Nero

Nero

Diagramas de secuencia

# **Registro de Cuenta** Para registrarse en la aplicación, es necesario que el usuario presione el botón de registro, tras esto deberá introducir sus datos personales, al introducirlos deberá volver a presionar el botón de registro, con lo que el sistema guardará a este usuario junto a sus datos en la base de datos.

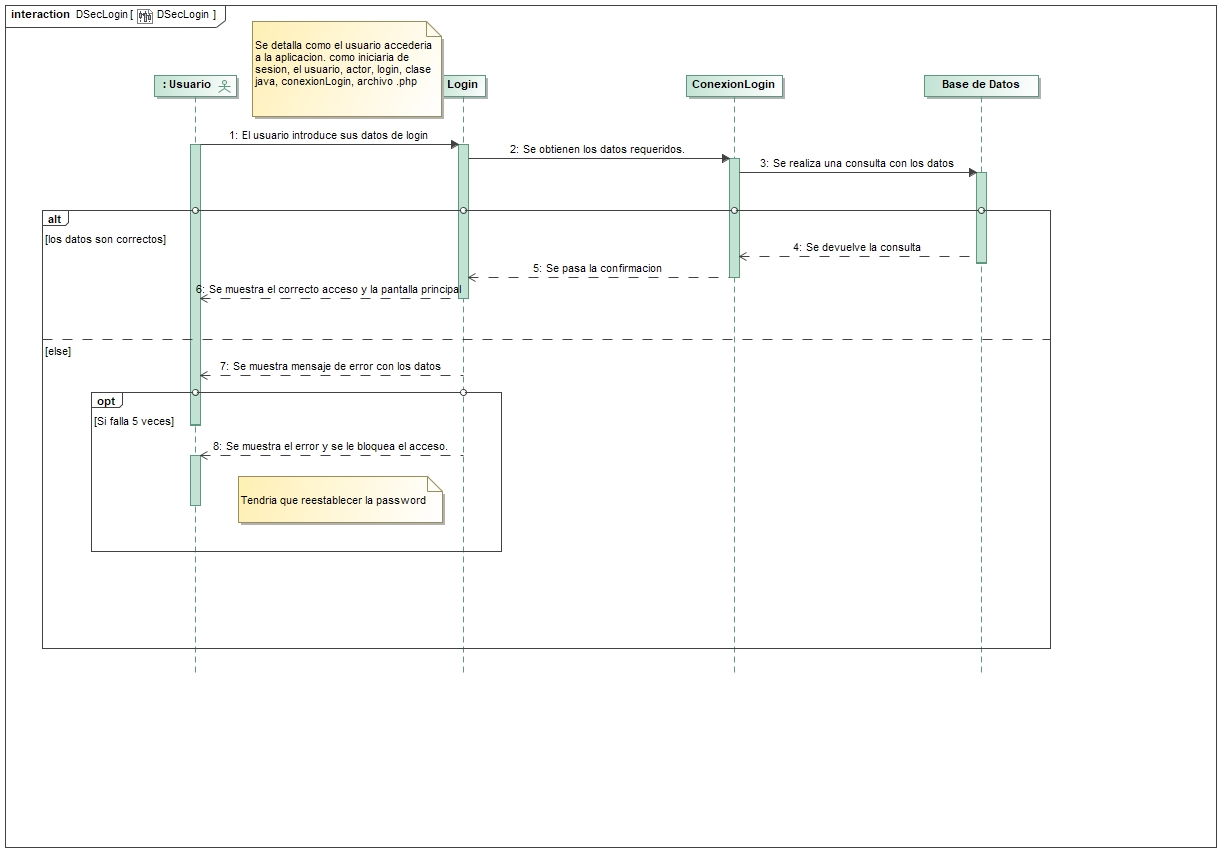
# Si el correo electrónico introducido o la contraseña son incorrectas, se le mostrará al usuario un mensaje advirtiendo de ello

# 

**Inicio de sesión**

Se detalla las secuencias con respecto al login del usuario en la aplicación. El usuario introduciría los datos de nombre o email y la password. Una clase .java recoge los datos al igual que los envía a un archivo .php donde esta escrita la consulta a realizar para obtener el acceso por parte del usuario. La respuesta a la consulta la obtiene el .php que se la manda a la clase .java esta se encarga de mostrar por pantalla la correspondiente respuesta de la base de datos. Si todo marcha correctamente la base de datos le devolverá al .php que ha sido un éxito, esta a su vez se lo envía a la clase de java que mostrará que los datos han sido correctos y llevará al usuario a la interfaz principal de la aplicación. En caso negativo, se le mostrará al usuario el error en los datos.

Existe una excepción en este caso y es cuando el numero de intentos supera lo permitido, entonces se bloquea el acceso a la cuenta del usuario y tendría que reestablecer la password, de todo esto se encarga la clase java.

La clase java en este caso es “Login” y el archivo php es “ConexionLogin”. 

# **Editar perfil**

# El usuario accede al menú de configuración, y dentro de él selecciona la opción de editar perfil, tras esto, la interfaz de solicita al sistema los datos modificables del usuario, al recibirlos, se les muestra al usuario, y se le permite modificarlos. Al modificarlos, presionará el botón de guardar cambios, por lo que el sistema guardará dichos cambios, y se le mostrará al usuario un mensaje avisando de que sus cambios han sido guardados.

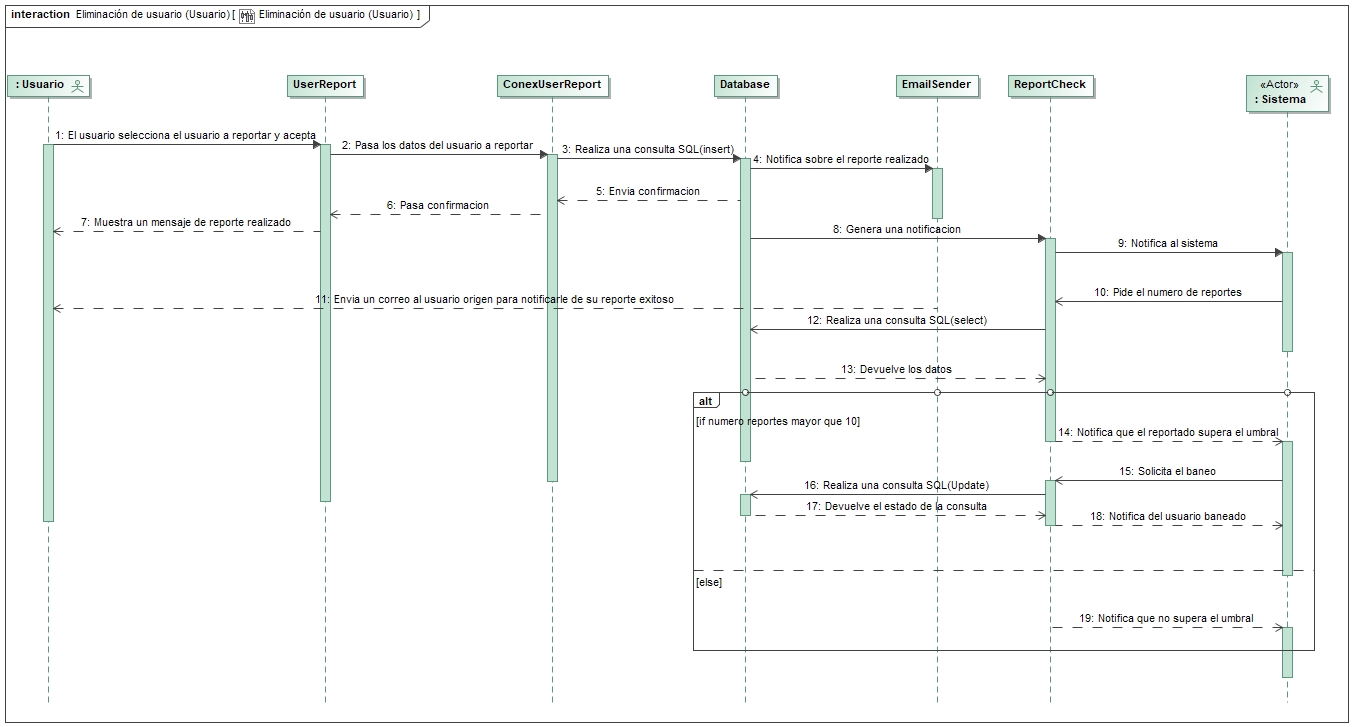
# 

# **Reporte y baneo de usuario**

# Nuestra aplicación debe controlar cada usuario y para ello se ha implementado un sistema de reportes de usuarios. El sistema consiste en que un usuario cualquiera tiene la opción de reportar a otro usuario si este no cumple las normas de la aplicación. Posteriormente el sistema se encarga de gestionar el baneo del usuario reportado.

El usuario reporta al otro usuario que desee e indica su motivo, el sistema efectúa el reporte sobre el usuario sumándole 1 a la casilla baneado de dicho usuario en la base de datos, tras esto, un .php envía un correo al usuario con su reporte.

Posteriormente se notifica al sistema del reporte realizado, y el sistema solicita ver el número de reportes de dicho usuario, si el número de reportes es mayor a 10, el sistema solicitará el baneo de dicho usuario, efectuándose automáticamente.



# **Uso del mapa**

# Se describe las secuencias de interacción entre el usuario y el mapa dentro de la aplicación.

# El usuario abriría el mapa, a lo cual el sistema le pediría que active su ubicación, mientras no active su ubicación no podría usar el mapa. Si el usuario activa su ubicación automáticamente se realiza una consulta a la base de datos sobre las fuentes cercanas a la ubicación de dicho usuario. El objetivo de esto es no saturar el mapa con todas las fuentes existentes ya que sería difícil moverse por el si existe un gran numero de fuentes. Una vez hecha la consulta se muestran aquellas fuentes cercanas y el usuario pasaría a interactuar con el mapa moviendo, haciendo zoom, etc. Para realizar esto correctamente intervienen además de la base de datos y el usuario, una clase java y un archivo php para realizar la consulta. En la clase java se detalla todo lo que sería pedir la ubicación y capturarla, movimiento del mapa, etc. El archivo php solo sirve para realizar la consulta y devolver resultados.

# En este caso concreto el .java sería “mapsactivity” y el php “conexionmapsactivity”.

# 

# **Registro de Fuente**

# El usuario presiona el botón de añadir fuente, y se encuentra en una interfaz que le pide añadir el nombre, fotos, ubicación y horario de ésta. Si al seleccionar la ubicación la fuente se encuentra a más de 100 metros de la localización del usuario, se le informará de que la ubicación proporcionada no es válida, si se encuentra a menos de 100 metros, se guardarán las características de ésta fuente en la clase Fuente de la base de datos, y se le mostrará al usuario el mapa actualizado con su fuente acompañado de un mensaje informando de que la fuente ha sido añadida correctamente.

# 

# **Comentar Fuente**

# El usuario escribe un comentario en la interfaz de la fuente que quiere comentar, el sistema revisa si ese comentario excede el límite de 1000 caracteres o no, si lo excede le pide al usuario modificar su comentario, si no, el comentario se guarda en la clase comentario, asociada a su fuente. Al completar estos pasos el sistema le devuelve al usuario la interfaz de la fuente actualizada para que pueda ver que su comentario ha sido realizado.

# 

# **Valoración Fuente**

El usuario accede a la fuente que desee, una vez dentro de ella el usuario califica a la fuente con la puntuación del 1 al 5 que desee, tras esto, se pasa la valoración al sistema, que se encarga de actualizar la valoración total de la fuente.

Una captura de pantalla de una red social

Descripción generada automáticamente

# **Modificar una fuente**

El usuario accede al menú de configuración, en el accede a una opción que le permite visualizar sus fuentes, una vez dentro de ellas, selecciona la fuente que desee modificar.

Al seleccionarla, el usuario verá los campos que puede modificar y hará los cambios que considere, tras esto presionará en guardar cambios por lo que el sistema modificará estos datos y los guardará en la base de datos, informando posteriormente al usuario de que sus cambios han sido guardados.

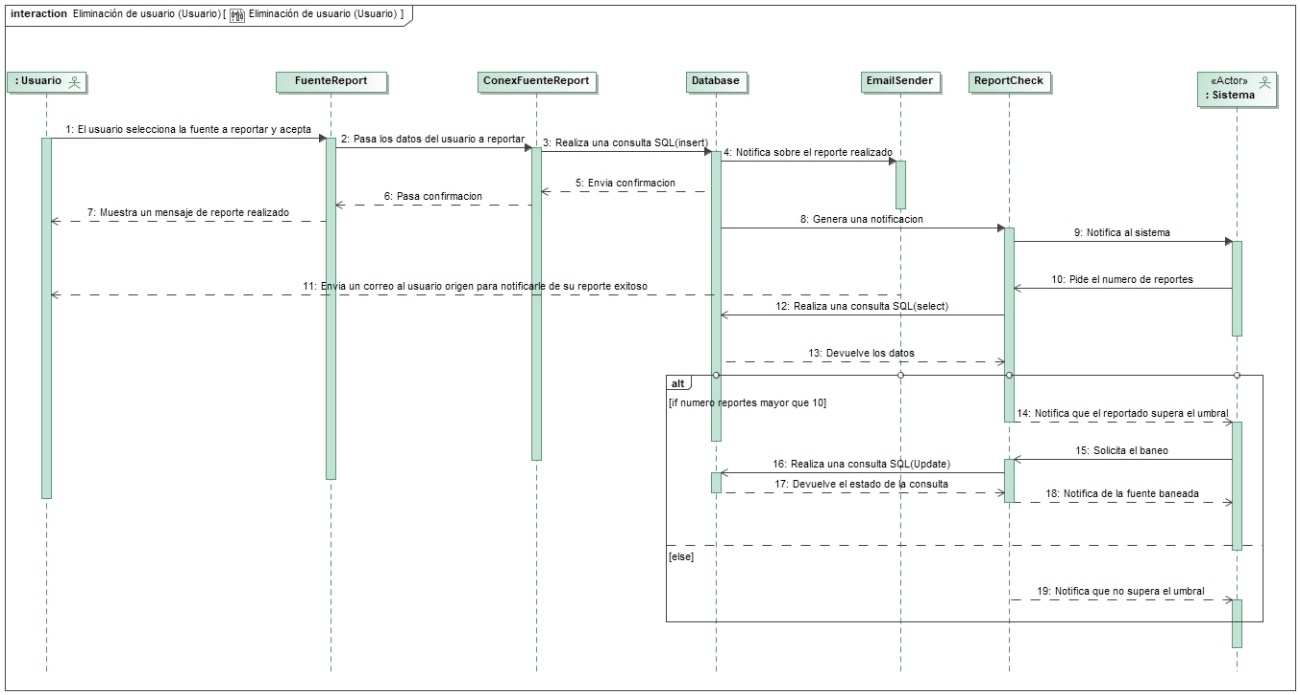
Una captura de pantalla de una red social

Descripción generada automáticamente

# **Reporte y baneo de fuente**

El usuario reporta a la fuente que desee e indica su motivo, el sistema efectúa el reporte sobre el usuario sumándole 1 a la casilla baneado de dicha fuente en la base de datos, tras esto, un .php envía un correo al usuario con su reporte.

Posteriormente se notifica al sistema del reporte realizado, y el sistema solicita ver el número de reportes de dicha fuente, si el número de reportes es mayor a 10, el sistema solicitará el baneo de dicha fuente, efectuándose automáticamente.



# **Eliminar Comentario**

# El usuario reporta al comentario que desee e indica su motivo, el sistema efectúa el reporte sobre el comentario sumándole 1 a la casilla baneado de dicho comentario en la base de datos, tras esto, un .php envía un correo al usuario con su reporte.

# Posteriormente se notifica al sistema del reporte realizado, y el sistema solicita ver el número de reportes de dicho comentario, si el número de reportes es mayor a 10, el sistema solicitará el baneo de dicho comentario, efectuándose automáticamente.

# 

# **Guardar datos en el móvil**

El usuario accede al menú de configuración, donde podrá efectuar los ajustes que prefiera, y al presionar el botón de guardado, dichos cambios se almacenarán en la caché del móvil

\*Este diagrama es un diagrama a futuro, para cuando se implementen opciones que cambien el aspecto de la aplicación, como un modo oscuro o efectos de sonido y música dentro de la aplicación por ejemplo, actualmente no tiene ningún uso ya que no disponemos de esos ajustes. Lo hemos decidido dejar para tener constancia de ello sin en algún momento decidimos perfeccionar la aplicación para su salida a mercado.\*

# 