

Pynstagram

Erik Pereira Fernández

Introducción

Para mi proyecto he optado por una página para subir fotografías al estilo instagram, donde los usuarios pueden subir las fotografías que quieran, dar like a fotografías e incluso comentar en dichas fotografías.

He considerado 3 entidades principales, la entidad Usuario, la entidad Fotografía, y la entidad Comentario, ya que a mi modo de ver tienen entidad en sí mismas.

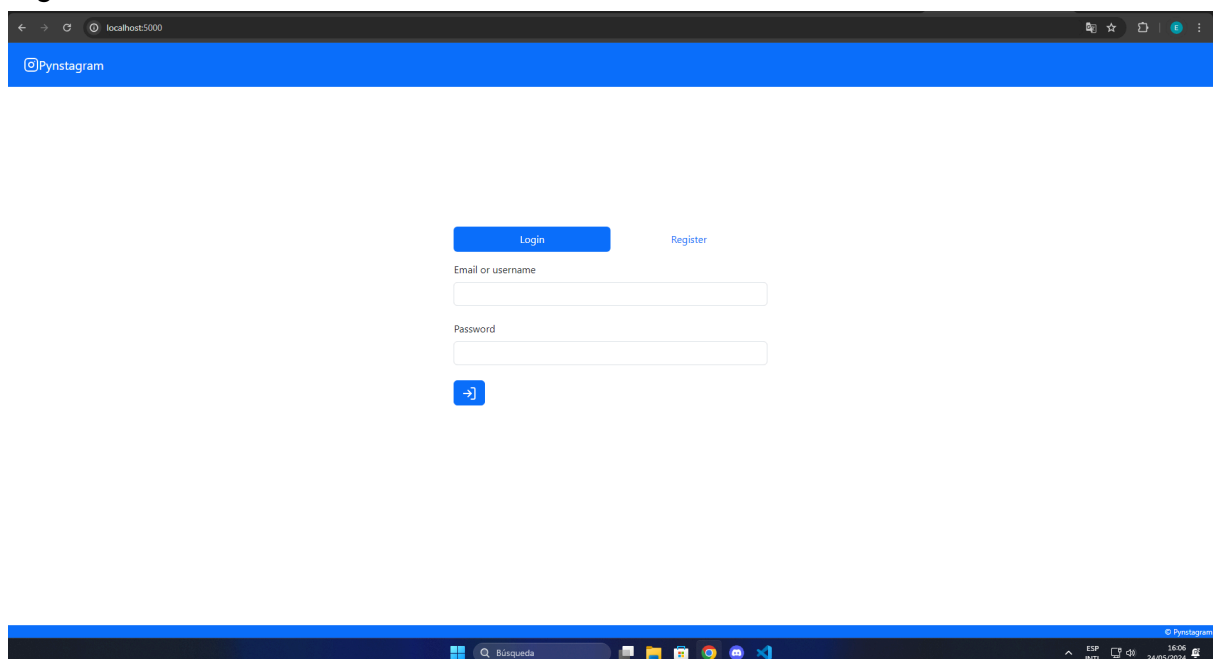
Antes de proceder con la explicación de las funcionalidades más importantes (no me centraré tanto en describir el código línea por línea, para eso ya está el código fuente, si no en describir la funcionalidad general, es decir, que hace), voy a hacer una pequeña overview del proyecto.

Importante: Para ejecutar el programa debemos, en primer lugar **desde la raíz del proyecto** levantar la bd con redis-server &, y acto seguido iniciar el programa con python src/app.py. **MUY IMPORTANTE DEBIDO A LAS RUTAS USADAS PARA LAS IMÁGENES, DE NO INICIARSE ASÍ DARÁ ERROR AL INTENTAR SUBIR UNA IMAGEN O MODIFICAR LA DE PERFIL.**

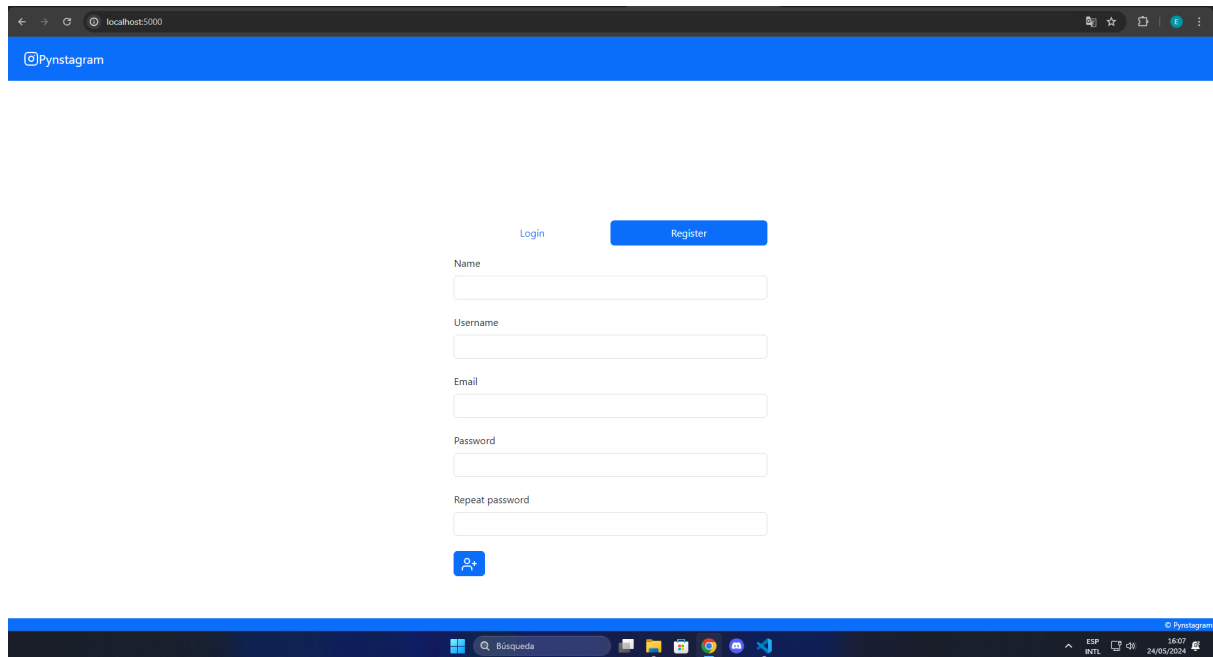
Overview

Cuando iniciamos el programa nos encontramos con una pantalla en la que podremos iniciar sesión o registrarnos en el caso de que no exista nuestro usuario.

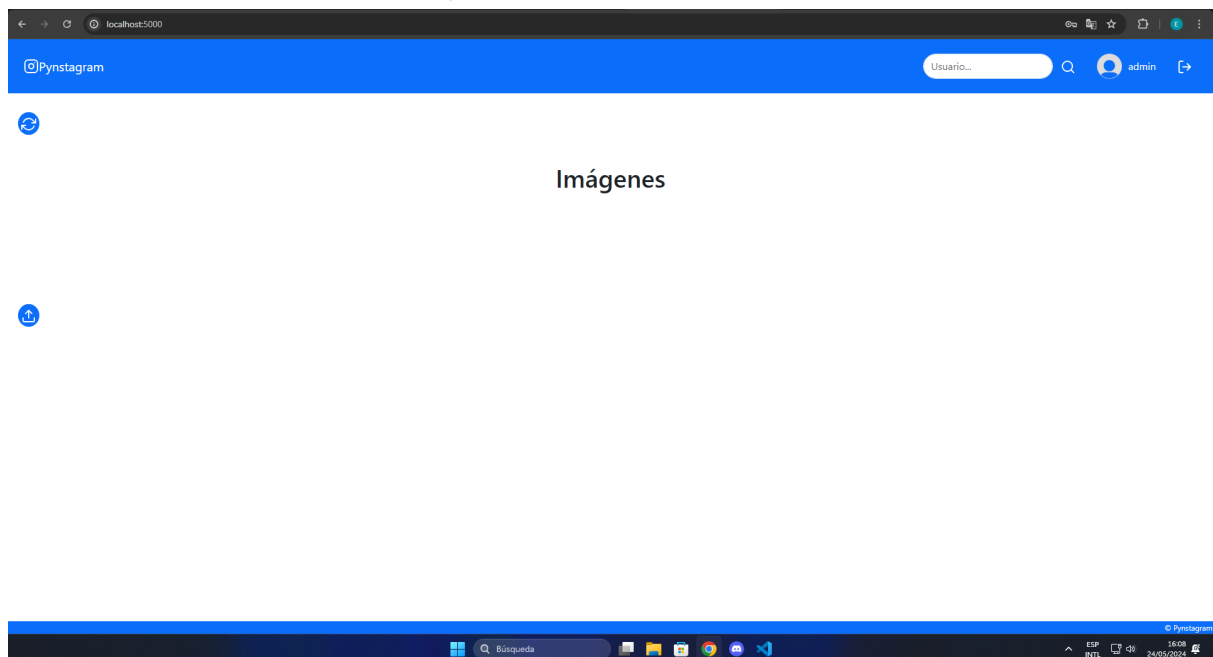
Login:



Register:



Una vez iniciamos sesión con un usuario válido nos encontramos con la pantalla principal, que es donde podremos subir imágenes:



Vamos a destacar 6 funcionalidades de esta pantalla:

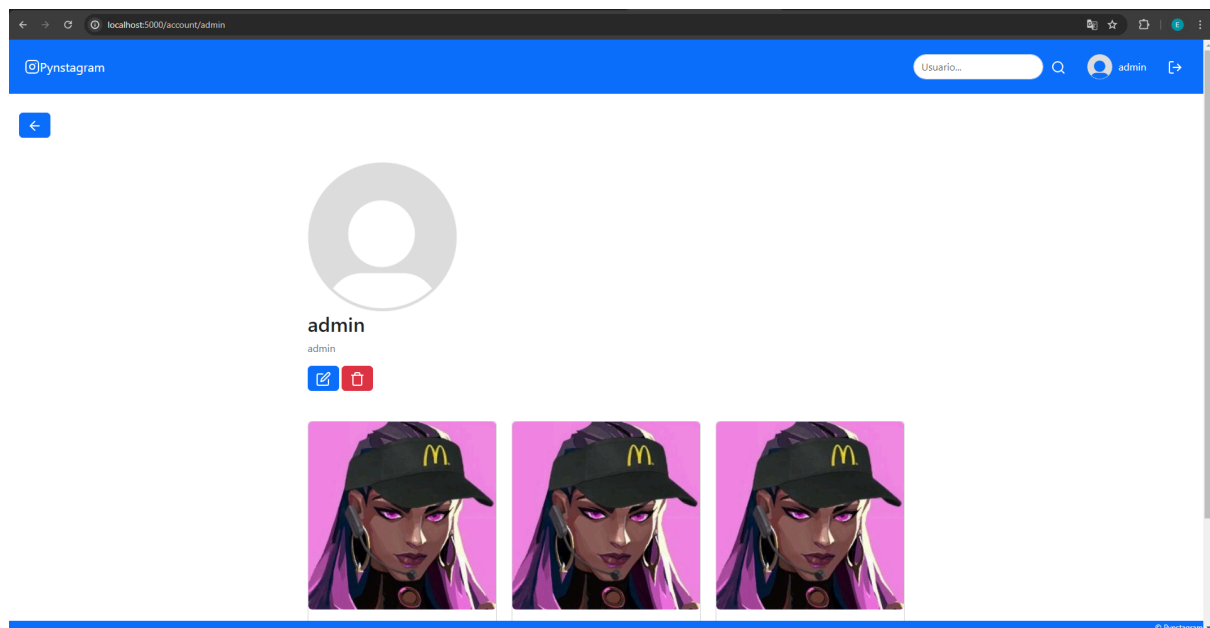
1. El botón situado arriba a la izquierda nos permite recargar la página, para por ejemplo si un usuario ha subido una fotografía desde otro ordenador y nosotros queremos actualizarla, no tener que hacer uso del refresh del navegador.
2. El botón situado debajo del anterior es el botón que nos permitirá subir las fotografías.
3. Si pulsamos en el logo de Pynstagram del header nos llevará a la página principal siempre, es decir, a esta página.

4. Existe un buscador de usuarios, si bien es necesario conocer el nombre de usuario de la persona que queremos buscar.
5. Si pulsamos en el icono de nuestra fotografía de perfil o en nuestro nombre de usuario, nos llevará a nuestra pantalla de cuenta, desde donde podemos hacer numerosas funcionalidades definidas a continuación.
6. En último lugar existe un botón para desloguearse.

Vamos a ir detallando funcionalidades.

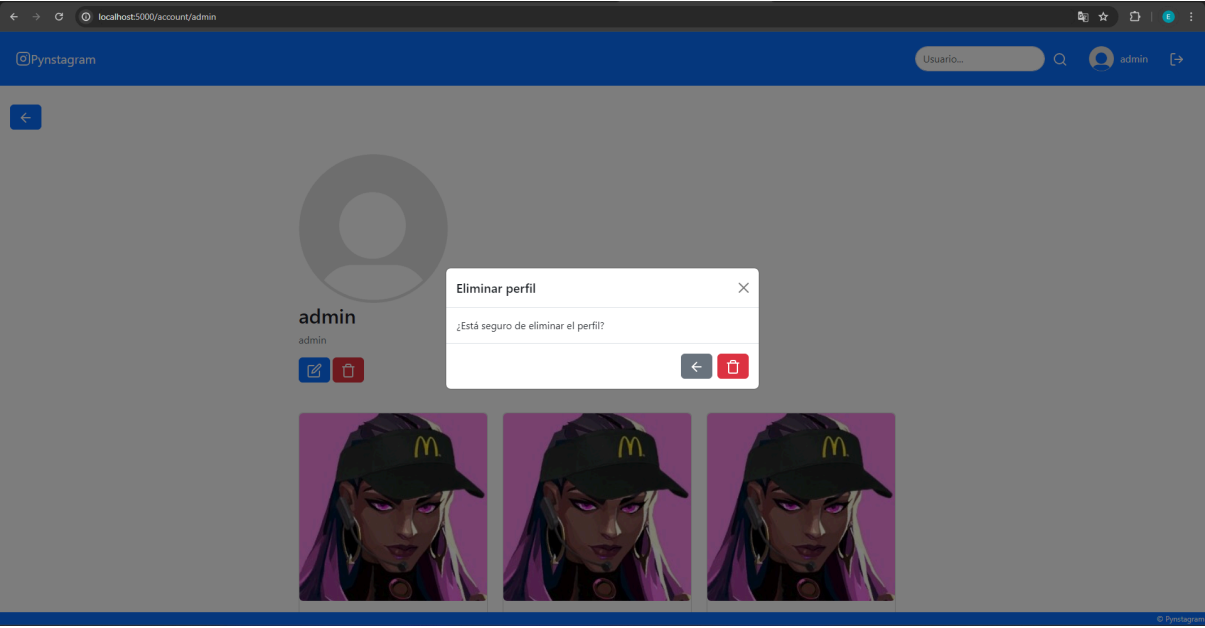
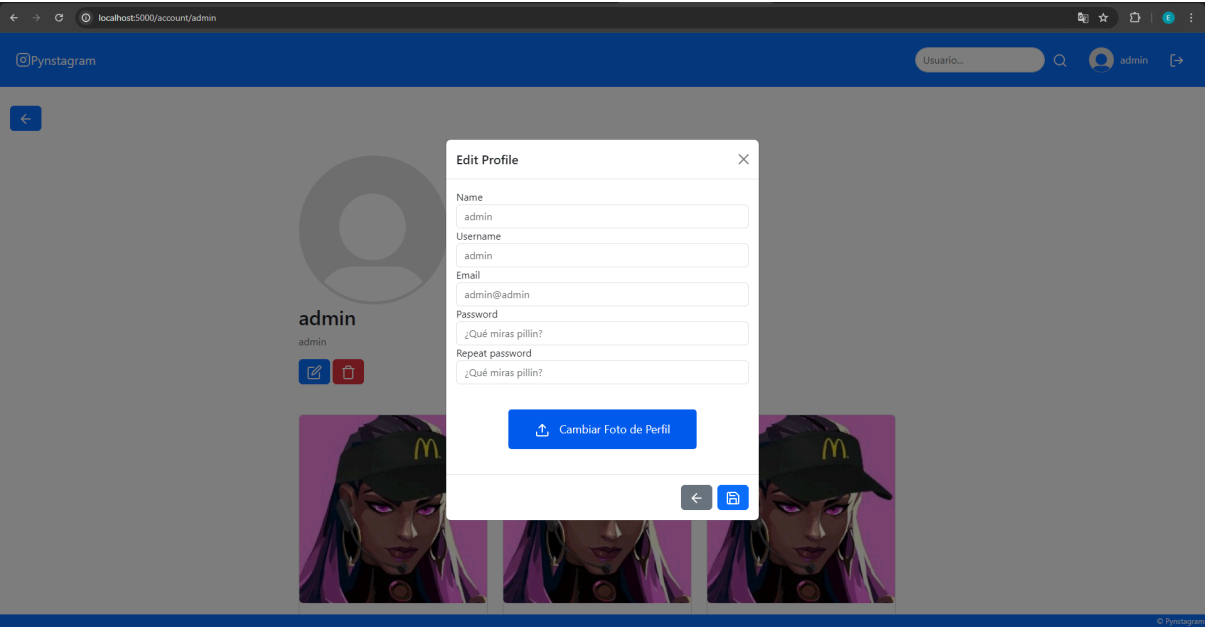
Usuarios

Al entrar en la pantalla de cuenta de un usuario podemos ver su fotografía, nombre, nombre de usuario y las fotografías que el mismo tiene subidas. Si el usuario es nuestro propio usuario se nos permite tanto modificar/borrar nuestro perfil como modificar/borrar las fotografías subidas. Cabe destacar que siempre que pulsemos sobre una fotografía se nos va a abrir un modal con los comentarios y una vista más agradable que la cuadrícula de fotografías.



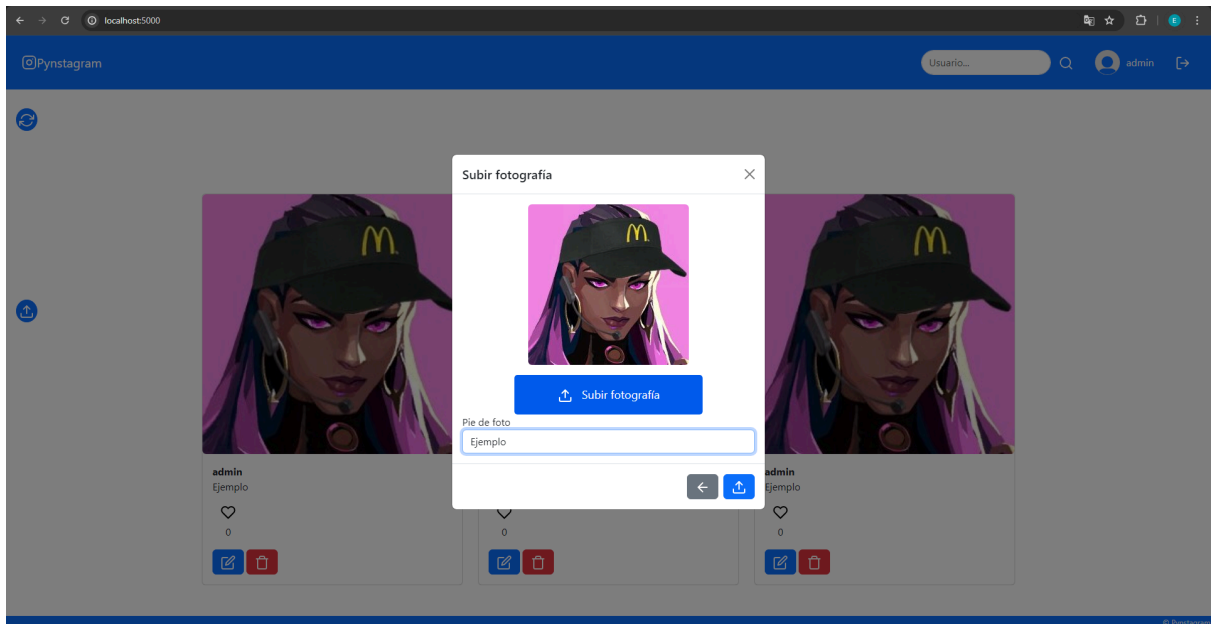
Como podemos observar al ser nuestro usuario vemos los dos botones que he mencionado anteriormente. En caso de pulsar el botón para editar se nos abrirá un modal para editar nuestro usuario, si no queremos editar algo simplemente lo dejamos vacío.

En caso de borrar un usuario conlleva “mucho” trabajo, ya que se borra todo lo que tiene que ver con dicho usuario: fotografías, comentarios, likes, y a mayores en el resto de usuarios se tiene que ver reflejado, ya que deben de borrarse los indicios de haber likeado o comentado fotografías del usuario que se va a borrar.



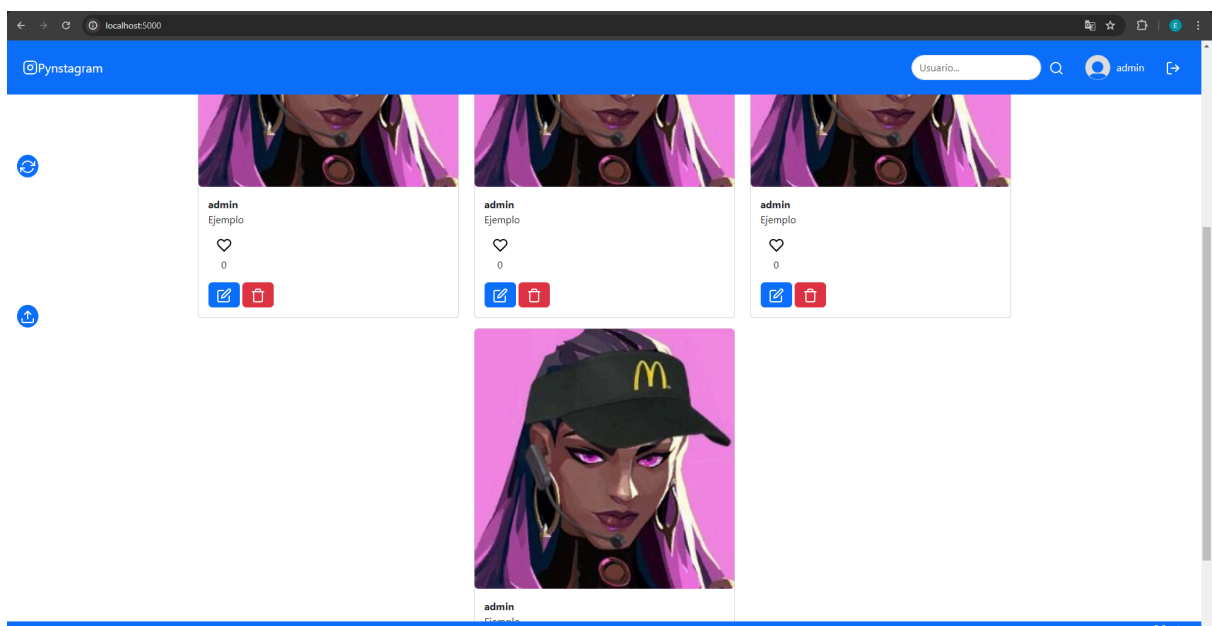
Imágenes

Para subir una imagen debemos, desde la pantalla de índice, la pantalla principal, presionar el botón que se ha comentado con anterioridad, en dicho caso se nos abrirá un modal para introducir la fotografía y un pie de foto.



Cuando subimos una imagen esta se sube a una ruta del proyecto, en este caso a `src/static/uploaded_photos`. Cuando se modificaba la fotografía de perfil del usuario se subía a `src/static/profile_pictures`. Cabe destacar que si se elimina una fotografía esta se borra de la carpeta de fotografías, al igual que cuando un usuario cambia una foto de perfil se elimina o también se hace al eliminar al usuario, que se borran todas las fotografías y la imagen de perfil.

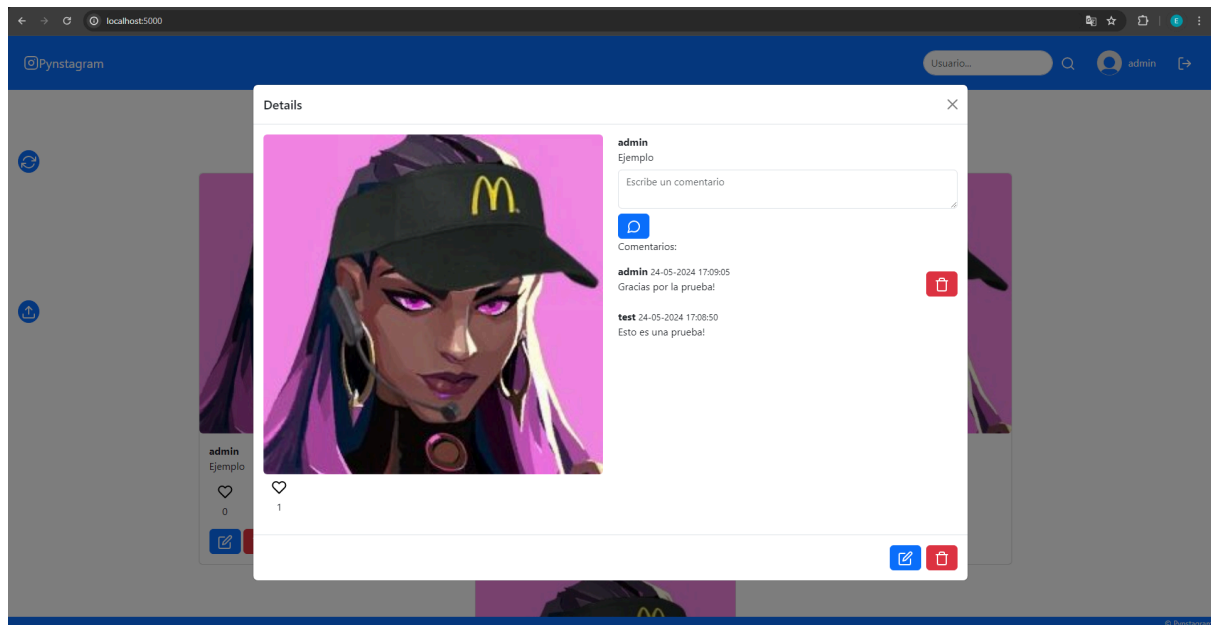
Una vez subida la fotografía estará disponible en la cuadrícula de fotografías.



Como podemos observar una fotografía puede editarse o borrarse, en caso de editarla únicamente se nos permite cambiar el pie de foto, y para borrarla se nos pide confirmación, ambos en un modal.

Si nos fijamos también tenemos un icono de like, en el que si hacemos click daríamos like a dicha fotografía.

Así mismo si abrimos la fotografía podemos tener una vista más amigable con los comentarios a su lado derecho:



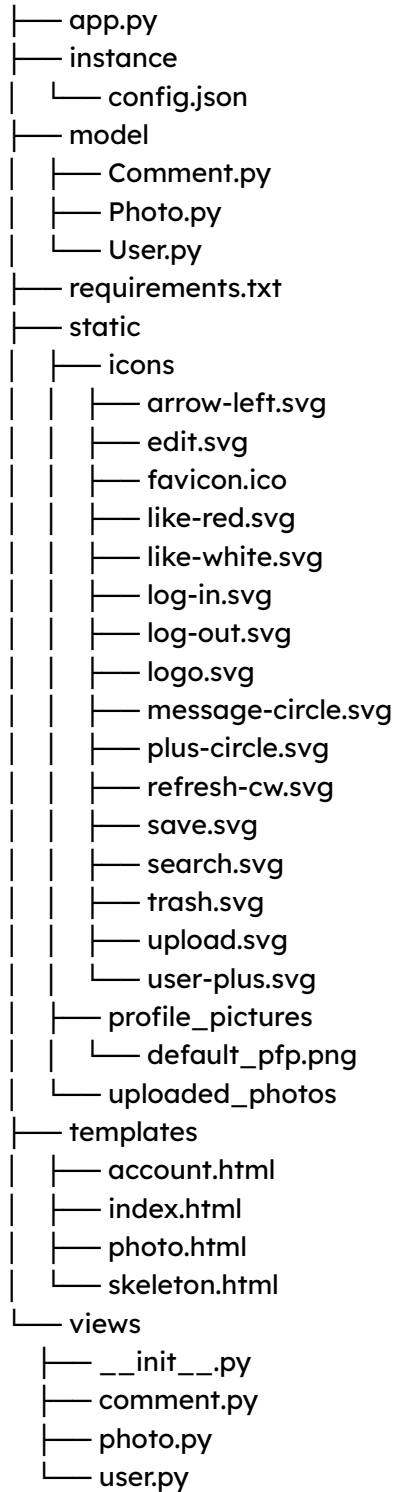
Comentarios

Como se ve en la fotografía superior, los comentarios se relacionan a cada fotografía y, en caso de ser el comentario del usuario actual, este puede eliminarlo sin ningún problema.

Código

En cuanto a la estructura del código de nuestro proyecto vamos a destacar lo siguiente:

src



Iré explicando los puntos más importantes de cada archivo poco a poco.

requirements.txt

Se proporciona un fichero de requirements con el que podemos instalar las dependencias necesarias.

instance/

En ella se encuentra el archivo de configuración, con la clave necesaria para inicializar algunos módulos.

static/

No tiene misterio alguno, simplemente se guardan los iconos que se utilizan para la representación web y las imágenes tanto subidas por los usuarios como las fotografías de perfil.

templates/

Almacena los archivos html necesarios para mostrar la página, se compone de:

- Un skeleton, esqueleto de todas las templates.
- Un index, que es la página principal, donde se muestran las imágenes.
- Un account, que se corresponde con la página en la que se muestra la cuenta.
- Un photo, que es el encargado de presentar las fotografías en su modal,... .

Y por último las dos más importantes y en las que nos vamos a centrar:

model/

En esta carpeta están las clases que hacen referencia a las entidades, en mi caso la clase User, utilizada para instanciar usuarios, la clase Photo, utilizada para instanciar fotografías, y la clase Comment, para los comentarios.

Comment
msg time username
find(srp, username: str, msg): from_dict(cls, data) get_safe_id(srp)

Photo
caption comments liked_by likes url username
add_comment(comment) find(srp: str, url): get_comments(srp) get_safe_id(srp) like(user) remove_comment(comment)

User
commented_photos email liked_photos name profile_picture uploaded_photos username
add_commented_photo(photo) add_liked_photo(photo) chk_pswd(other_pswd) current() find(srp: str, username): get_id() get_safe_id(srp) modify(name, username, email, pswd) remove_commented_photo(photo) set_email(email) set_name(name) set_password(pswd) set_profile_picture(profile_picture) set_username(username) upload_photo(photo)

Como podemos observar la clase Comment almacena el mensaje que se postea, la fecha y hora a la que se hace y el usuario que lo ha subido.

La clase Photo almacena la url de la fotografía, el pie de foto, el número de likes, la lista de personas que le ha dado like, la lista de comentarios y el usuario que la ha postado.

El Usuario es un poco más complejo, almacena el nombre, nombre de usuario, email, url de la foto de perfil, lista de urls de las fotos subidas, lista de fotos likeadas y lista de fotos comentadas.

No voy a entrar en detalle acerca de los métodos ya que son autodescriptivos por su nombre.

views/

La carpeta views almacena los 3 templates para las entidades definidas, esto es, las acciones que se pueden realizar sobre dichas entidades, y que serán estudio de una explicación posterior y posiblemente de una serie de diagramas para entender su funcionamiento.

- comment.py se ocupa de las acciones a realizar con los comentarios, esto es, añadir comentarios (add) o borrarlos (delete).
- photo.py se ocupa de las acciones a realizar con las fotografías, esto es, añadirlas(add), modificarlas(modify), borrarlas(delete) y likearlas(like).
- user.py se ocupa de las acciones a realizar con los usuarios, esto es, añadir usuarios(add), modificarlos(modify) o borrarlos(delete).

Por último lugar voy a hablar acerca del archivo app.py, que es el archivo principal en el que se definen todos los componentes y que contiene las siguientes acciones:

- Logear usuarios.
- Desloguear usuarios.
- Cargar la página principal, pasándole los datos necesarios (usuario, sirope para la DB, lista de imágenes)

Cabe destacar que no se proporciona un esquema de la bd como tal, ya que es una base de datos no relacional que lo que guarda vendría a ser algo similar al diccionario del objeto, por ejemplo para un comentario guarda:

“msg” : “Ejemplo de mensaje”,
“time” : “Fecha”,
“usrname” : “Ejemplo de usuario”

Así con todas las entidades, y la relación entre ellas es implícita, ya que almaceno por ejemplo en un usuario una lista de url fotos subidas, con lo que puedo acceder al objeto Photo de la BD.

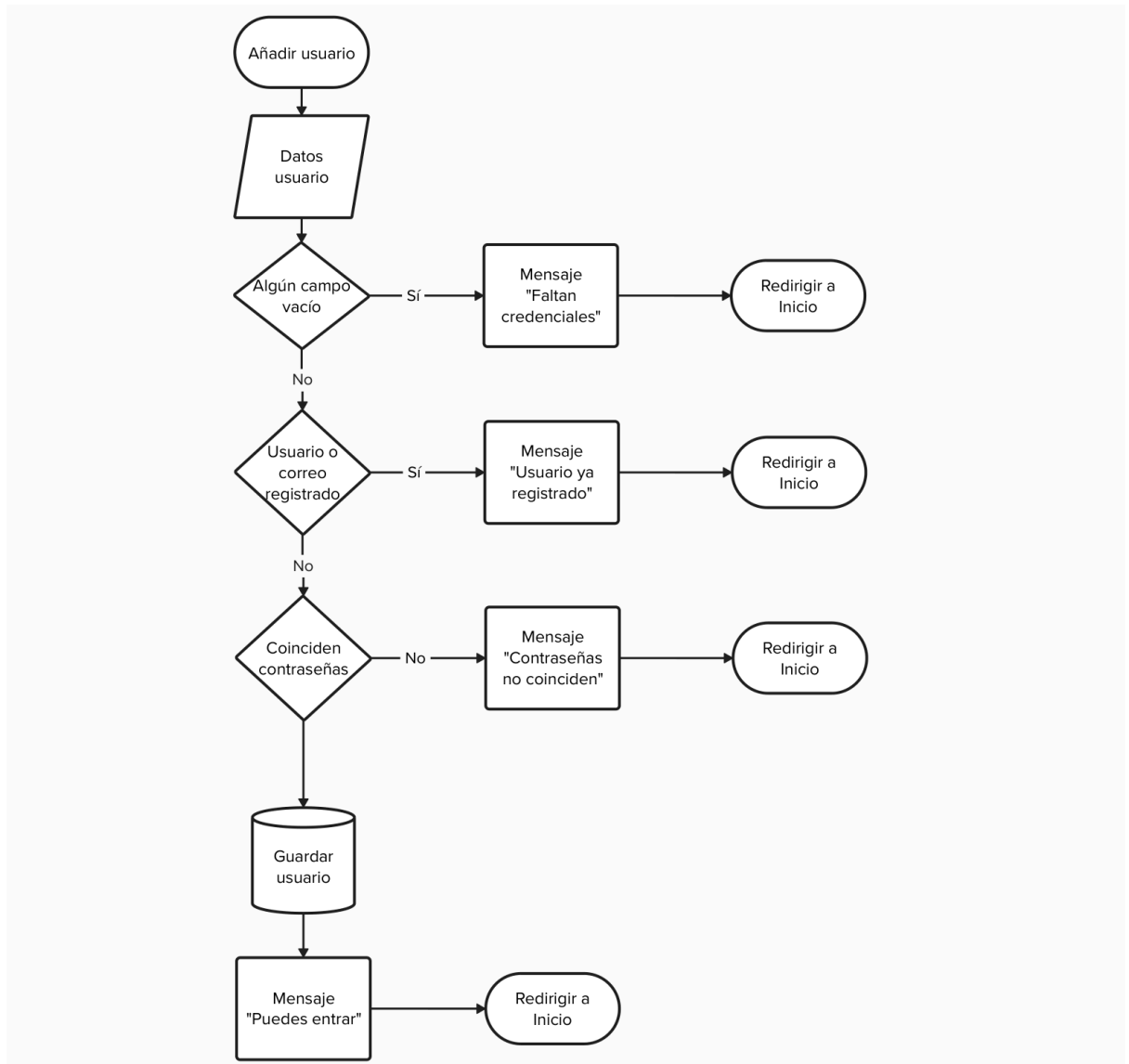
Funcionalidades

Empezaré por los usuarios, pero dejaré el delete para el final.

Añadir usuarios

Cuando un usuario no está registrado y se registra se realizan los siguientes pasos:

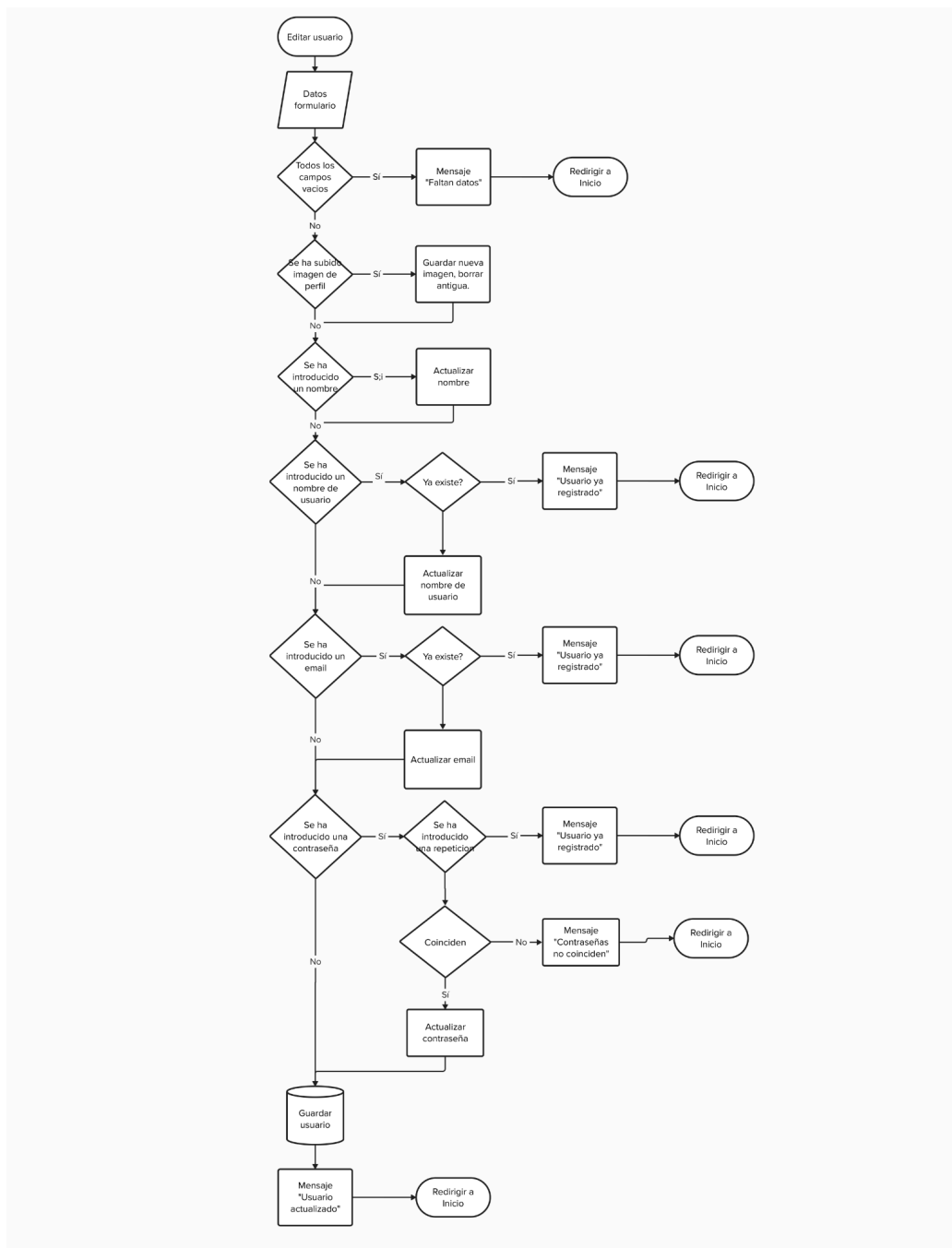
- En primer lugar se obtienen del formulario que ha rellenado todos los datos necesarios: nombre, email, nombre de usuario, contraseña y la contraseña repetida.
- Se comprueba que se hayan rellenado todos los campos, de no ser así se lanzaría el mensaje de que Faltan credenciales.
- Se comprueba que el usuario no existe ya en el sistema, esto es, que el email y el nombre de usuario no coincida con ninguno ya existente, en ese caso se muestra el mensaje de usuario ya registrado.
- Se comprueba que las contraseñas coincidan si no coinciden se muestra el mensaje correspondiente.
- Si todo ha ido correctamente, se crea el nuevo usuario y se introduce en la base de datos gracias a sirope.



Editar usuarios

Para editar al usuario se realizan los siguientes pasos:

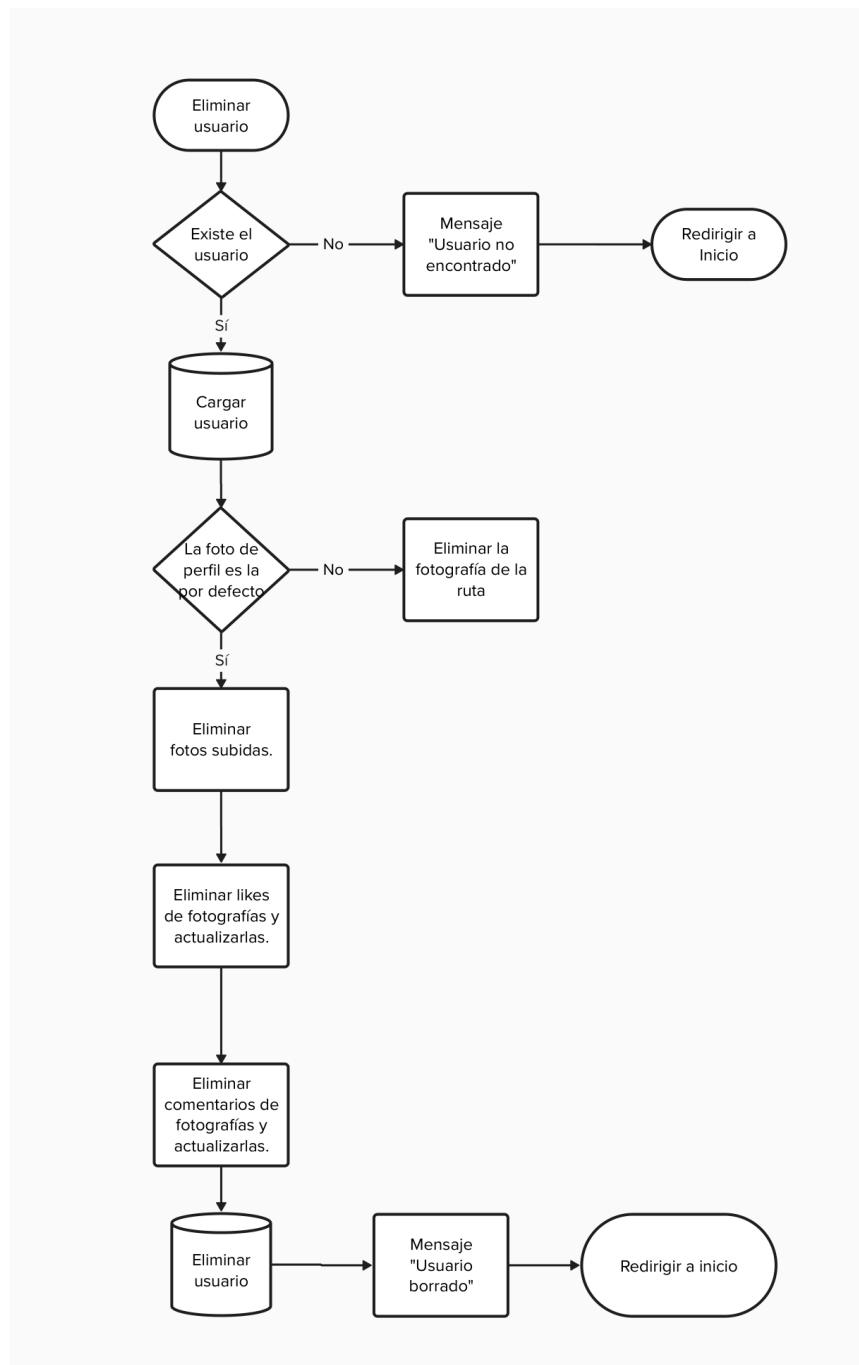
- Se obtienen los datos del formulario.
- Si no se ha rellenado ningún campo se muestra el mensaje de que no hay nada que modificar.
- Por cada campo que se ha introducido, se comprueba que sea válido.
- En el caso de la imagen de perfil, si se cambia y no estaba la imagen por defecto se borra la que está guardada.
- En el caso de modificar la contraseña debe de introducirse tanto la contraseña como la contraseña repetida y ser iguales.
- Si todo ha ido correctamente se guarda el usuario en la base de datos, se muestra el mensaje de que se ha modificado correctamente y se redirige a inicio.



Borrar usuarios

Para eliminar a un usuario se realizan los siguientes pasos:

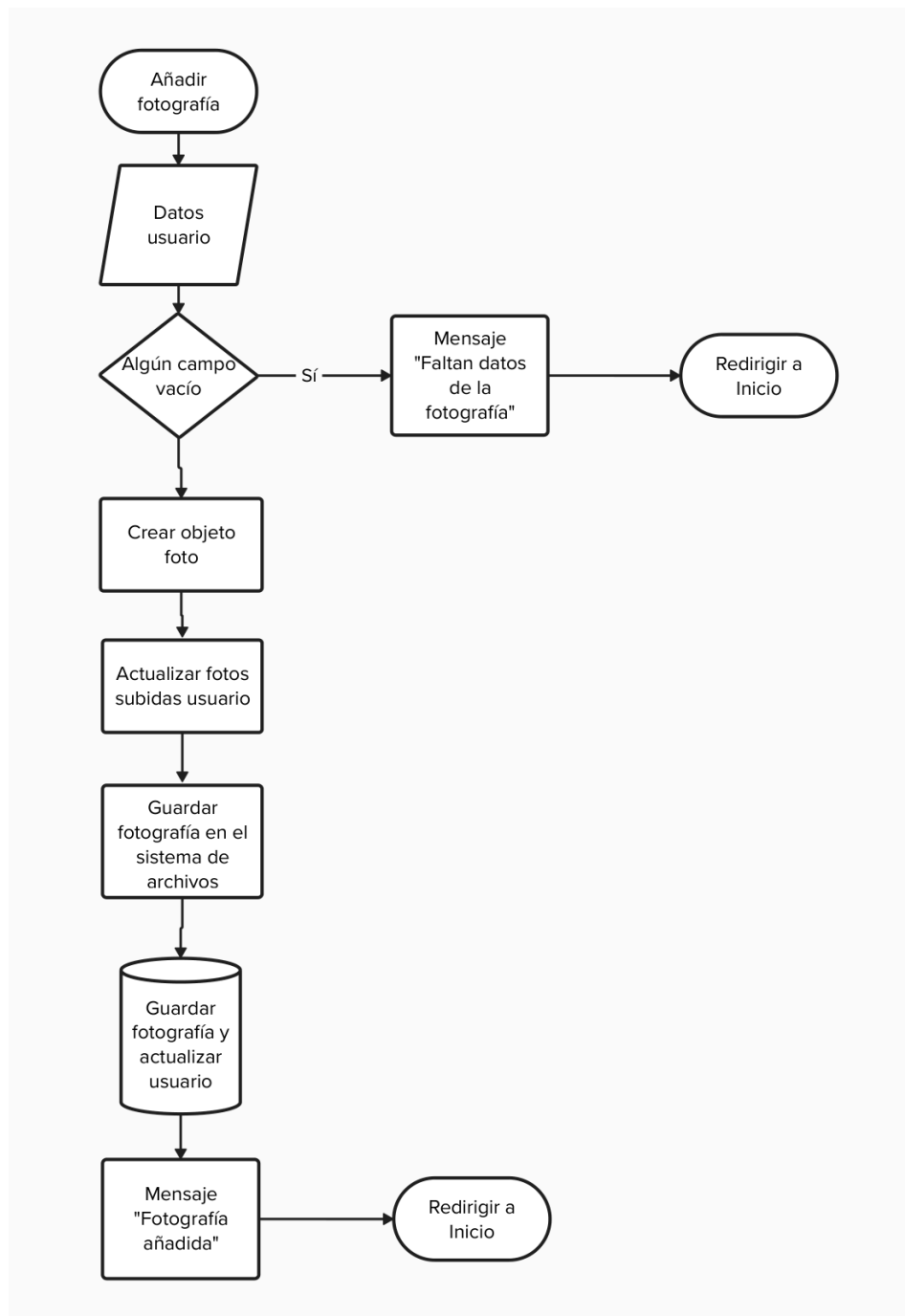
- Se comprueba que existe en la base de datos, si no se muestra el mensaje “Usuario no encontrado”.
- Si existe y la fotografía que tiene de perfil no es la por defecto se elimina.
- Para cada fotografía que tiene subida, se carga de la base de datos y se elimina, tanto de la base de datos como de la carpeta del proyecto.
- Para comentario que ha realizado, se carga la foto en la que ha realizado el comentario y se elimina.
- Para cada fotografía que ha likeado, se carga y se elimina el like.
- Se borra el usuario de la base de datos, se muestra el mensaje de “Usuario eliminado” y se redirige a inicio.



Añadir imágenes

Para subir una fotografía se realizan los siguientes pasos:

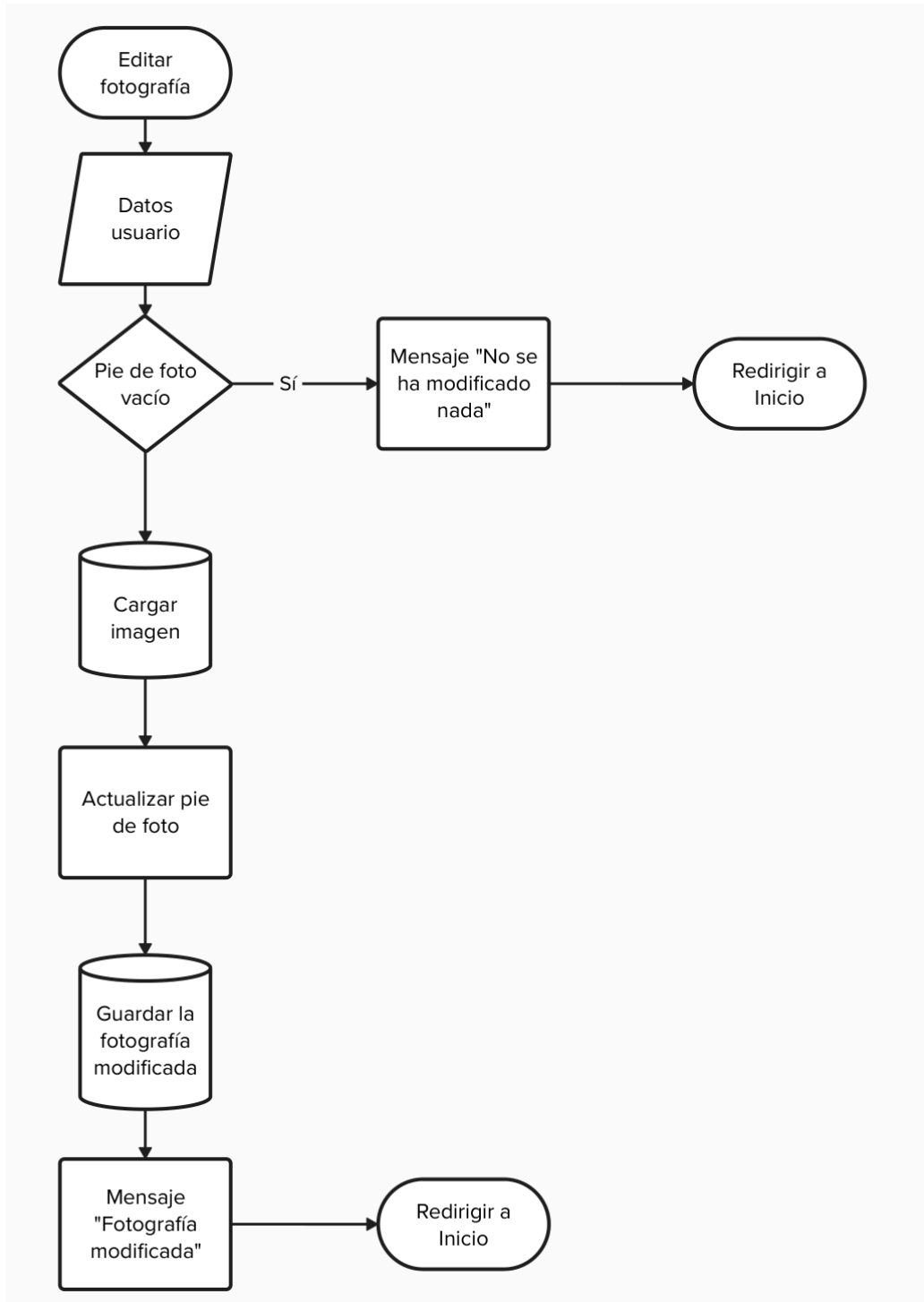
- Se comprueba si se ha incluido la fotografía y el pie de foto en el formulario, si no es así se hace el display del mensaje “Faltan datos de la fotografía” y se redirige a inicio
- Se almacena la foto en la carpeta de fotos subidas.
- Se crea una instancia de Photo, teniendo en cuenta el usuario actual, que la ha subido.
- Se indica al usuario que ha subido esa foto, para que lo almacene en su lista de fotografías subidas.
- Se guarda en la base de datos la fotografía y el usuario actualizado, haciendo el display de “Fotografía añadida” y redirigiendo a inicio.



Modificar fotografía

Lo único que se puede modificar de una fotografía ya subida es el pie de foto:

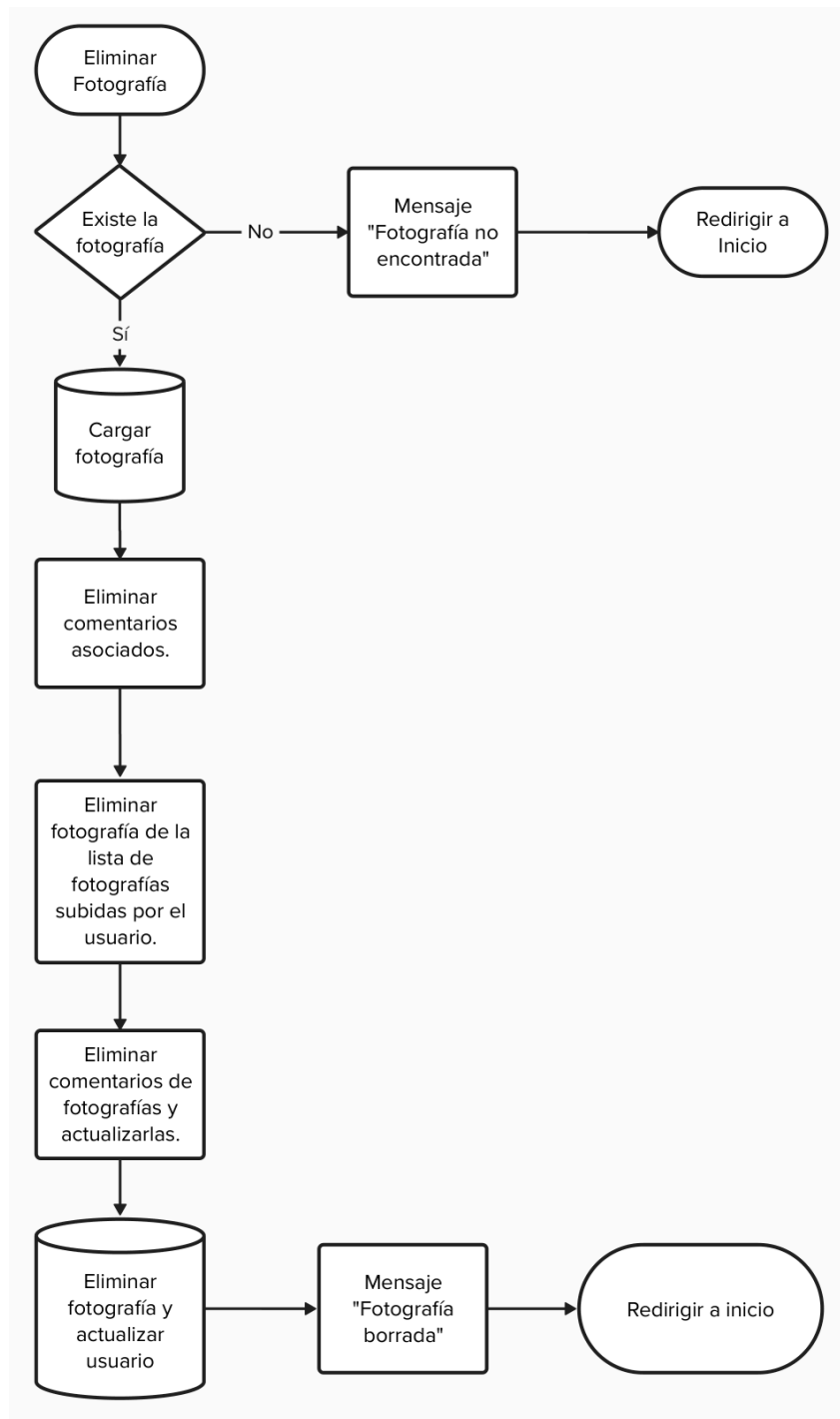
- Se obtienen los datos del formulario, si no se ha proporcionado un pie de foto se hace el display del mensaje “No se ha modificado nada” y se redirige a Inicio.
- Se carga la imagen de la base de datos.
- Se actualiza el pie de foto.
- Se guarda la imagen actualizada en la base de dato, se hace el display de “Fotografía modificada” y se redirige a inicio.



Borrar fotografía

Para borrar una fotografía se hace lo siguiente:

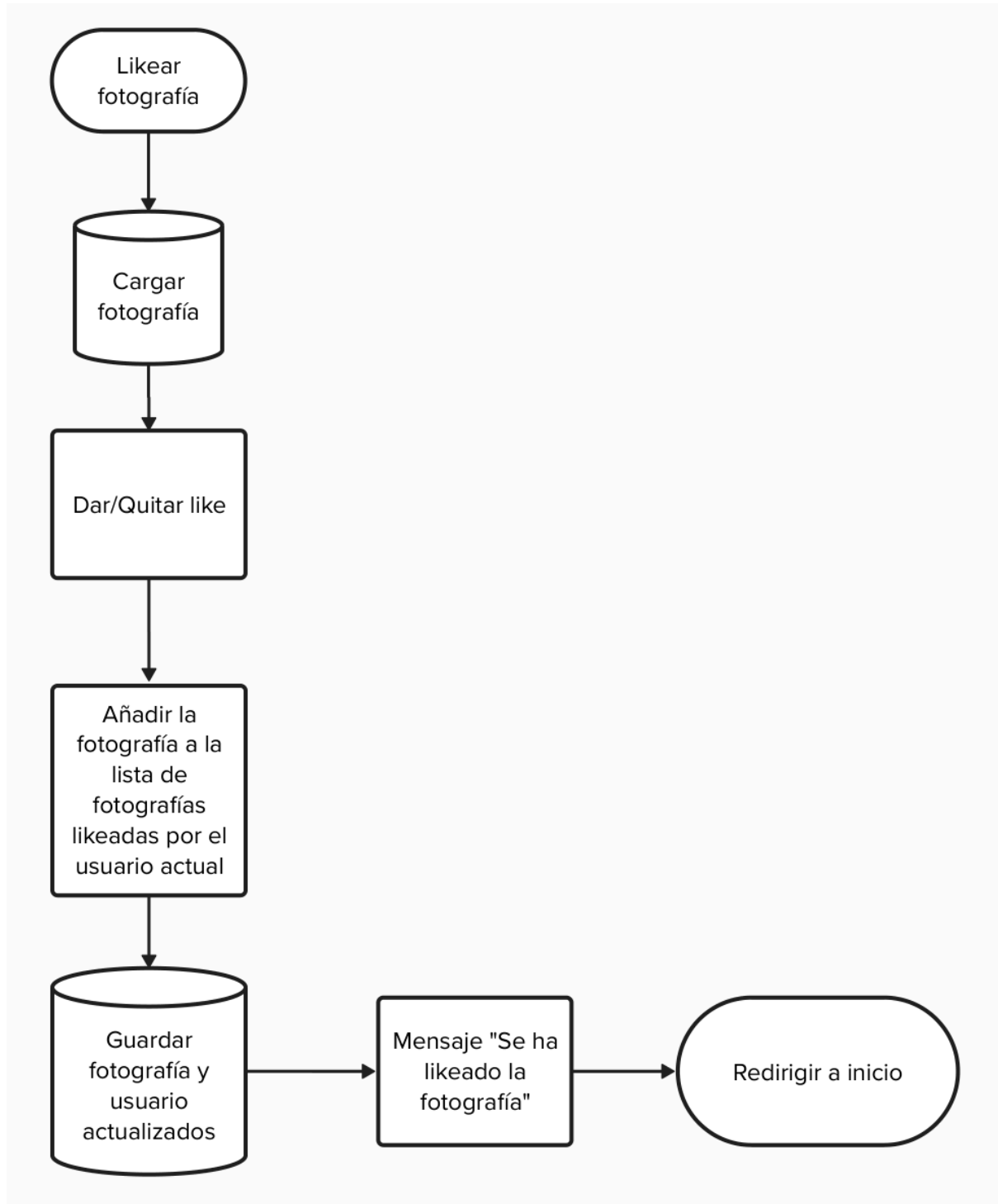
- Se comprueba que existe en la base de datos, si no es así se muestra el mensaje “Fotografía no encontrada”.
- Se carga la fotografía de la base de datos.
- Por cada comentario que tiene la foto se elimina.
- Se elimina la fotografía de la lista de fotografías subidas por el usuario.
- Se borra la fotografía del sistema de archivos.
- Se actualiza el usuario en la base de datos.
- Se borra la fotografía de la base de datos y se muestra el mensaje “Fotografía borrada” y se redirige a Inicio.



Dar like a una fotografía

Para dar like a una fotografía:

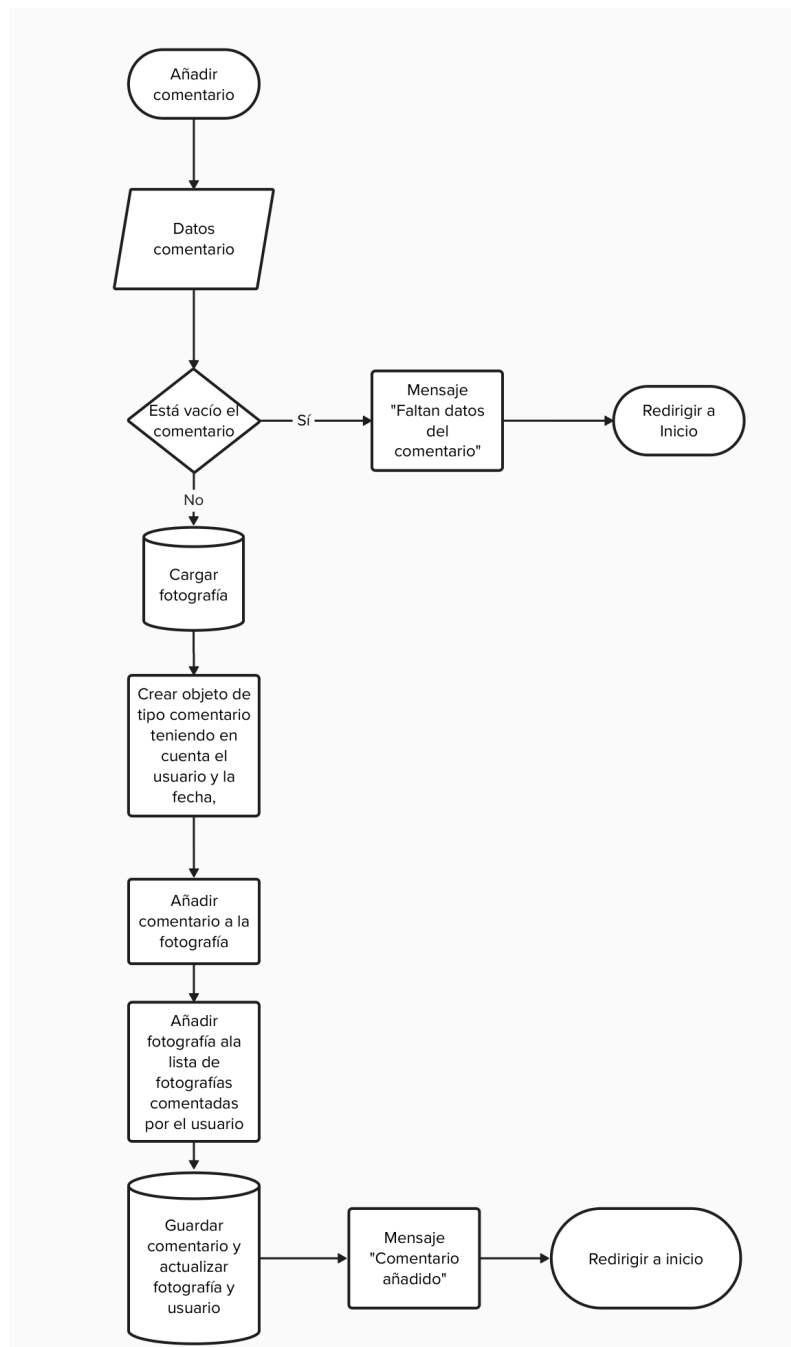
- Se carga la fotografía de la base de datos,
- Se da/quita like a la fotografía(a nombre del usuario actual), si está sin likear se le da y si no se le quita.
- Se añade la fotografía a la lista de fotografías likeadas por el usuario actual.
- Se actualiza la fotografía y el usuario actual en la base de datos.



Añadir comentario

Para añadir un comentario se realiza lo siguiente:

- Se recogen los datos del formulario.
- Si no hay comentario, se muestra el mensaje “Faltan datos del comentario” y se redirige a Inicio.
- Se carga la foto de la base de datos.
- Se crea una nueva instancia de comentario, teniendo en cuenta el usuario actual y la fecha del comentario.
- Se añade el comentario a la foto.
- Se añade la fotografía comentada a la lista de fotografías comentadas del usuario.
- Se guarda el comentario en la base de datos y se actualizan la fotografía y el usuario, se visualiza el mensaje “Comentario añadido” y se redirige a Inicio.



Eliminar comentario

Para eliminar el comentario:

- En primer lugar se comprueba si existen la fotografía y el comentario, de no ser así se visualiza el mensaje “Error, comentario no borrado” y se redirige a Inicio.
- Se carga el comentario y la fotografía.
- Se elimina el comentario de la fotografía.
- Se elimina la fotografía de la lista de fotografías comentadas por el usuario.
- Se elimina el comentario de la base de datos, se actualiza el usuario y la fotografía y se visualiza el mensaje “Comentario borrado” redirigiendo a Inicio.

