
Trabajo de Fin de Grado

Héctor Olivares Rivero

Proyecto omRocket

Diagramas de Secuencia

Tabla de contenido

1. Índice de Figuras	2
2. Diagramas de Secuencia	3
2.1 Diagrama de Secuencia: Evaluar Campaña (Fig.1)	3
2.2 Diagrama de Secuencia: Consultar Historial (Fig.2)	9
2.3 Diagrama de Secuencia: Gestionar Usuarios (Fig.3)	11

1. Índice de Figuras

Fig. 1: Diagrama de Secuencia: Evaluar Campaña

Fig. 2: Diagrama de Secuencia: Consultar Historial

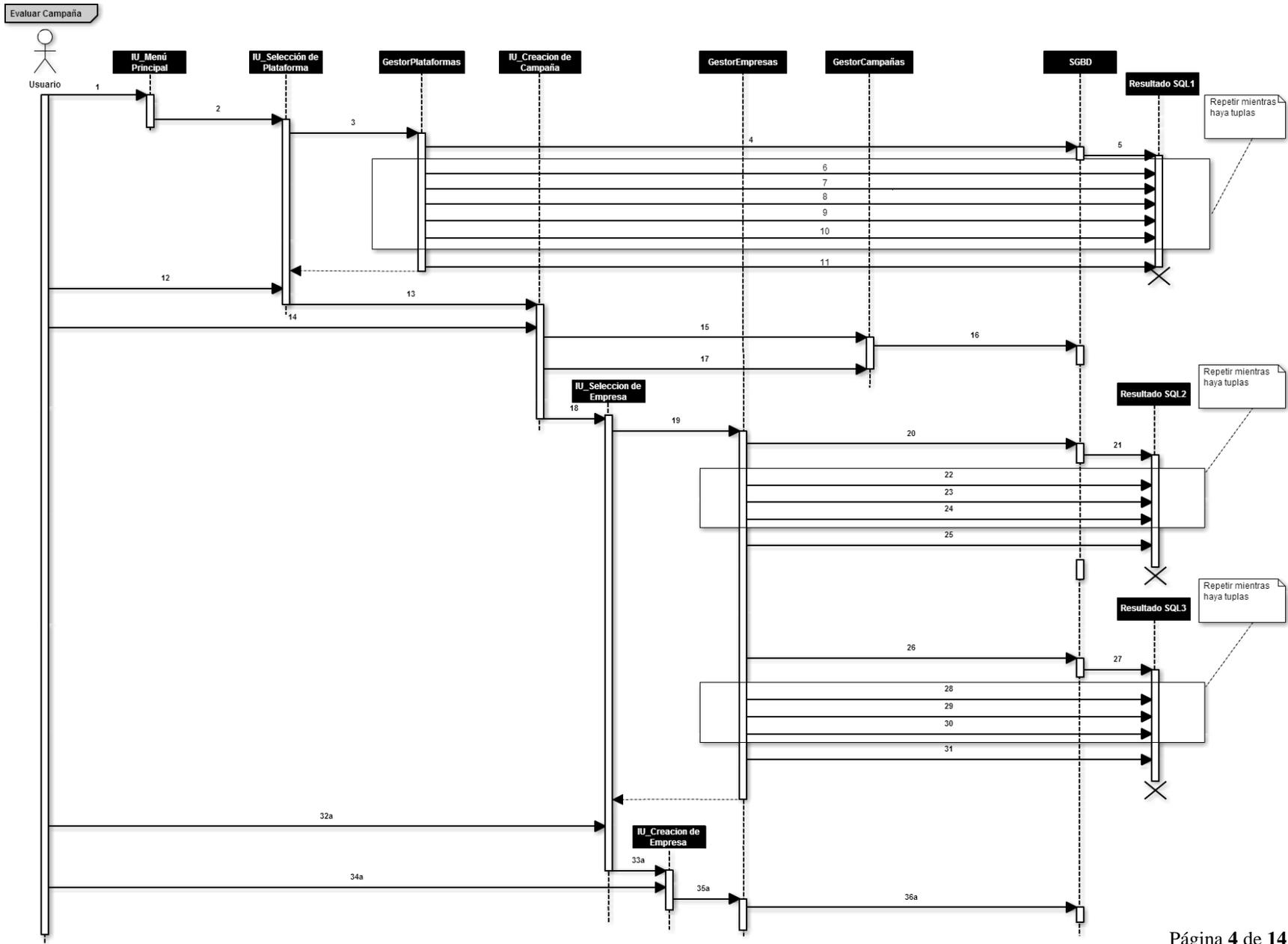
Fig.3: Diagrama de Secuencia: Gestionar Usuarios

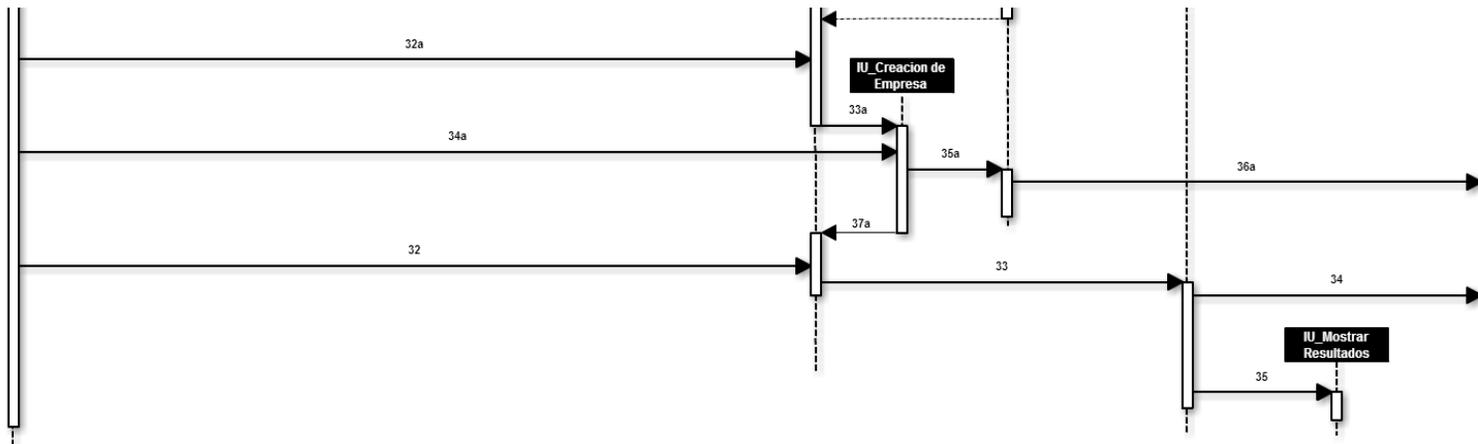
2. Diagramas de Secuencia

Este apartado de diseño contiene los diagramas que detallan los flujos de eventos y las interacciones entre módulos de las principales funcionalidades del sistema.

2.1 Diagrama de Secuencia: Evaluar Campaña (Fig.1)

Este diagrama muestra el flujo de eventos y las interacciones entre módulos para el caso de uso evaluar campaña:





(imagen continúa el diagrama a partir del 37a).

Nota: Los gestores serán las clases controladoras (nombreController) y contendrán descripciones de todos los métodos que contengan. Aquí solo se comentarán algunos de ellos para facilitar la comprensión de los diagramas. Del mismo modo que en el diagrama de clases, en la implementación las variables (incluidas las pasadas por parámetro) irán precedidas del símbolo "\$" aunque aquí no se refleje.

- 1- El Usuario selecciona "Go!" en la tabla de navegación del menú principal.
- 2- New()
- 3- actionIndex(): Coleccion <Plataforma> -- *Método encargado de cargar todas las plataformas existentes en la BBDD.*
- 4- execSQL("SELECT *
FROM tbl_plataforma): ResultadoSQL
- 5- New()

6- Next()

7- getString("nombrePlat"): String

8- getString("descripcionPlat"): String

9- getInt("idPlat"): String -- *Se obtiene también la PK aunque no se cargue en pantalla, será enviada como parámetro.*

10- getImageUrl(): String

11- Close()

12- El usuario selecciona la Plataforma para la que desea realizar la campaña.

13- New(idPlat)

14- El usuario introduce los indicadores de la campaña para que sea evaluada y pulsa "Continuar".

15- nuevaCamp=actionCreateCampForUserAndPlatform(idPlat): void -- *una nueva campaña es creada con este método y se almacena su modelo en la variable nuevaCamp de modo que se pueda recoger su PK directamente para utilizarla más adelante.*

16- `execSQL("INSERT INTO tbl_campaña (nombre_campaña, estado, fecha_creacion, fecha_modificacion, fecha_registro, beneficios, gastos, coste_hora, , horas_campaña, n_leads) VALUES (%nomCamp%, %est%, %fCreac%, %fMod%, %fReg%, %benef%, %gast%, %costeH%, %hCamp%, %nLeads%)");`

17- getInt("idCamp"): String

18- New(idCamp)

19- actionIndex(idCamp): Coleccion <Empresa> -- *método encargado de ejecutar 2 SQLs: por un lado cargar todas las campañas con las que el usuario logueado ha trabajado anteriormente (ha tenido contratos previos) y por otro, todas las empresas de la BD.*

20- `execSQL("SELECT tbl_empresa.nombre_empresa AS nom_empr, presupuesto AS presup
FROM tbl_Empresa
INNER JOIN tbl_emp_user_fecha`

ON tbl_emp_user_fecha.id_user=Yii::app->user->id"): ResultadoSQL

Yii::app->user->id -- comando de la api de Yii que devuelve el id del usuario con sesión activa. Evita arrastrar el id a través de las interfaces mejorando la seguridad y simplificando métodos.

21- New()

22- Next()

23- getString("nom_empr"): String

24- getInt("presupuesto"): int

25- Close()

26- execSQL("SELECT tbl_empresa.nombre_empresa AS nom_empr2, presupuesto AS presup2

FROM tbl_empresa"): ResultadoSQL

27- New()

28- Next()

29- getString("nom_empr2"): String

30- getInt("presup2"): int

31- Close()

32a- Si el usuario desea añadir una nueva empresa al sistema, seleccionará "Crear nueva Empresa" en la interfaz de Selección de empresa.

33a-New()

34a-El usuario introduce los datos de la nueva empresa y pulsa "Crear"

35a- actionCreate(): Empresa

36a- execSQL("INSERT INTO tbl_empresa (nombre_campaña, presupuesto) VALUES (%nom_camp%, %presup%)");

37a- New() --redirección a la interfaz de selección de empresa.

32- Tanto si el usuario si el usuario ha creado una nueva empresa, como si no, elegirá una empresa de las cargadas en la interfaz y se procederá a evaluar la campaña.

33- actionEvaluateCamp("idEmp"): void --Método encargado de hacer los cálculos relativos a los resultados introducidos sobre la campaña creada unos pasos atrás para evaluarla. Posteriormente actualizará el valor del "resul" en la tabla campaña y en todas las que sea necesario.

Redireccionará a la vista de resultados pasándole como parámetros, los obtenidos en el método: el valor del roi, el gasto total de la campaña y una serie de mensajes e imágenes dependientes del resultado de esta.

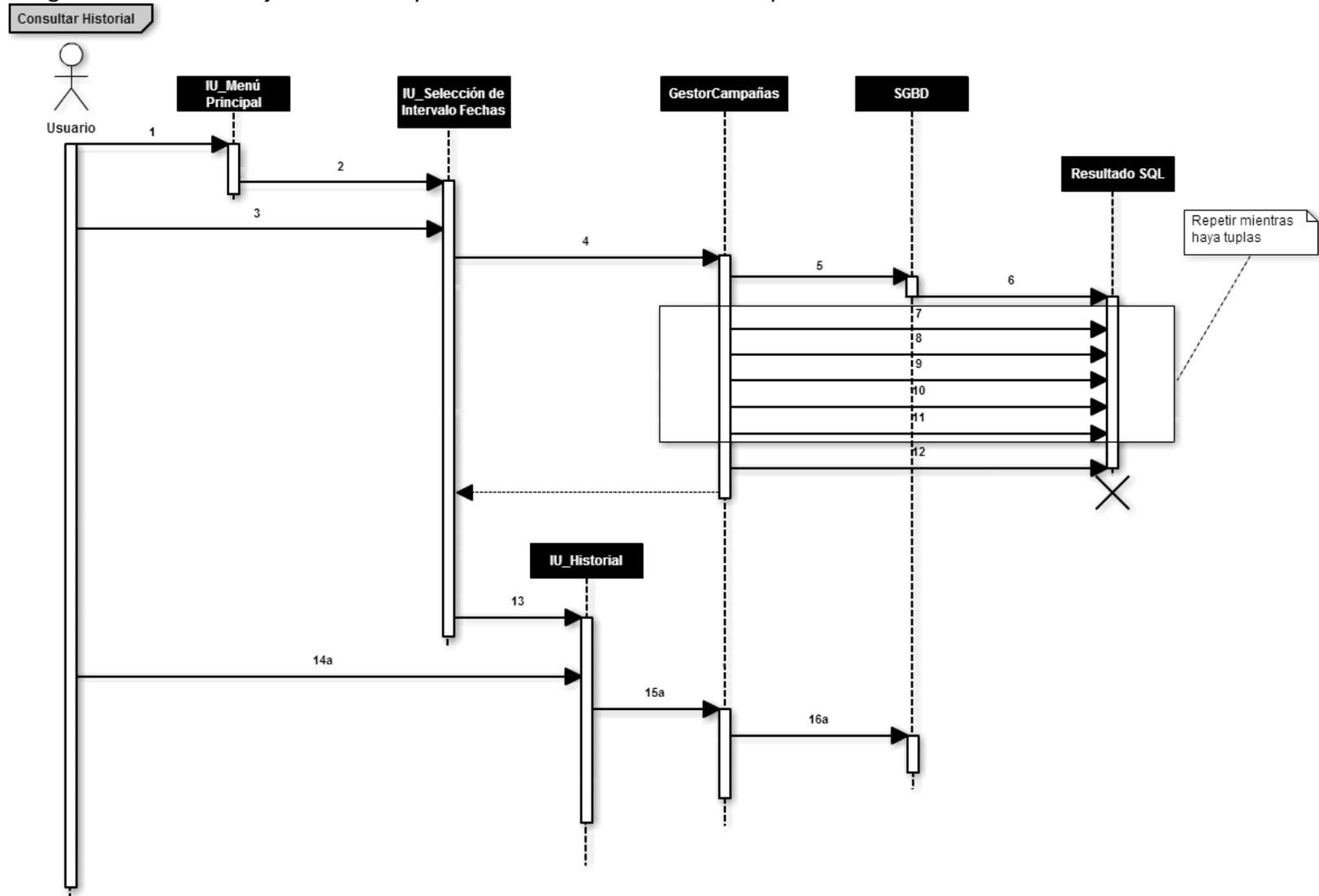
34- execSQL("UPDATE tbl_campaña -- actualizará el valor resul de campaña con el valor obtenido en el método de evaluación.

```
SET resul=%roi%");
```

35- New(roi, totalgastado, msng, msng1, imgsng)

2.2 Diagrama de Secuencia: Consultar Historial (Fig.2)

Este diagrama muestra el flujo de eventos y las interacciones entre módulos para el caso de uso consultar historial:

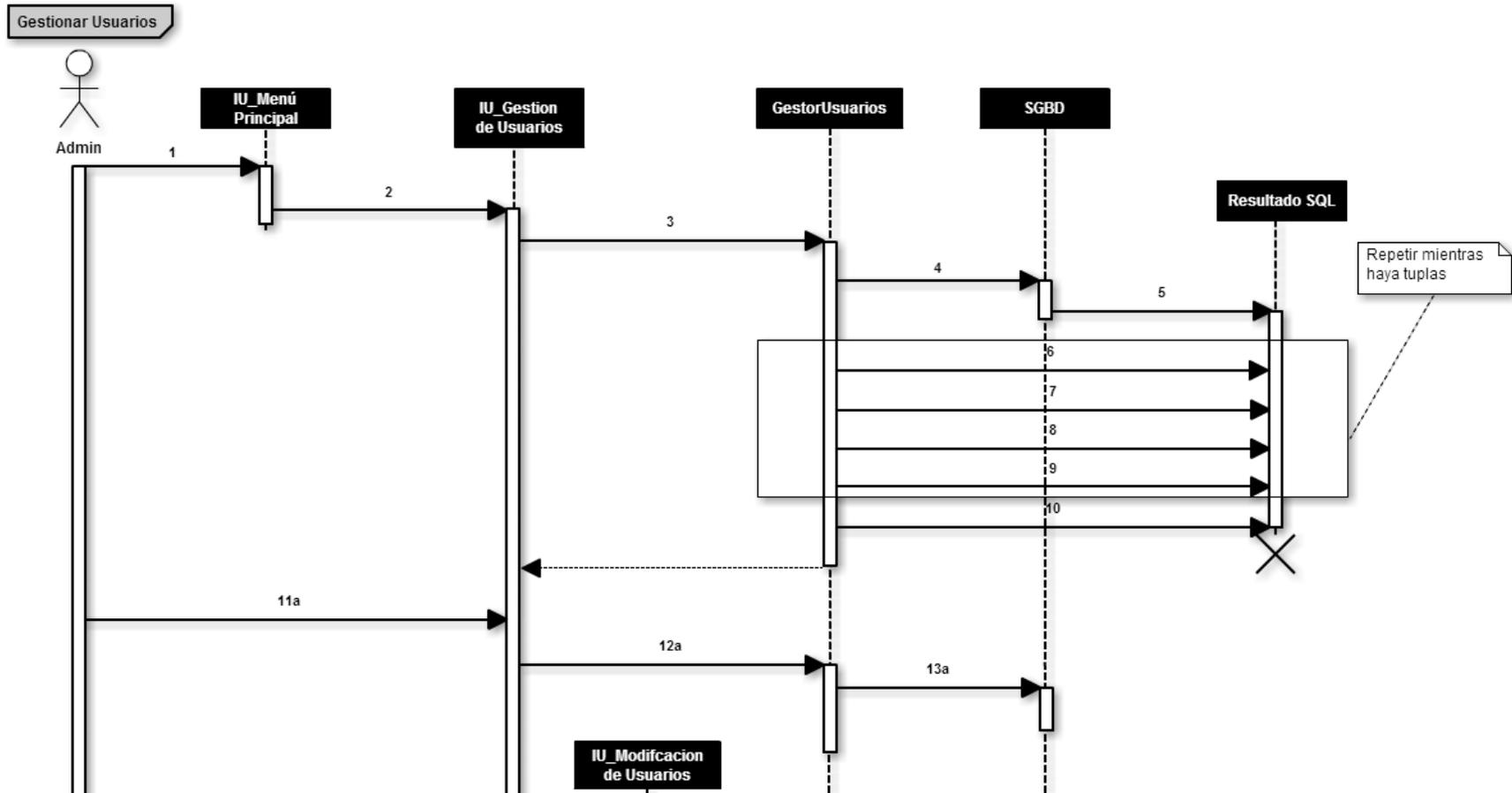


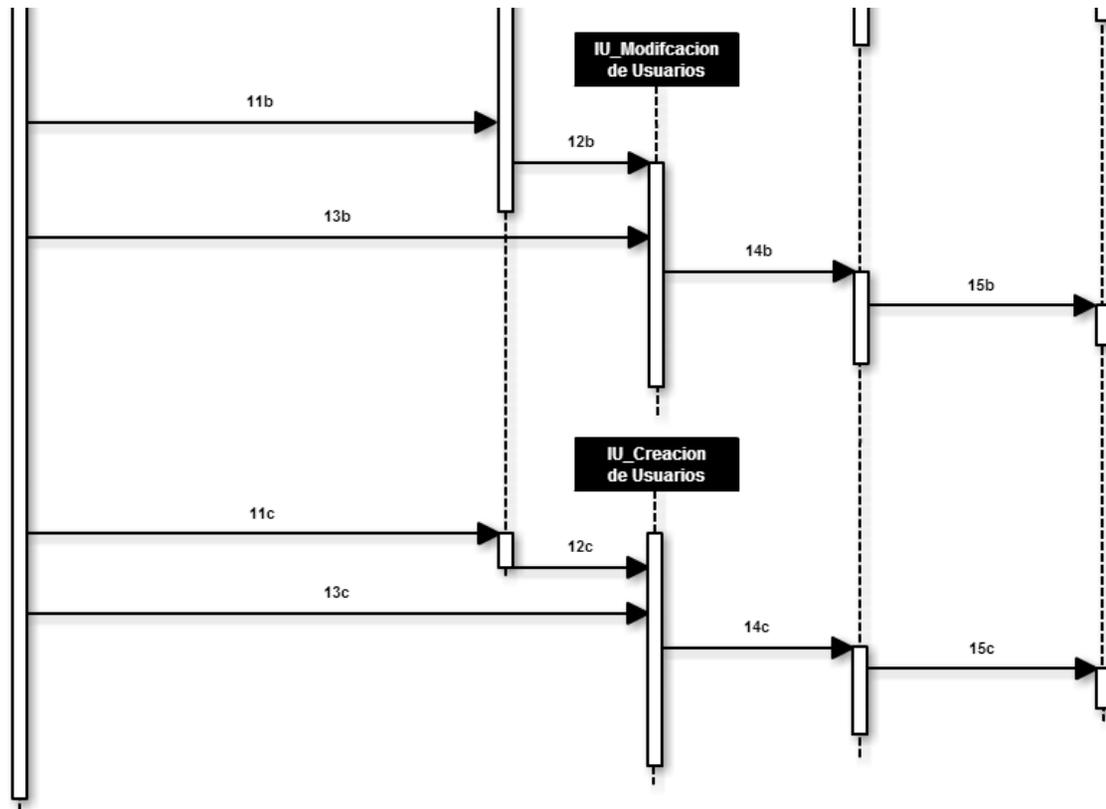
- 1- El Usuario selecciona "Historial" en la tabla de navegación del menú principal.
- 2- New()
- 3- El usuario introduce la fecha a partir de la cual desea comenzar la búsqueda y la fecha límite y pulsa "Filtrar".
- 4- Filter(*fech_ini*, *fech_fin*): Coleccion <Campana> -- *Método encargado de realizar el filtrado para dicho usuario entre las fechas introducidas.*
- 5- execSQL("SELECT *
FROM tbl_campana
WHERE fecha_creacion>%fech_ini%
AND fecha_finalizacion<%fech_fin%
AND id_user= Yii::app->user->id"): ResultadoSQL
- 6- New()
- 7- Next()
- 8- getString("nombre_campana"): String
- 9- getDate("fecha_creacion"): Date
- 10- getDate("fecha_finalizacion"): Date
- 11- getFloat("resul"): float
- 12- Close()
- 13- New(Coleccion<Campana>)-- *se le mostrará el historial de las campañas obtenidas con el método anterior y que estarán comprendidas entre las fechas seleccionadas.*
- 14a- Si el usuario desea eliminar la campaña, seleccionará el icono en forma de cubo de basura sobre la campaña que lo desee.
- 15a- actionDelete(*idCampana*)

16a- `execSQL("DELETE FROM tbl_campaña WHERE id = %IdCampaña%");` -- La tabla `tbl_campaña` tiene habilitada la opción `ON DELETE CASCADE`, de modo que eliminará las claves extranjeras de las tablas correspondientes.

2.3 Diagrama de Secuencia: Gestionar Usuarios (Fig.3)

Este diagrama muestra el flujo de eventos y las interacciones entre módulos para el caso de uso gestionar usuarios:





- 1- El Admin selecciona "Admin: Gestionar Usuarios" en la tabla de navegación de su menú principal.
- 2- New()
- 3- actionAdmin(): Coleccion <User> -- *Método encargado de realizar la carga con todos los usuarios del sistema*
- 4- execSQL("SELECT *
FROM user"): ResultadoSQL
- 5- New()

6- Next()

7- getString("nombre_usuario"): String

8- getString("email"): String

9- getInt("id"): int

10- Close()

11a- Si el Admin desea eliminar usuario, seleccionará el icono en forma de cubo de basura del usuario deseado.

12a- `actionDelete(idUser)` -- *Elimina el usuario seleccionado y todas sus campañas y contratos de la BBDD. Redirecciona de nuevo a la página del administrador, lo que permite continuar con la gestión de usuarios sin más preámbulos.*

13a- `execSQL("DELETE FROM user
WHERE id = %pIdUser%");`

11b- Si el Admin desea modificar algún usuario, seleccionará el icono en forma de lápiz sobre el usuario deseado.

12b- `New(idUser)`

13b- El Administrador introducirá las credenciales del usuario y pulsa "Modificar".

14b- `actionUpdate(idUser)` -- *En estos casos es necesario arrastrar la id del usuario como parámetro y no es correcto obtenerla de la sesión, ya que lógicamente se estaría modificando el usuario logueado y no el seleccionado por el administrador. Redirecciona de nuevo a la página del administrador, lo que permite continuar con la gestión de usuarios directamente.*

15b- `execSQL("UPDATE user -- por seguridad, el método Update envía las variables necesarias a través de Post y no de Get.
SET nombre_usuario=%nom_usu%,
email=%mail%,
password_hash=%pwd%);` -- *encriptará automáticamente la password introducida por el admin a su password_hash.*

11c- Si el administrador desea añadir un nuevo usuario al sistema seleccionará "Crear Usuario"

12c- New()

13c- El administrador introduce las credenciales del nuevo usuario y pulsa "Crear".

14c- actionCreate(): User -- *Método encargado de crear un nuevo usuario. Recogerá los parámetros del formulario y utilizará el método Post para insertarlos en el nuevo modelo.*

15c- `execSQL("INSERT INTO user (nombre_usuario, email) VALUES (%nom_user%, %mail%)");`