

Cliente WEB:

Este componente es la parte visual de la aplicación, en este se ven reflejados los datos que se guardan en la base de datos. El cliente web se comunica con el modelo de objetos a través de protocolos como HTTP y realiza request y responses a través de un controlador cuya función es la de intermediario entre el modelo de negocio y la parte web.

Capa cliente: Dentro de esta capa se muestran los datos al usuario, a través de las vistas el mismo visualiza los datos y el controller sirve como un intermediario para conectar las vistas con los distintos servicios de donde se obtiene los datos. En nuestro modelo poseemos las siguientes vistas:

Vistas:

- Vista de Login: Primera pantalla que visualiza el usuario, en esta vista se ingresa a la aplicación web a través de un navegador y un protocolo.
- Vista de Eventos: Se visualiza por default al ingresar correctamente en la aplicación, en esta pantalla se pueden crear/modificar/eliminar eventos
- Vista Guardarropas: Se visualizan los datos correspondientes a los guardarropas, en esta pantalla se puede crear/modificar y eliminar un guardarropa.
- Vista de Prendas: Se debe primero pasar por la vista de Guardarropas para poder llegar a esta vista, se visualiza toda la información correspondiente a una prenda. Desde esta se pueden crear/modificar y eliminar una determinada prenda.
- Vista Calendario: En esta vista se representan los eventos en forma de un calendario, el cual permite filtrar por día, semana y mes.

Servidor:

El servidor se compone todos los elementos y objetos utilizados en el modelo de negocio que poseen la lógica de este. Al igual que el cliente, el servidor se encuentra estructurado en una arquitectura de capas, las cuales se comunican entre si mediante interfaces. Ofreciendo la ventaja de poder separar responsabilidades permitiendo realizar cambios sobre una capa reduciendo el impacto en las demás, testear cada capa por separado e incluso reutilizar capas, si hace falta. El servidor esta montado en MS Azure.

RouterController: Es el encargado de manejar los request que solicita el cliente WEB y devuelve el response que envía el Cliente del modelo. Este controller es el encargado de exponer los endpoints que va a consumir la aplicación web. Maneja todos los responses status y se encarga de transformar la respuesta del modelo de objetos a una respuesta que el cliente WEB pueda entender, en este caso las request consumen JSON así que las response también son JSON. Para la comunicacion utiliza el protocolo REST.

Capa de servicios externos: Esta capa se ocupa de comunicarse con la api de OpenWeather utilizando la interfaz Proveedor. Esta interfaz luego es implementada por el Administrador de Proveedores, la cual permite obtener información del pronóstico actual, pronóstico extendido y obtener la temperatura de un día específico.

Capa de dominio: En esta capa se encuentran todos los objetos del negocio, en las cuales se maneja la lógica del mismo, esta capa es la intermedia entre los endpoints y servicios que consume el cliente web y la capa de persistencia de datos. Cuando se requiere persistir un objeto se comunican entre capas a través de una interfaz en común para lograr la persistencia de los mismo. Para obtener los datos persistidos también se comunican con la interfaz y esta se lo envía a la interfaz del controller quien recibe los request desde el cliente web.

- Abrigo: son aquellos que pueden ser utilizados por los Usuarios. Los mismos poseen un rango de clima dentro del cual pueden ser utilizados y un nivel de abrigo dentro de ese rango.
- Atuendo: un Atuendo estará compuesto por un conjunto de Prenda que se detallan a continuación:
 - Prenda superior.
 - Prenda inferior.
 - Prenda calzado.
 - Prenda accesorio.
 - Calificación de sugerencias: el usuario deberá poder volver a ver las sugerencias previamente aceptadas, y calificarlas según cuánto efectivamente le abrigaron.

Los atuendos pueden ser aceptados o no por los usuarios para un evento específico. El sistema solo permite aceptar un atuendo para el mismo evento. Asimismo, el usuario puede calificar las sugerencias previamente aceptadas, según cuánto efectivamente lo abrigaron.

- Prenda: son aquellas que los usuarios llevarán puestas. Como se detalló previamente, un conjunto de prendas representará un atuendo. Las prendas están compuestas por:
 - Nombre.
 - Tipo.
 - Material.
 - Color primario.
 - Color secundario.

El color primario de una prenda no puede ser igual al color secundario de la misma.

- Evento: los usuarios pueden concurrir a diferentes eventos. Los mismos tienen lugar en una fecha y hora determinada y los usuarios que asisten a él deben asistir con el atuendo adecuado ya que no todos los atuendos son aceptados por todos los eventos.
- Guardarropa: aquí se guardan las prendas. Los guardarropas poseen un espacio determinado, por lo que no es posible guardar cualquier cantidad de prendas en ellos. Los

- guardarropas tienen la característica de que pueden ser compartidos, es decir, que más de un usuario pueden tener el mismo guardarropa.
- Suscripción Gratuita: en este caso se determina que la cantidad máxima de prendas que puede tener un usuario es menor a 5
 - Suscripción Premium: no se establece una restricción en la cantidad de prendas que pueda tener un usuario.
 - Usuario: los usuarios serán partícipes y utilizarán lo previamente descripto. Los mismos podrán aceptar o rechazar atuendos, asistir o no a eventos, y serán ellos quienes administren sus propios guardarropas, quitando o agregando prendas a los mismos.

Capa de Persistencia: En esta capa se persisten los objetos del modelo, se realiza el mapeo de un objeto a una entidad relacional mediante el ORM Hibernate. Estas entidades se guardan en la base de datos denominada QueMePongo. Cada entidad de nuestra base de datos posee un repositorio el cual se encarga de persistir, actualiza, buscar o eliminar un registro en la base de datos correspondiente a la entidad. Por ejemplo, el repo Usuario se encarga de agregar o eliminar un usuario en la base de datos QueMePongo. Se utiliza MariaDb como base de datos SQL.