaflet) y la previsión del tiempo.



- **Spring Boot**

Spring Boot es una herramienta que nos facilita la creación de aplicaciones web. Se encarga de todas las configuraciones necesarias y gestiona todas las inyecciones de dependencias. Una de las grandes ventajas de usar esta tecnología es el respaldo de la comunidad y la gran cantidad de foros con dudas y respuestas que existen.

- **Spring Boot Security**

Otra de las razones por las que este proyecto ha sido desarrollado con Spring Boot es su vertiente de seguridad, Spring Boot Security, que nos proporciona herramientas para gestionar los distintos roles y permisos para las llamadas que se van a realizar en nuestra aplicación web. Es una manera rápida y efectiva de configurar la seguridad y la distribución de usuarios en un sitio web.

- **MySQL**

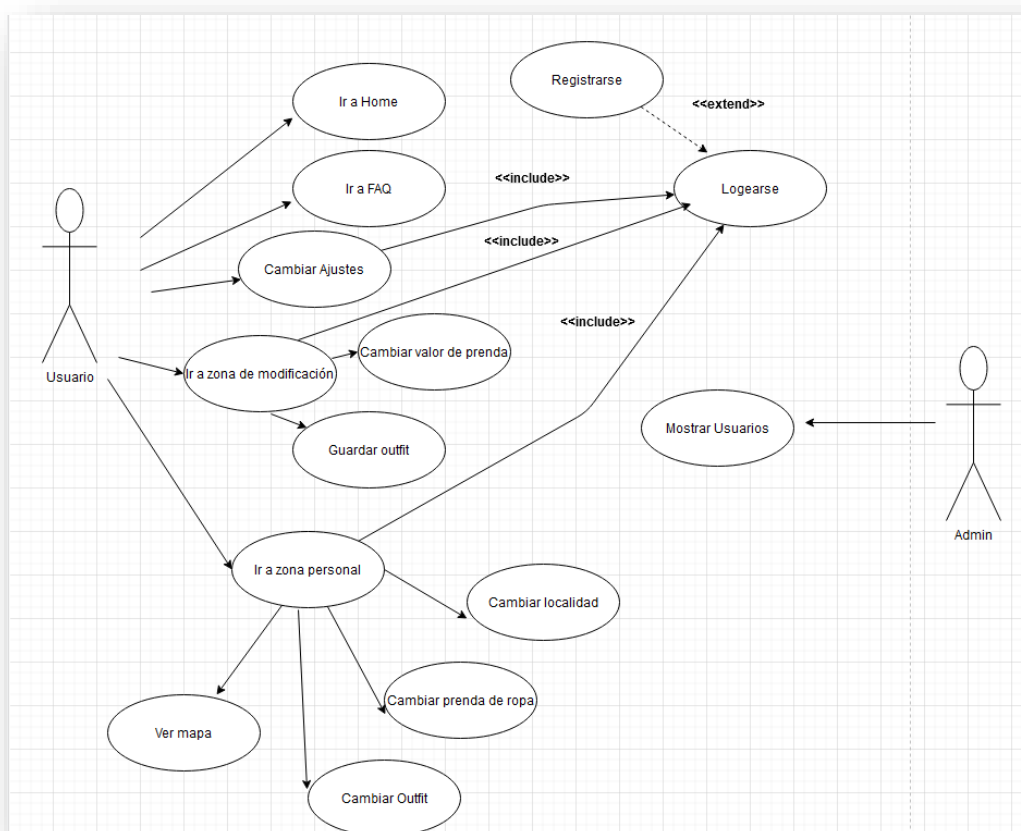
MySQL es un sistema de gestión de bases de datos relacional. Es uno de los más usados y esa ha sido la razón de su uso. Debido a su gran número de usuarios es muy fácil encontrar la solución a cualquier tipo de duda en internet.

- **ObjectAID**

Plugin especializado en la generación de diagramas de clase a partir de código. Es muy sencillo de usar y permite generar todo tipo de imágenes y elementos con tan solo arrastrar las clases.

3. Diseño de la aplicación

3.1. Diagramas y definición de casos de uso





Definición de casos de uso:

Nombre: Visualización Página Principal -

Id: 001

Actores: Usuario

Descripción: Visualización Página Principal

Pasos:

1. Se accede a la página principal por medio de enlace.
2. Se carga la página.
3. Se muestra la página.
4. El usuario puede navegar por la página a través de los distintos hipervínculos en los botones.

Nombre: Visualización de FAQ -

Id: 002

Actores: Usuario

Descripción: Visualización de FAQ

Pasos:

1. Se accede a la página por medio de enlace.
2. Se carga la página
3. El usuario podrá acceder a distintas preguntas frecuentes y ver la respuesta.



Nombre: Modificación de ajustes -

Id: 003

Actores: Usuario Autenticado

Descripción: Modificación de ajustes

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se accede a la página por medio de enlace en el menú.
2. Se carga la página de ajustes con una vista general de ajustes básicos
3. Si muestra un menú básico de ajustes a la izquierda.
4. El usuario puede moverse con el menú por las distintas secciones con ajustes.
5. El usuario modifica algún ajuste.
6. SI: El usuario pulsa el botón guardar:
 - 6.1 SI: Los ajustes introducidos son válidos: Se actualizan y se carga vista general de ajustes
 - 6.2 SI: Los ajustes introducidos no son válidos. Se muestra un mensaje de error.

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Ir a la zona de modificación de prendas -

Id: 004

Actores: Usuario Autenticado

Descripción: Ir a la zona de modificación de prendas

Precondiciones: Estar registrado en la web y autenticado.



Pasos:

1. Se accede a la página por medio de enlace.
2. Se carga la página

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Cambiar el valor de temperatura de una prenda -
Id: 005

Actores: Usuario Autenticado

Descripción: Cambiar el valor de temperatura de una prenda

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se muestra un SELECT con los distintos tipos de prenda (Cabeza-Tronco-Piernas-Pies).
2. Se carga una imagen con una prenda de las del tipo seleccionado, y una flecha a la izquierda y otra a la derecha.
3. Se muestra una barra inferior como Input Range con el valor de calor de dicha prenda.
3. Si se clica en alguna de las flechas se cambia la prenda mostrada.
4. El usuario modifica el valor del input.
5. Si el usuario pulsa guardar, se guardan los cambios.

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)



Nombre: Acceder a la zona personal -

Id: 007

Actores: Usuario Autenticado

Descripción: Acceder a la zona personal

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se accede a la página por medio de enlace.
2. Se carga la página
3. Se carga el mapa.
4. Se carga el selector de localidad y la temperatura actual y previsible
5. Se carga el cambiador de ropa.

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Log in -

Id: 008

Actores: Usuario

Descripción: Log in

Pasos:

1. Se pulsa sobre la opción de Log In
2. Aparece una ventana Modal con la información de login, superpuesta a la pantalla actual y en la mitad de la misma.
- 3.0 SI: Se pulsa sobre el botón de “No estoy registrado”: Caso Registro (Id: 009)



- 3.1. Se rellenan los datos.
4. Si se pulsa en Log-In se comprueba en la base de datos
 5. SI: Los datos han sido verificados se accede a la web a la página Home
 6. SI NO: Se muestra un mensaje de error.

Alternativa: El paso 1 puede ser sustituido por: Intento de acceso a una zona restringida por medio de botón.

Nombre: Registro -
Id: 009

Actores: Usuario

Descripción: Registro

Pasos:

1. Se llega a la ventana de registro.
2. Se carga la vista de registro.
3. Se muestran todos los campos a rellenar.
4. Se muestra un Select con un desplegable de ciudades
5. SI: Se pulsa en guardar y todos los datos son adecuados al formato se guarda el usuario.
 - 5.1. Si se confirma se borran los datos y se manda a la vista de Home.
6. SI no son adecuados: Se muestra un mensaje de error.



Nombre: Cambiar prenda de ropa -

Id: 010

Actores: Usuario Autenticado

Descripción: Cambiar prenda de ropa

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

0. Se visualiza el cambiador de ropa
1. Se carga la configuración de ropa inicial, con cabeza, tronco, pantalón y zapatillas. A cada lado de cada sección aparece una flecha
2. Se carga una estimación de sensación térmica con la configuración de ropa.
3. Al pulsar una de las flechas, se cambia la prenda de la misma zona y se vuelve a mostrar la sensación térmica actualizada

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Cambiar localidad -

Id: 011

Actores: Usuario Autenticado

Descripción: Cambiar localidad

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se visualiza la previsión de la localidad y la información meteorológica
2. Si se pulsa en el nombre de la localidad, se muestra un Select



3. Si se pulsa en otra localidad se actualiza la información meteorológica y la previsión para dicha localidad.

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Cargar outfit -

Id: 012

Actores: Usuario Autenticado

Descripción: Cargar outfit

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se muestra un avatar con 4 flechas laterales a cada lado
2. Se elige un número para cargar dicho outfit.
- 2.5 Se ilumina dicho número y se carga la combinación de ropas en caso de existir.
4. Se puede cambiar ropa con Cambiar prenda de ropa (Id: 010)

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Guardar nuevo outfit -

Id: 014

Actores: Usuario Autenticado

Descripción: Guardar nuevo outfit



Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se muestra un avatar con 4 flechas laterales a cada lado
2. Se elige un número para cargar dicho outfit.
3. Se selecciona un nombre para el Outfit.
4. Si se pulsa Guardar, se modifica el Outfit o se crea si no había uno preexistente.

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

Nombre: Ver mapa -

Id: 015

Actores: Usuario Autenticado

Descripción: Visualización del mapa

Precondiciones: Estar registrado en la web y autenticado.

Pasos:

1. Se hace clic en el mapa
2. Si pulsa en el cuadro superior derecho se muestran las capas.
3. Si se pulsa en una capa se activa

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)



Nombre: Ver todos los usuarios -

Id: 015

Actores: Admin

Descripción: Ver todos los usuarios

Precondiciones: Estar registrado en la web y autenticado.

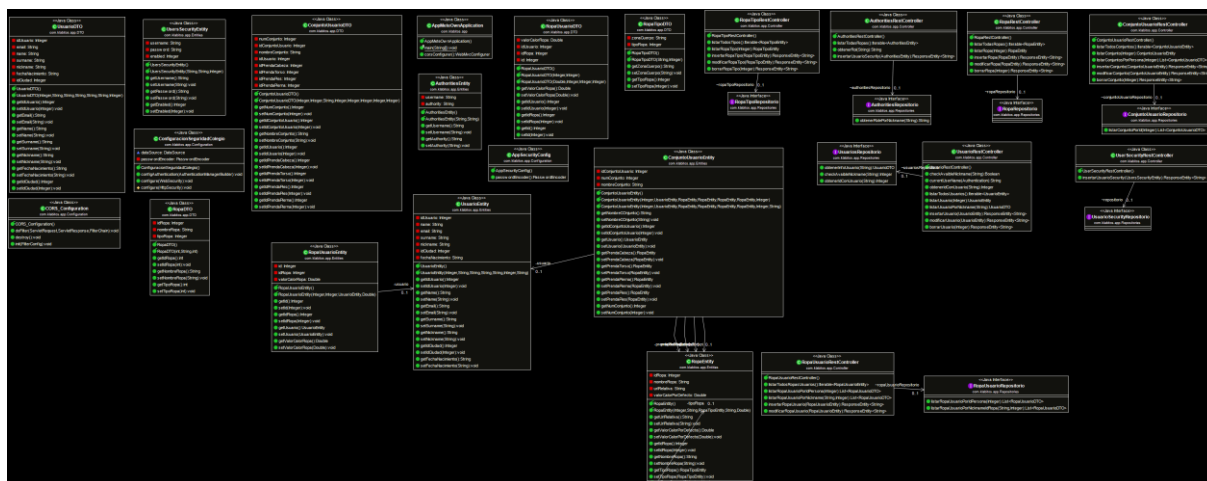
Pasos:

1. Se carga la página de ajustes.
2. Si el usuario es ADMIN se muestra la pestaña admin
3. Si se pulsa en admin se carga la pestaña.
4. Si se pulsa en el botón se carga una página donde se ven todos los usuarios.

Alternativa: Si no está autenticado: Caso Log-In (Id: 008)

3.2. Diagrama de clases

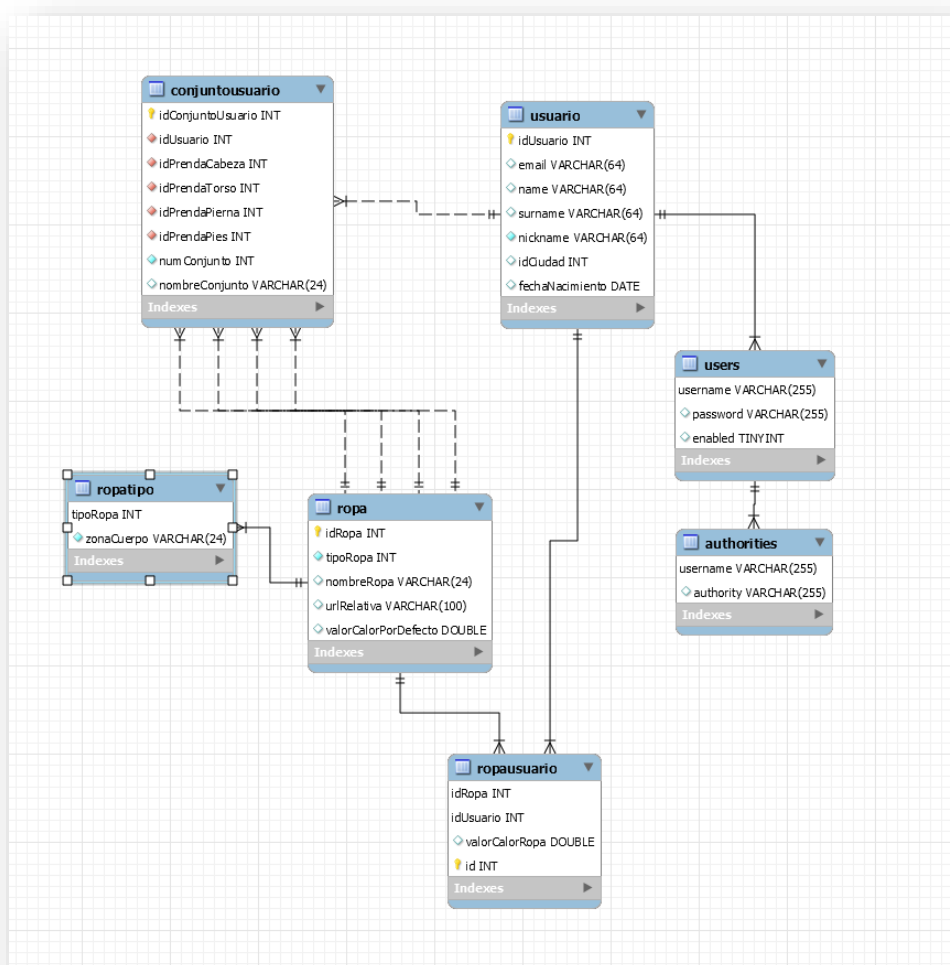
Dentro del proyecto se encuentra el archivo .ucls y la imagen.



3.3. Modelo entidad-relación

Hemos elegido mostrar el modelo completo de la base de datos con los tipos usados, las claves y las relaciones:

Modelo de base de datos:





Arquitectura de la aplicación

3.4. Estructura del proyecto

La aplicación MeteOwn tiene tres pilares fundamentales, la base de datos, la parte “back-end” y la parte “front-end”.

La base de datos se encarga de almacenar toda la información importante relativa a usuarios, prendas e información personal. La parte front-end maneja las vistas y es la parte que se carga en cliente, siendo el lugar donde el usuario puede hacer distintas llamadas por medio de una interfaz web. Por último, el back-end se encarga de controlar la API-REST, manejando las llamadas que hacen los usuarios en la parte front y dando respuestas acordes a la base de datos.

FRONT-END

Se ha utilizado una arquitectura basada en componentes y vistas. Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. En nuestro caso las distintas vistas de la interfaz web están compuestas de distintos componentes, todos funcionales individualmente.

- Vistas

- FAQ

Contiene información sobre las preguntas más frecuentes que un usuario puede hacerse al utilizar la aplicación web, también las respuestas.

- Home

Página principal, sirve como página de bienvenida.



- ModifyClothes

Página con toda la funcionalidad relativa a la personalización de las prendas. Cada usuario puede modificar los valores térmicos de cada ropa. Además, en esta página se pueden crear y modificar los distintos outfits.

- Settings

Muestra toda la información personal del usuario activo y permite cambiarla.

- SignUp

Permite registrar a nuevos usuarios en la aplicación.

- Users

Vista a la que solo el ADMIN tiene acceso. Muestra todos los usuarios registrados y su información personal.

- WeatherMe

Página principal del usuario en la aplicación. Contiene un mapa situado en tu localidad y una previsión del tiempo en la misma. Muestra el armario virtual en el que puedes elegir distintas prendas u outfits creados con anterioridad. También muestra la estimación de sensación térmica con dichos outfits en el lugar elegido.

- **Componentes**

La mayor parte de los componentes presentados a continuación han sido diseñados como tal en caso de ser necesaria su incorporación en futuras actualizaciones de la web y para separar las distintas funcionalidades.

- Armario Carrousel

Muestra el armario virtual con los indicadores de cambio de prenda

- Calendario Tiempo



Muestra la previsión del tiempo durante cuatro días.

- MapaConInfo

Muestra un mapa con distintas capas que muestra fenómenos meteorológicos y su intensidad en forma de colores.

- Menu

Contiene todos los enlaces a las distintas vistas y es el principal punto de comunicación entre vistas de la web.

- ModalLogin

Muestra el formulario de autenticación con usuario y contraseña.

- OutPutArmario

Se encarga de mostrar una caja con la sensación térmica.

- **Mixins**

Para evitar repetir código se han utilizado mixins con funciones específicas recurrentes, como son:

- comprobacionTipoRol

Encargado de comprobar si el usuario autenticado tiene el rol de USER o de ADMIN.

- comprobarLoginCorrecto

Encargado de comprobar que la autenticación se ha realizado de manera satisfactoria y que el usuario adecuado está en sus páginas personales. Es un método de seguridad que comprueba si intentas acceder a una página que no es la tuya.



- funcionalidadRopas

Contiene un conjunto de métodos encargados de gestionar todo el proceso detrás del cambio de ropa en el armario virtual.

- obtenerIdUsuario

Obtiene el Id del usuario activo para realizar distintas llamadas.

- obtenerSetsUsuarioActivo

Obtiene un array con todos los outfits del usuario activo.

BACK

La arquitectura REST se tiene como principio el envío de peticiones por parte del cliente para recuperar o modificar recursos, y el servidor responde con el resultado, que puede ser con los datos que hemos pedido o el estado de la petición. En el back podemos encontrar los siguientes paquetes dentro de **Main**:

- **App.**

Contiene la clase que inicia la parte Back de la aplicación.

- AppMeteOwnApplication.java

Inicia la aplicación e incluye un @Bean para evitar problemas hipotéticos del CORS.



- **Configuration**

Contiene todas las clases necesarias para la configuración de la seguridad de la aplicación:

- AppSecurityConfig.java

Un encoder para las contraseñas que se insertan a la Base de Datos. El formato es BCrypt.

- ConfiguracionSeguridadColegio.java

Clase con todas las rutas permitidas y bloqueadas a los distintos usuarios y administradores. Incluye clausulas sobre el login y el logout como las páginas predefinidas para ambos eventos.

- CORS Configuration.java

Segunda clase de configuración ante hipotéticos problemas del CORS. En caso de que falle la primera esta controlaría el error.

- **Controller**

Contiene todos los mapeos de las distintas peticiones que se pueden realizar y los métodos que las resuelven:

- AuthoritiesRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de Authorities de la base de datos.

- ConjuntoUsuarioRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de ConjuntoUsuario de la base de datos.



- RopaRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de Ropa de la base de datos.

- RopaTipoRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de RopaTipo de la base de datos.

- RopaUsuarioRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de RopaUsuario de la base de datos.

- UserSecurityRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de Usuarios de SpringSecurity de la base de datos.

- UsuarioRestController.java

Responde a las llamadas para obtener o modificar información de la tabla de Usuarios de la base de datos.

- **DTO**

Encargado de mapear las distintas tablas de la base de datos. Son necesarias para muchas de las respuestas de los métodos de los controladores que piden información:

- ConjuntoUsuarioDTO.java
- RopaDTO.java



- RopaTipoDTO.java
- RopaUsuarioDTO.java
- UsuarioDTO.java

- **Entities**

Encargado de realizar un mapeo exhaustivo de todas las tablas de la base de datos, con nombres, tipos y claves:

- ConjuntoUsuarioEntity.java
- RopaEntity.java
- RopaTipoEntity.java
- RopaUsuarioEntity.java
- UsuarioEntity.java
- UserSecurityEntity.java
- AuthoritiesEntity.java

- **Repositories.**

Interfaz usada para el correcto funcionamiento de la API, hereda de CrudRepository y contiene métodos personificados para realizar las llamadas que no están automatizadas:

- ConjuntoUsuarioRepositorio.java
- RopaRepositorio.java
- RopaTipoRepositorio.java
- RopaUsuarioRepositorio.java
- UsuarioRepositorio.java
- UserSecurityRepositorio.java
- AuthoritiesRepositorio.java



Por otra parte, en la carpeta **resources** tenemos el `application.properties`, con valores importantes para el correcto uso de la aplicación, como el nombre de la base de datos, el usuario y la contraseña.



3.5. Librerías externas utilizadas

LEAFLET

Leaflet, tal y como hemos mencionado antes, es una biblioteca JavaScript de código abierto que se utiliza para crear aplicaciones de mapas web. Es una biblioteca muy sencilla de implementar y totalmente personalizable. Además, está preparada a la perfección para ser usada con nuestra API meteorológica.

BOOTRAP

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Su implementación reside en el uso de clases personalizadas en los distintos componentes que queremos modificar.



4. Manual de despliegue

Despliegue a partir de Docker:

Requerimientos:

- Docker

Los nombres de las 3 imágenes que necesitaremos son:

- klabitos/imgbbdd4
- klabitos/backmeteown3
- klabitos/vue_tipo3_parte25

Comenzaremos creando una red donde desplegaremos los contenedores, para ello desde nuestro terminal docker ejecutaremos lo siguiente:

docker network create appmeteown

Para ejecutar nuestra imagen de la base de datos:

***docker run -d --network appmeteown --name mi_mysql -e MYSQL_ROOT_PASSWORD=123456
-e MYSQL_DATABASE=meteown_proyecto -e MYSQL_PASSWORD=123456 -p 3307:3306
klabitos/img_bbdd4***

Después, para entrar en ese contenedor:

docker exec -it mi_mysql mysql -uroot -p

Nos pedirá una contraseña, que será 123456. (En caso de fallar, esperar 1 minuto y volver a intentar). Ahora crearemos un usuario escribiendo tres comandos:

***create user 'mysqluser' identified by 'secret';
GRANT ALL PRIVILEGES ON *.* TO 'mysqluser'@'%';
exit;***



Ahora que tenemos desplegada la base de datos de manera correcta seguimos con el back:

```
docker run -d --network appmeteown --name back -p 8080:8080 klabitos/backmeteown3
```

Por último, desplegaremos el front.

```
docker run -d --rm --name appmeteown_front --network appmeteown -p 8081:8080 klabitos/vue_tipo3_parte25
```

Ahora desde el navegador podremos acceder al proyecto con ***localhost:8081/***

Despliegue a partir de archivos en GitHub:

Requerimientos para un funcionamiento óptimo:

- MySQL Workbench y server
- Eclipse Enterprise Edition con Spring Boot
- VUE3 Cli
- Visual Studio
- Node. js version 8.9 o superior

Con los archivos de GitHub desplegaremos la aplicación de manera local. Para ello, en primer lugar, importaremos la base de datos, usando el script que se incluye. Luego en nuestro proyecto back habrá que configurar la contraseña de la base de datos adecuada al usuario, para ello modificaremos el application.properties.

Para el back, importaremos en eclipse el proyecto como Maven, y lo ejecutaremos con Spring Boot.

Para el front, abriremos el proyecto con Visual Studio, instalaremos las dependencias con **npm install** y ejecutaremos **npm run serve**.

El back habrá sido desplegado en el puerto 8080, el front en el 8081, por lo que podremos acceder a la web en ***localhost:8081/***



Navegadores soportados:

Safari 10.1+, Chrome 54+ y Firefox 63+.

Requisitos hardware:

Procesador Pentium 4 o superior con soporte SSE2, 550MB de RAM y al menos 200MB de espacio libre en el disco duro.