

Memoria-Proyecto final TDW

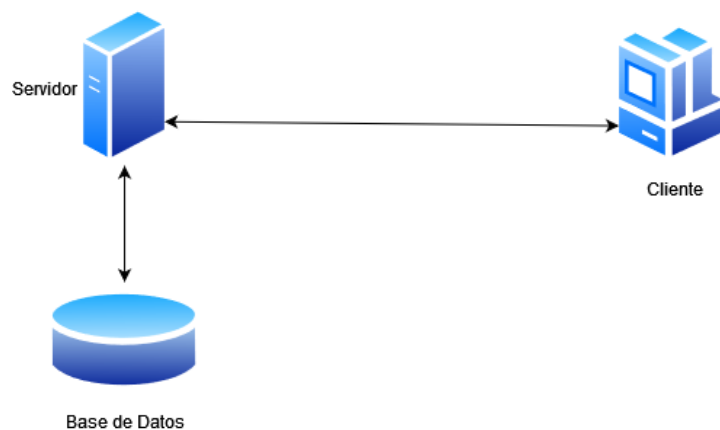
Introducción

El desarrollo de este proyecto ha consistido en la fusión de las dos prácticas, práctica de frontend y práctica de PHP, y en la ampliación de la funcionalidad incorporando casos de uso para la gestión de usuarios de la aplicación y perfil de usuario.

En primer lugar, se pretendía adaptar las operaciones disponibles en la práctica de frontend para que la persistencia estuviese a cargo del backend en vez del sistema de persistencia local del navegador. Para ello, se ha hecho uso de Ajax enviando a la parte del servidor las peticiones adecuadas en cada momento. Además, cabe resaltar que las peticiones Ajax construidas se hacen a través de la API REST que fue implementada en la práctica de PHP.

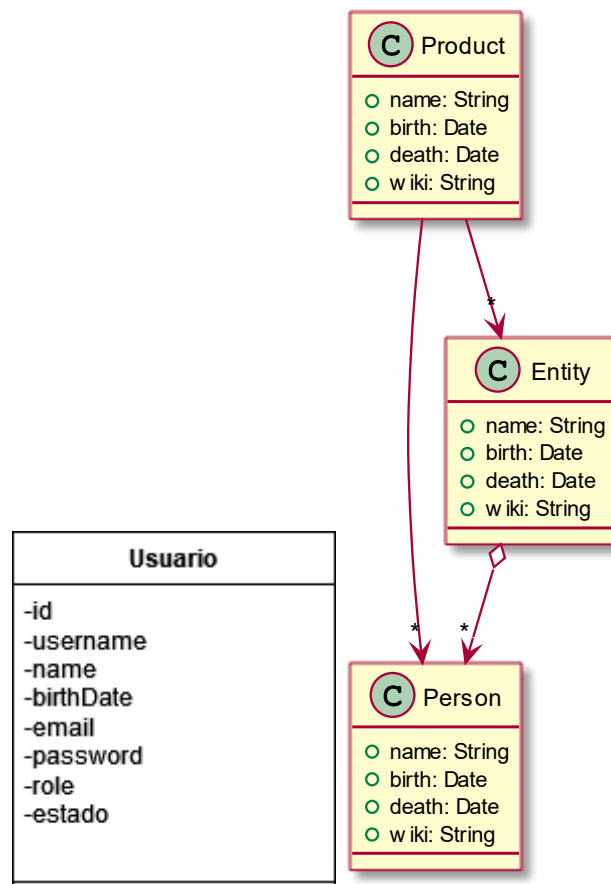
En segundo lugar, la ampliación de la funcionalidad ha supuesto, además de más código en la parte cliente, cambios en los scripts PHP del servidor añadiendo atributos a la entidad User y modificando controladores propios de esta entidad para que cumpla con el objetivo marcado.

Dicho esto, podemos concretar que este proyecto aplica una arquitectura cliente-servidor de la siguiente manera:



Modelo de datos

Este proyecto sigue el siguiente modelo de datos:



Los requisitos del proyecto han hecho que se incorporen en la clase User (o usuario) los atributos name, birthDate y estado, nuevos respecto al modelo de datos de la práctica de PHP. El resto de las clases como Producto, Persona y Entidad no han sido modificadas.

Problemas encontrados

Los principales problemas encontrados los podemos dividir entre la parte cliente y la parte servidor. Para la parte del servidor, el primer objetivo fue modificar la clase User para incorporar aquellos atributos nuevos y que son necesarios para la implementación de las funcionalidades nuevas del sistema. A continuación, se modificó la clase LoginController para que se tuviese en cuenta el estado del usuario al logearse y discriminar a aquellos con estado inactivo o no autorizado. Por último, fue necesario modificar las clases CreateCommand y UpdateCommand, controladores para Usuario, para que en éstas se procesen los nuevos atributos.

Para la parte cliente, la primera tarea a realizar fue adaptar las operaciones CRUD para productos, personas y entidades a las operaciones de la API REST. Para llevar a cabo esto, se han reutilizado los métodos ya existentes que realizaban estas operaciones en la parte cliente, pero modificándolas para que, usando Ajax, realicen peticiones HTTP del tipo cGET, GET, POST, PUT, DELETE. A continuación, se implementaron las mismas operaciones HTTP para usuarios. Cabe destacar que los usuarios READER únicamente pueden hacer peticiones de tipo GET para productos, personas y entidades. Los usuarios de tipo WRITER pueden realizar un CRUD completo sobre productos, personas, entidades y usuarios.

En segundo lugar, otra tarea fue la de incorporar un inicio de sesión para todos los usuarios del sistema para acceder a la aplicación a través de su identificador de usuario y contraseña. Para esto, la presentación de la aplicación se resume en un formulario para hacer inicio de sesión y

otro formulario para registrarse. Si al introducir los datos de usuario y contraseña estos se encuentran en la base de datos la aplicación permite el acceso y guarda el JWT, resultado presente en la cabecera Authorization de la respuesta, en el sessionStorage ya que es necesario para realizar la mayoría del resto de peticiones HTTP. Además, el inicio de sesión debe discriminar a aquellos usuarios con estado 'unauthorized' o 'inactive' ya que no tienen permiso de acceso.

El registro de usuarios funciona bajo el segundo formulario en la página de presentación. Este cuenta con una entrada de texto para introducir el nombre del nuevo usuario y en caso de que no exista en la base de datos se sigue adelante con el registro pidiendo contraseña, email, etc. Para comprobar si se trata de un usuario existente se hace una petición HTTP con Ajax a la ruta /api/v1/users/username/{username}; si devuelve un código 404 el usuario no existe y por ende se puede registrar. Una vez terminado este proceso, la cuenta de usuario queda en estado 'unauthorized' hasta que un usuario WRITER lo autorice.

Por último, otra de las funcionalidades a incorporar fue la gestión del perfil de usuario. En el inicio de la página se muestra un botón con el identificador de usuario que lleva a una pantalla con la información del usuario permitiendo editarla y dejando opción a cambiar la contraseña de acceso.