|  |
| --- |
|  |
|  |

1. **Caso de uso: NEGOCIAR GROLLIES**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J1

**Objetivo en Contexto:**

El realizador selecciona negociar grollies para proponer un cambio en la recompensa

# Actor principal: Realizador de favor

**Actores** **secundarios**: Pedidor de favor, Sistema, Base de datos

**Qué datos usa:** Los datos respectivos al favor seleccionado

**Precondiciones:**

El realizador de favor selecciona un favor en la lista de favores y después selecciona negociar grollies

**Postcondiciones:**

**Éxito:** Se modifica correctamente la recompensa propuesta por el realizador de favor

**Fallo:** Mensaje de error: no se ha podido modificar la recompensa

# Flujo principal:

# 

1. El realizador de favor selecciona un favor entre la lista de favores
2. El realizador de favor selecciona negociar grollies
3. El realizador de favor propone nueva recompensa
4. El pedidor de favor acepta la nueva recompensa
5. El sistema accede a la base de datos y cambia la recompensa que había.
6. El sistema notifica el correcto cambio de la recompensa

# Flujos secundarios:

4.1 Si el pedidor de favor no acepta la nueva recompensa el sistema lo notificará y no se modificará la base de datos.

2.2 Si no es posible la comunicación entre el sistema y la base de datos se notificará y se volverá a la pantalla de búsqueda de favores

1. **Caso de uso: CAMBIAR E-MAIL**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J2

**Objetivo en Contexto:**

El objetivo es que el usuario desea cambiar su e-mail.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos y sistema.

**Qué datos usa:** e-mail antiguo y e-mail nuevo del usuario.

**Precondiciones:**

El usuario selecciona “cambiar configuración” y posteriormente cambia su e-mail.

**Postcondiciones:**

**Éxito:** Se modifica correctamente su e-mail.

**Fallo:** No se modifica su e-mail.

# Flujo principal:

# 

1. El usuario selecciona “cambiar configuración”.
2. El usuario reescribe su contraseña.
3. El usuario escribe su nuevo e-mal dos veces.
4. El usuario acepta el cambio del nuevo e-mail.
5. El sistema accede a la base de datos y cambia el e-mail.
6. El sistema notifica el correcto cambio de e-mail.

# Flujos secundarios:

2.1 Si el usuario no escribe bien su contraseña, aparecerá un mensaje de error con otro posible intento.

3.2 Si los dos e-mails no coinciden entre sí no se podrá realizar el cambio.

1. **Caso de uso: CAMBIAR IDIOMA**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J3

**Objetivo en Contexto:**

El objetivo es que el usuario desea cambiar el idioma.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos y sistema.

**Qué datos usa:** Ninguno

**Precondiciones:**

El usuario selecciona “cambiar configuración” y posteriormente “cambiar idioma”.

**Postcondiciones:**

**Éxito:** Se modifica correctamente el cambio de idioma.

**Fallo:** No se modifica el cambio de idioma.

# Flujo principal:

# 

1. El usuario selecciona “cambiar configuración”.
2. El usuario selecciona “cambiar idioma”.
3. El usuario selecciona el idioma deseado entre los disponibles.
4. El usuario acepta el cambio de idioma.
5. El sistema accede a la base de datos y cambia el idioma.
6. El sistema notifica el correcto cambio de idioma.

# Flujos secundarios:

1. **Caso de uso: CAMBIAR CONTRASEÑA**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J4

**Objetivo en Contexto:**

El objetivo es que el usuario desea cambiar la contraseña.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos y sistema.

**Qué datos usa:** Contraseña antigua y contraseña nueva.

**Precondiciones:**

El usuario selecciona “cambiar configuración” y posteriormente “cambiar contraseña”.

**Postcondiciones:**

**Éxito:** Se modifica correctamente el cambio de contraseña.

**Fallo:** No se modifica el cambio de contraseña.

# Flujo principal:

# 

1. El usuario selecciona “cambiar configuración”.
2. El usuario selecciona “cambiar contraseña”.
3. El usuario escribe su anterior contraseña.
4. El usuario escribe su nueva contraseña dos veces.
5. El sistema accede a la base de datos y cambia la contraseña.
6. El sistema notifica el correcto cambio de contraseña.

# Flujos secundarios:

2.1 Si el usuario no escribe bien su anterior contraseña, aparecerá un mensaje de error con otro posible intento.

3.2 Si las dos contraseñas no coinciden entre sí, no se podrá realizar el cambio.

1. **Caso de uso: CAMBIAR FOTO DE PERFIL**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J5

**Objetivo en Contexto:**

El objetivo es que el usuario desea cambiar la foto de perfil.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos y sistema.

**Qué datos usa:** Imagen externa a la aplicación e imagen actual.

**Precondiciones:**

El usuario selecciona “cambiar configuración” y posteriormente “cambiar imagen”.

**Postcondiciones:**

**Éxito:** Se modifica correctamente el cambio de imagen.

**Fallo:** No se modifica el cambio de imagen.

# Flujo principal:

# 

1. El usuario selecciona “cambiar configuración”.
2. El usuario selecciona “cambiar imagen”.
3. El usuario accede a la galería de su teléfono y selecciona una imagen.
4. El sistema accede a la base de datos y cambia la imagen.
5. El sistema notifica el correcto cambio de imagen.

# Flujos secundarios:

1. **Caso de uso: LEER NORMAS DE USO**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J6

**Objetivo en Contexto:**

El objetivo es que el usuario desea leer las normas de uso.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos y sistema.

**Qué datos usa:** Ninguna.

**Precondiciones:**

El usuario selecciona “cambiar configuración” y posteriormente “leer normas de uso”.

**Postcondiciones:**

**Éxito:** Podrá leer las normas de uso.

**Fallo:** No podrá leer las normas de uso.

**Flujo principal:**

1. El usuario selecciona “cambiar configuración”.
2. El usuario selecciona “leer normas de uso”.
3. El sistema accede a la base de datos y accede a las normas de uso.

# Flujos secundarios:

1. **Caso de uso: CONECTAR CON FACEBOOK**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J7

**Objetivo en Contexto:**

El objetivo es que el usuario desea conectar la aplicación con Facebook.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos y sistema.

**Qué datos usa:** e-mail y contraseña de Facebook

**Precondiciones:**

El usuario selecciona “cambiar configuración” y posteriormente “conectar con Facebook”.

**Postcondiciones:**

**Éxito:** Se conecta con Facebook.

**Fallo:** No se conecta con Facebook.

# Flujo principal:

# 

1. El usuario selecciona “cambiar configuración”.
2. El usuario selecciona “conectar con Facebook”.
3. El usuario escribe su e-mail y contraseña de Facebook para poder acceder.
4. El sistema accede a la base de datos y accede con Facebook.
5. El sistema notifica la conexión con Facebook.

# Flujos secundarios:

3.1 Si el usuario no escribe bien e-mail o contraseña, aparecerá un mensaje de error con otro posible intento.

1. **Caso de uso: BANEAR USUARIO**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J8

**Objetivo en Contexto:**

El usuario comete una infracción grave y es baneado.

# Actor principal: Administrador

**Actores** **secundarios**: Base de datos, Sistema, Usuario.

**Qué datos usa:** La información que causa la infracción.

**Precondiciones:** El usuario ha cometido una infracción y el sistema lo detectó.

**Postcondiciones:**

**Éxito:** El usuario ha sido baneado con éxito.

**Fallo:** Se determina que la infracción no es tan grave.

# Flujo principal:

# 

1. El Sistema detecta una infracción de las normas de uso.
2. El Administrador revisa si la infracción es grave o no.
3. El Administrador determina que es grave y decide banear.
4. El Usuario es baneado.

# Flujos secundarios:

2.1 Si la infracción no es grave el Administrador pone un strike Usuario en el caso de que la infracción sea leve.

2.2 Si la infracción no es grave el Administrador retira la disputa al Usuario en el caso de que la infracción fuese errónea.

1. **Caso de uso: PONER STRIKE A USUARIO**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J9

**Objetivo en Contexto:**

El usuario comete una infracción leve se le pone un strike.

# Actor principal: Administrador

**Actores** **secundarios**: Base de datos, Sistema, Usuario.

**Qué datos usa:** La información que causa la infracción.

**Precondiciones:** El usuario ha cometido una infracción y el sistema lo detectó.

**Postcondiciones:**

**Éxito:** Al usuario le colocan un strike con éxito en el caso de haber 3 se banea al usuario.

**Fallo:** Se determina que la infracción no es tan grave.

# Flujo principal:

# 

1. El Sistema detecta una infracción de las normas de uso.
2. El Administrador revisa si la infracción es grave o no.
3. El Administrador determina que es leve y decide poner un strike.
4. El Usuario es puesto un strike.

# Flujos secundarios:

2.1 Si la infracción no es grave el Administrador retira la disputa al Usuario en el caso de que la infracción fuese errónea.

4.1 Si hay tres strikes el Administrador pone en estado de baneo al Usuario.

1. **Caso de uso: ELIMINAR FAVOR NO ADECUADO.**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J10

**Objetivo en Contexto:**

El usuario publica un favor no adecuado y el sistema procede a tratarlo.

# Actor principal: Administrador

**Actores** **secundarios**: Base de datos, Sistema, Usuario.

**Qué datos usa:** La información que causa la infracción.

**Precondiciones:** El usuario ha cometido una infracción y el sistema lo detectó.

**Postcondiciones:**

**Éxito:** Se confirma que la infracción es grave y se borra.

**Fallo:** Se determina que la infracción no es tan grave y se reinstaura el favor.

# Flujo principal:

# 

1. El Administrador confirma que el usuario cometió una infracción.
2. El Administrador coloca strike o baneo.
3. El Administrador borra el favor no adecuado.

# Flujos secundarios:

1. **Caso de uso: BLOQUEAR CHAT .**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J11

**Objetivo en Contexto:**

Bloquear el chat de un usuario.

# Actor principal: Administrador

**Actores** **secundarios**: Base de datos, Sistema, Usuario.

**Qué datos usa:** Los datos del chat a bloquear y/o los datos del que solicita bloquear.

**Precondiciones:** El usuario ha cometido una infracción y es denunciada por otro usuario o usuario decide bloquear directamente el chat sin denunciarlo.

**Postcondiciones:**

**Éxito:** Se confirma que la infracción y se bloquea o se bloquea directamente.

**Fallo:** El administrador no acepta la denuncia y la única manera de bloquear es por parte del usuario.

# Flujo principal:

# 

1. El Administrador confirma que el usuario cometió una infracción.
2. El Administrador elimina el chat y no permite al usuario hablar al solicitante.
3. El Administrador procede a ver el castigo necesario.

# Flujos secundarios:

* 1. El Usuario procede a bloquear al otro usuario sin denunciar.

1. **Caso de uso: NOTIFICAR ACEPTACIÓN DE FAVOR.**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J12

**Objetivo en Contexto:**

Notifica al usuario que su favor ha sido aceptado

# Actor principal: Sistema

**Actores** **secundarios**: Base de datos, Usuario.

**Qué datos usa:** Los datos propios de los usuarios implicados.

**Precondiciones:** El favor ha sido solicitado por un usuario y tras ser aceptado el sistema envía una notificación al mismo.

**Postcondiciones:**

**Éxito:** Se confirma que se acepta la realización del favor y se le envía notificación.

**Fallo:** El dueño del favor niega al usuario la realización de este.

# Flujo principal:

# 

1. El Pedidor del Favor Acepta que el otro usuario realice su favor.
2. El Sistema recibe la información y formaliza una notificación.
3. El Sistema envía la notificación al realizador del favor.

# Flujos secundarios:

1.1 El Pedidor del favor niega la posibilidad de hacer el favor terminando el flujo.

1. **Caso de uso: PEDIR FAVOR.**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J13

**Objetivo en Contexto:**

Publicar un favor en la plataforma.

# Actor principal: Pedidor de Favor.

**Actores** **secundarios**: Base de datos, Sistema.

**Qué datos usa:** Los datos proporcionados por el usuario.

**Precondiciones:** El pedidor del favor necesita estar registrado y tener grollies para dar en su cuenta.

**Postcondiciones:**

**Éxito:** El favor es publicado.

**Fallo:** El favor no ha podido ser publicado.

# Flujo principal:

# 

1. El Pedidor del Favor publica un favor poniendo toda la información.
2. El Sistema actualiza la feed con este nuevo favor.

# Flujos secundarios:

* 1. El Sistema envía mensaje de error al usuario por no poderse publicar el favor.

1. **Caso de uso: PONER TIEMPO DE ENTREGA.**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J14

**Objetivo en Contexto:**

Poner un tiempo máximo para entregar el favor.

# Actor principal: Pedidor de Favor.

**Actores** **secundarios**: Base de datos, Sistema, realizador del favor.

**Qué datos usa:** Los datos proporcionados por el usuario.

**Precondiciones:** El pedidor del favor necesita estar registrado y tener un favor publicado.

**Postcondiciones:**

**Éxito:** El tiempo de favor es aplicado.

**Fallo:** El tiempo del favor no se pudo actualizar o se rechaza.

# Flujo principal:

# 

1. El Pedidor del Favor pone un tiempo para hacer el favor.
2. El Realizador del Favor acepta el tiempo y procede a realizar el favor.

# Flujos secundarios:

* 1. El Realizador del Favor reniega del tiempo.

2.2.1 El Realizador del Favor negocia un nuevo tiempo.

2.2.2 El Realizador del Favor termina el favor y decide no hacerlo.

1. **Caso de uso: SOLICITAR PREMIOS.**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J15

**Objetivo en Contexto:**

Cambiar Grollies por premios.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos, Sistema, Admin.

**Qué datos usa:** Los datos del catálogo de premios, del usuario y de su entidad bancaria.

**Precondiciones:** El usuario necesita haber hecho favores para poder conseguir regalos.

**Postcondiciones:**

**Éxito:** El premio tiene el coste en Grollies necesario al número de estos del usuario.

**Fallo:** El usuario no tiene suficientes Grollies para comprar el premio.

# Flujo principal:

1. El Usuario Selecciona un premio del catálogo.

2. El Sistema Acepta la transacción y envía el premio reduciendo los Grollies correspondientes de la cuenta del usuario.

# Flujos secundarios:

* 1. El Sistema Niega la transacción por falta de fondos o stock.

1. **Caso de uso: CAMBIAR CONFIGURACIÓN.**

**Nombre del AUTOR:** Santiago Mourenza

**Identificador**: J16

**Objetivo en Contexto:**

Cambiar alguna configuración de la app.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos, Sistema, Admin.

**Qué datos usa:** Ninguno.

**Precondiciones:** El usuario necesita haber iniciado sesión con su usuario y contraseña.

**Postcondiciones:**

**Éxito:** Entra en la configuración de la app.

# Flujo principal:

1. El Usuario Selecciona cambiar configuración.

2. Se cambia al menú de configuración y el usuario elije que cambio quiere hacer.

# Flujos secundarios:

* 1. El usuario decide no cambiar nada y vuelve atrás.

1. **Caso de uso: SOLICITAR GROLLIES.**

**Nombre del AUTOR:** Jaime Martínez

**Identificador**: J17

**Objetivo en Contexto:**

El usuario quiere comprar Grollies en la aplicación

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos, Sistema, Admin.

**Qué datos usa:** Los datos del catálogo de Grollies, del usuario y de su entidad bancaria.

**Precondiciones:** El usuario quiere comprar Grollies en la aplicación.

**Postcondiciones:**

**Éxito:** El usuario compra Grollies

# Flujo principal:

1. El Usuario selecciona comprar Grollies

2. El Usuario selecciona un método de pago

3. Rellena los datos de pago según el método elegido

4. El usuario compra Grollies

# Flujos secundarios:

* 1. El usuario decide no comprar Grollies y volver atrás

1. **Caso de uso: PROPONER GROLLIES.**

**Nombre del AUTOR:** Jaime Martínez

**Identificador**: J18

**Objetivo en Contexto:**

Proponer una determinada cantidad de Grollies por obtener un favor.

# Actor principal: Usuario.

**Actores** **secundarios**: Base de datos, Sistema, Admin.

**Qué datos usa:** Ninguno

**Precondiciones:** El usuario demandante quiere ofrecer una cantidad de Grollies por obtener un favor.

**Postcondiciones:**

**Éxito:** El usuario demandante acepta o rechaza la oferta

# Flujo principal:

1. El Usuario ofrece un favor.

2. El Usuario propone una oferta en Grollies por obtener el favor

# Flujos secundarios:

* 1. El usuario decide no realizar la oferta y volver atrás

1. **Caso de uso: GESTIONAR ALMACÉN DE DATOS**

**Nombre del AUTOR:** Pablo Torre

**Identificador**: J19

**Objetivo en Contexto:**

Tratar con los cambios en los datos del sistema

# Actor principal: Sistema

**Actores** **secundarios**: Base de datos, Administrador, Usuario

**Qué datos usa:** Todos los cambios efectuados por el usuario/administrador que deben ser registrados además de toda la información almacenada previamente

**Precondiciones:**

El usuario solicita ver la Localización del Producto en el Plano.

**Postcondiciones:**

**Éxito:** Se guarda el cambio en la base de datos.

**Fallo:** Mensaje de error: no se ha podido efectuar el cambio

# Flujo principal:

# 

1. El sistema realiza uno de los siguientes cambios en la aplicación: añadir/eliminar usuario, … HACER
2. El sistema accede a la base de datos y la modifica.
3. El sistema muestra una notificación con el cambio efectuado
4. El sistema vuelve a la pantalla previa a este caso de uso

# Flujos secundarios:

2.1 Si no se puede modificar la base de datos se notificará y se saltará al punto 4 del flujo principal. (Por ejemplo, si no se pone toda la información necesaria para añadir un usuario)

* 1. Si no es posible la comunicación entre el sistema y la base de datos se notificará y se saltará al punto 4 del flujo principal

1. **Caso de uso: BUSCAR UN FAVOR CON FILTRO**

**Nombre del AUTOR:** Pedro Palacios

**Identificador**: J20

**Objetivo en Contexto:**

Buscar entre la lista de favores publicados

# Actor principal: Usuario

**Actores** **secundarios**: Base de datos, Usuario

**Qué datos usa:** La información almacenada de todos los favores que están disponibles en ese instante.

**Precondiciones:**

El usuario se encuentra en la pantalla de inicio de la aplicación

**Postcondiciones:**

**Éxito:** Se muestran todos los favores posibles

**Fallo:** Mensaje de error. Se posibilita realizar una nueva búsqueda

# Flujo principal:

# 

1. El usuario selecciona la opción buscar un favor.
2. El sistema muestra la pantalla de búsqueda con filtro.
3. El usuario introduce una recompensa mínima en grollies (opcional).
4. El usuario decide buscar por localización y selecciona una distancia máxima (opcional).
5. El usuario introduce una fecha límite máxima (opcional).
6. El usuario lanza la búsqueda.
7. El sistema selecciona los favores de base de datos que verifican los filtros introducidos.
8. La base de datos devuelve el resultado de esta selección.
9. El sistema muestra por pantalla los favores seleccionados.

# Flujos secundarios:

6.1 Si no es posible la comunicación entre el sistema y la base de datos se le avisa al usuario de que el servicio no está disponible en ese momento.

8.1 En el caso en que la búsqueda no obtenga resultados se notifica al usuario y se le da la opción de modificar filtros de búsqueda

1. **Caso de uso: SELECCIONAR FAVOR**

**Nombre del AUTOR:** Alberto Almagro

**Identificador**: J21

**Objetivo en Contexto:**

El usuario selecciona un favor de los que aparecen en pantalla para acceder a más información sobre esta.

# Actor principal: Usuario

**Actores** **secundarios**: Base de datos, Sistema

**Qué datos usa:** Información mostrada por pantalla por *buscar discoteca*. La información almacenada de todas las discotecas en la base de datos

**Precondiciones:**

El usuario ha utilizado *buscar favor* con éxito

**Postcondiciones:**

**Éxito:** Se accede al perfil con la información del favor seleccionado

**Fallo:** Mensaje de error al cargar el favor seleccionado

# Flujo principal:

# 

1. El usuario selecciona un favor mostrado en pantalla.
2. El sistema pide a la base de datos la información del favor.
3. El base de datos devuelve la información del favor solicitado
4. El sistema muestra en pantalla la información de la discoteca seleccionada

# Flujos secundarios:

2.1 Si no es posible la comunicación entre el sistema y la base de datos se avisa al usuario de que el servicio no está disponible en ese momento.

1. **Caso de uso: CHATEAR CON USUARIO**

**Nombre del AUTOR:** Juan Carlos Llamas

**Identificador**: J22

**Objetivo en Contexto:**

El usuario selecciona el chat para hablar sobre el favor en cuestión

# Actor principal: Usuario

**Actores** **secundarios**: Usuario, Sistema, Base de datos

**Qué datos usa:** Los datos respectivos al favor seleccionado

**Precondiciones:**

El usuario selecciona un favor en la lista de favores y después selecciona chatear.

**Postcondiciones:**

**Éxito:** Se abre el chat correctamente, permitiendo a ambos usuarios hablar.

**Fallo:** Mensaje de error: no se ha podido abrir chat

# Flujo principal:

# 

1. El usuario selecciona un favor entre la lista de favores
2. El usuario selecciona el chat
3. El sistema accede a la base de datos y almacena un nuevo chat.
4. El sistema lleva a usuario es a una nueva pantalla en la que chatear.

# Flujos secundarios:

2.1 Si no se puede almacenar un nuevo chat en la base de datos se notificará al usuario, diciéndole que pruebe en otro momento.

2.2 Si no es posible la comunicación entre el sistema y la base de datos se notificará y se volverá a la pantalla de búsqueda de favores