

INFORME FINAL

Fecha: 13/01/2019

Participantes:

Edurne Lopez

Ana Miranda

Julen Perez

Leire Ruiz

ÍNDICE

Introducción	5
Casos de Uso	6
Diagrama de Casos de Uso	6
Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuario	7
Registro	7
Identificación	11
Identificación RRSS	14
Recuperación de contraseña	17
Cambio de contraseña	18
Visualización del ranking	22
Ranking mejores partidas usuario	22
Ranking mejor puntuación diaria usuario	25
Ranking mejores partidas	28
Ranking mejores jugadores	31
Guardar/Reanudar Partida	33
Guardar partida	33
Reanudar partida	38
Iniciar Partida	41
Ayuda	45
Personalización	48
Modelo de Dominio	¡Error! Marcador no definido.
Plan de Pruebas	55
Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuarios	55
Registro	55
Identificación	56
Recuperación de contraseña	56
Cambio de contraseña	57
Visualización del ranking	58
Guardar/Reanudar Partida	60
Guardar Partida	60
Reanudar Partida	61
Iniciar Partida	62
Ayuda	63

Personalización	64
Diagrama de clases	66
Diagrama de clases y base de datos a partir del modelo de dominio	66
Diagrama de clases final	68
Diagramas de comunicación	70
Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuarios	70
Registro	70
Diagrama de comunicación	70
Identificación	71
Recuperación de contraseña	72
Cambio de contraseña	73
Visualización del ranking	74
Ranking mejores partidas usuario	74
Ranking mejor puntuación del día	75
Ranking mejores partidas	76
Ranking mejores jugadores	77
Guardar/Reanudar partida	78
Guardar partida	79
Reanudar partida	80
Personalización	83
Diagramas de secuencia	84
Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuario	84
Registro	84
Identificación	85
Recuperación de contraseña	87
Cambio de contraseña	87
Visualización del ranking	89
Ranking mejores partidas usuario	89
Ranking mejor puntuación diaria usuario	89
Ranking mejores partidas	90
Ranking mejores jugadores	90
Guardar/Reanudar partida	91
Guardar partida	91
Reanudar partida	95
Personalización	99
Problemas en la implementación	101

1. Introducción

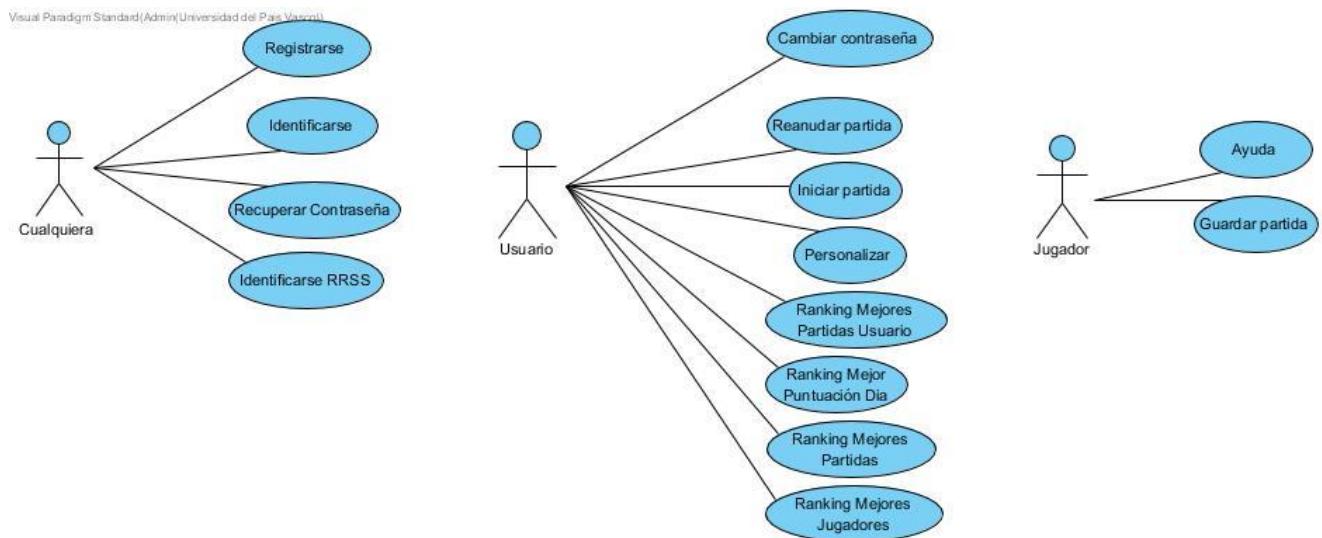
A continuación aparece un listado con las funcionalidades que hay que incorporar al sistema.

- 1. Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuarios.** Este proceso de identificación se realizará a través de cuentas de RRSS (Facebook, Google, ...). Cada usuario tendrá que indicar qué nombre de usuario quiere utilizar en Bar Bestial.
- 2. Visualización de rankings.** Al finalizar la partida se mostrarán varios rankings, (1) las mejores partidas de ese usuario con sus puntuaciones, (2) la mejor puntuación de ese día junto con el jugador que la haya obtenido, (3) las mejores partidas jugadas desde el inicio junto con los jugadores correspondientes, las puntuaciones que obtuvieron y la fecha de la partida y (4) los jugadores con mejor puntuación media, considerando las partidas ganadas desde el inicio.
- 3. Guardar/Reanudar partida.** En cualquier momento, un usuario puede decidir guardar una partida a medias. Al identificarse, podrá escoger entre iniciar una nueva partida o continuar con alguna de las que tenga guardadas.
- 4. Ayuda.** Cada vez que el usuario gana una partida y consigue más de 6 puntos (6 animales en el bar), gana una ayuda. La ayuda consiste en poder sacar una carta del contrario del bar y ponerla en la cola. Como máximo 2 veces por partida. Se penaliza la puntuación con un 10% cada vez que se utiliza la ayuda.
- 5. Personalización.** Posibilidad de crear “configuraciones” de forma que un jugador pueda utilizar imágenes personalizadas para sus cartas y elegir con cuál de sus configuraciones quiere jugar.

2. Casos de Uso

En este apartado, se describen los principales casos de uso planteados por los miembros del grupo, junto con las interfaces que se han añadido al proyecto “Bar Bestial”.

2.1 Diagrama de Casos de Uso



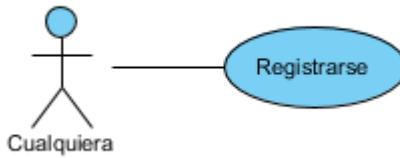
Jerarquía de actores



- **Cualquiera**. Persona que accede a la pantalla inicial del juego sin estar identificado.
- **Usuario**. Persona que se ha identificado en el juego mediante redes sociales o de manera local.
- **Jugador**. Persona que está jugando una partida en el Bar Bestial. Una vez acabada la partida volverá a ser un usuario.

2.2 Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuario

2.2.1 Registro

	
Nombre:	Registrarse
Descripción: Permite a los usuarios registrarse en el juego “Bar Bestial”.	
Actores: Cualquiera.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario pulsa el botón “Registrarte” (Ilustración 1).2. Se abre la ventana de registro en la cual el usuario introduce sus datos (Ilustración 2). [SI PULSA “CANCELAR”]<ol style="list-style-type: none">3.1. Se vuelve a la ventana del menú principal (Ilustración 1).[SI PULSA “ACEPTAR”]<ol style="list-style-type: none">[SI NO SE HAN RELLENADO TODOS LOS CAMPOS]<ol style="list-style-type: none">3.2.1. Se mostrará un mensaje (Ilustración 4) y al pulsar “Aceptar” se volverá a la pantalla de registro (Ilustración 2).[SI SE HAN RELLENADO TODOS LOS CAMPOS]<ol style="list-style-type: none">[SI EXISTE UN USUARIO CON EL MISMO EMAIL O NOMBRE]<ol style="list-style-type: none">3.2.2.1. Se mostrará un mensaje de error (Ilustración 5, Ilustración 6) y al pulsar “Aceptar” se volverá a la pantalla de registro (Ilustración 2).[SI NO EXISTE NINGÚN USUARIO CON EL MISMO EMAIL Y NOMBRE]<ol style="list-style-type: none">[SI LAS CONTRASEÑAS NO COINCIDEN]<ol style="list-style-type: none">3.2.2.2.1. Se mostrará un mensaje de error (Ilustración 7) y al pulsar “Aceptar” se volverá a la pantalla de registro (Ilustración 2).[SI LAS CONTRASEÑAS COINCIDEN]<ol style="list-style-type: none">3.2.2.2.2. Se registra al usuario y se abre la ventana principal del juego (Ilustración 3).	

Postcondiciones: El usuario quedará registrado en el juego .

Interfaz Gráfica:

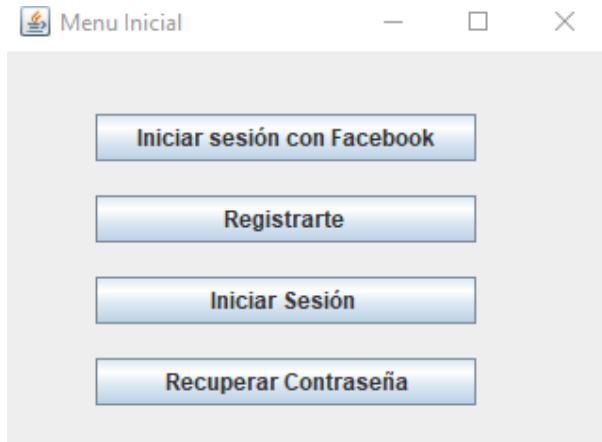


Ilustración 1

Illustración 2 muestra la interfaz gráfica de un formulario de registro. En la parte superior, hay un icono de una lucecita encendida y la etiqueta "Registro". El formulario consiste en cuatro campos de texto para datos personales:

Email	<input type="text"/>
Nombre	<input type="text"/>
Contraseña	<input type="text"/>
Confirmar contraseña	<input type="text"/>

Abajo del formulario, hay dos botones:

- Aceptar
- Cancelar

Ilustración 2

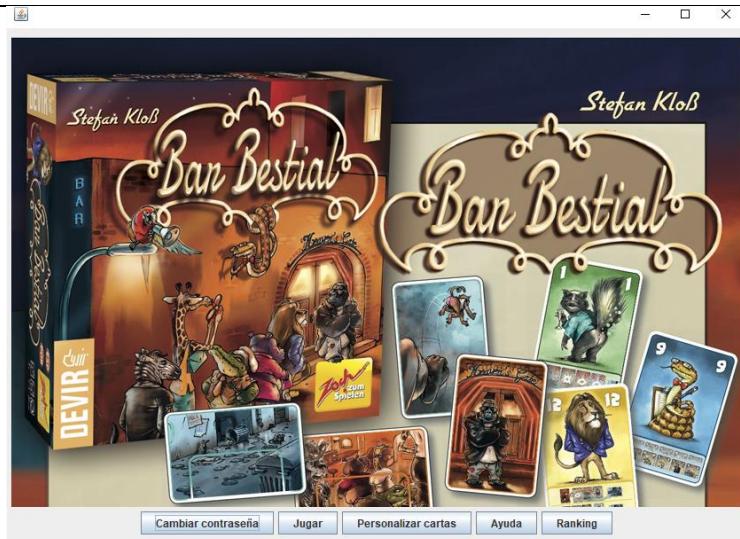


Ilustración 3

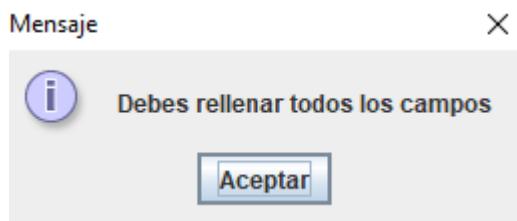


Ilustración 4

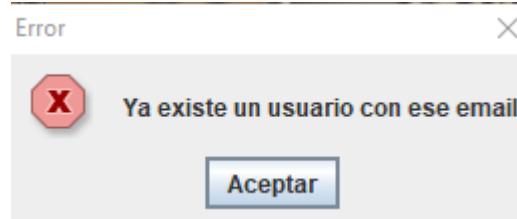


Ilustración 5

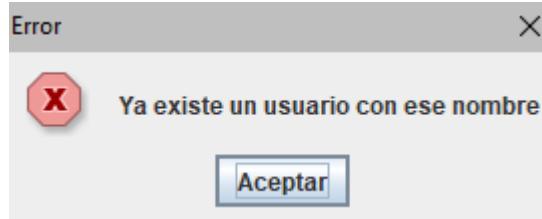


Ilustración 6

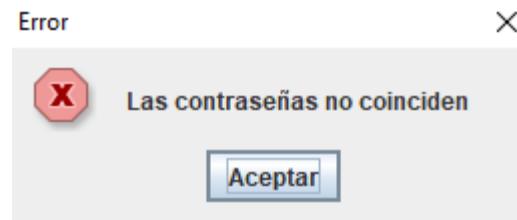


Ilustración 7

2.2.2 Identificación

<pre> graph LR C((Cualquiera)) --- I([Identificarse]) </pre>	
Nombre:	Identificarse.
Descripción: Permite a cualquier usuario identificarse en el juego Bar Bestial.	
Actores: Cualquiera.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none"> 1. El usuario pulsa el botón “Iniciar sesión” (Ilustración 1). 2. Se abre la ventana de inicio de sesión en el que introduce el email y la contraseña (Ilustración 2). <p>[SI PULSA “CANCELAR”]</p> <ol style="list-style-type: none"> 3.1. Se vuelve a la ventana del menú inicial (Ilustración 1). <p>[SI PULSA “ACEPTAR”]</p> <p>[SI NO SE HAN RELLENADO TODOS LOS CAMPOS]</p> <ol style="list-style-type: none"> 3.2.1. Se mostrará un error (Ilustración 5) y al pulsar “Aceptar” se volverá a la pantalla de Login (Ilustración 2). <p>[SI SE HAN RELLENADO TODOS LOS CAMPOS]</p> <p>[SI LOS DATOS NO SON CORRECTOS]</p> <ol style="list-style-type: none"> 3.2.2.1. Se muestra una pantalla de error (Ilustración 4) y al pulsar “Aceptar” se vuelve a la pantalla de inicio de sesión (Ilustración 2). <p>[SI LOS DATOS SON CORRECTOS]</p> <ol style="list-style-type: none"> 3.2.2.2. Se abre la ventana principal del juego. (Ilustración 3). 	
Postcondiciones: Una vez identificado correctamente, el usuario accede a la ventana principal del juego.	
Interfaz Gráfica:	

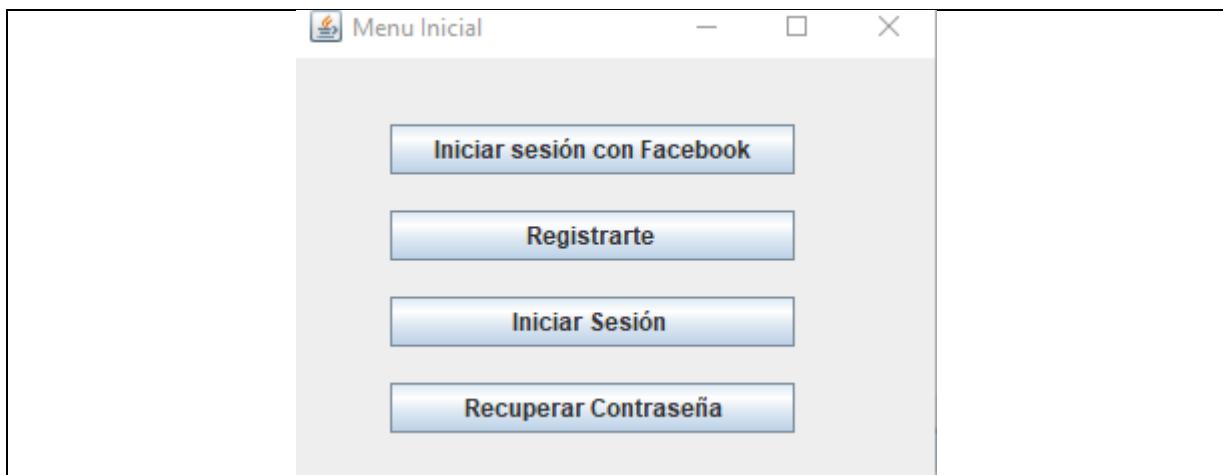


Ilustración 1

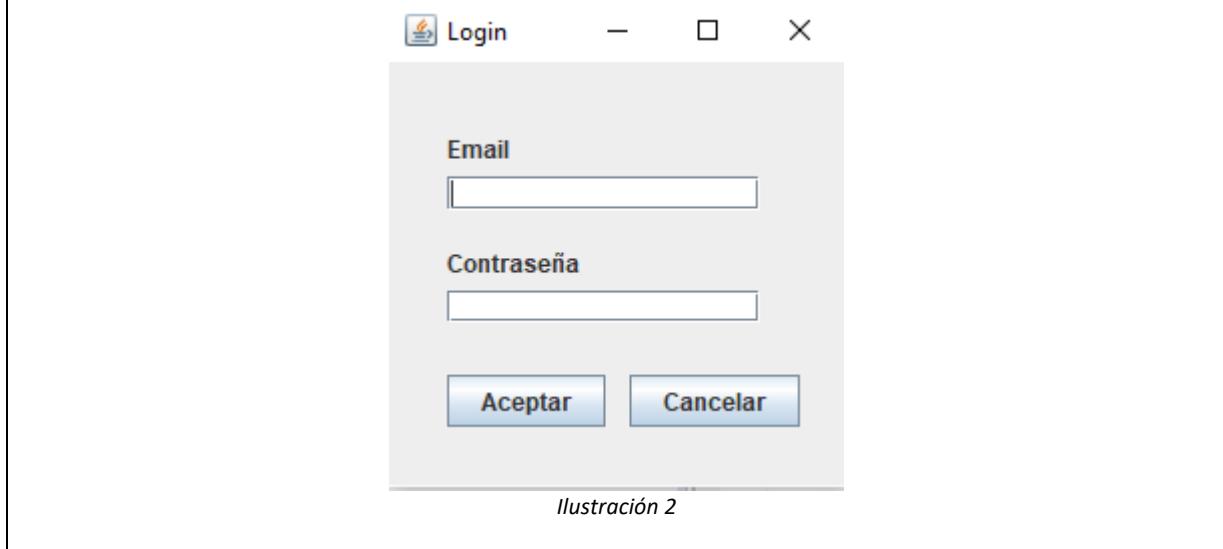


Ilustración 2



Ilustración 3

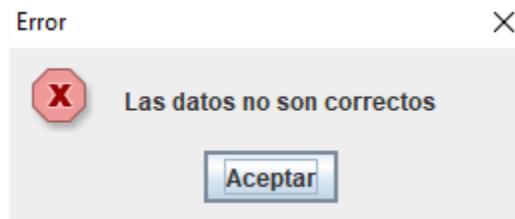


Ilustración 4

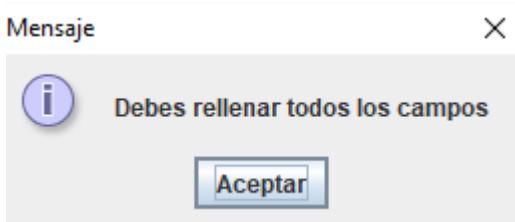
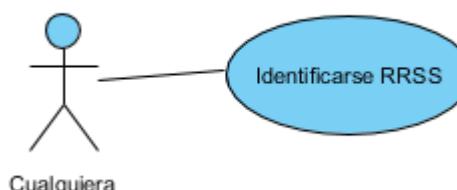


Ilustración 5

2.2.3 Identificación RRSS

	
Nombre:	Identificarse RRSS.
Descripción: Permite a cualquier persona identificarse en el juego mediante la red social Instagram.	
Actores: Cualquiera.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario pulsa el botón “Iniciar sesión con Instagram” (Ilustración 1).2. Se abre la ventana con la página de inicio de sesión de Facebook (Ilustración 2).3. Si los datos son correctos se abre la ventana principal del juego (Ilustración 3).	
Postcondiciones: Una vez identificado correctamente, el usuario será redirigido a la pantalla principal del juego.	
Interfaz Gráfica:	

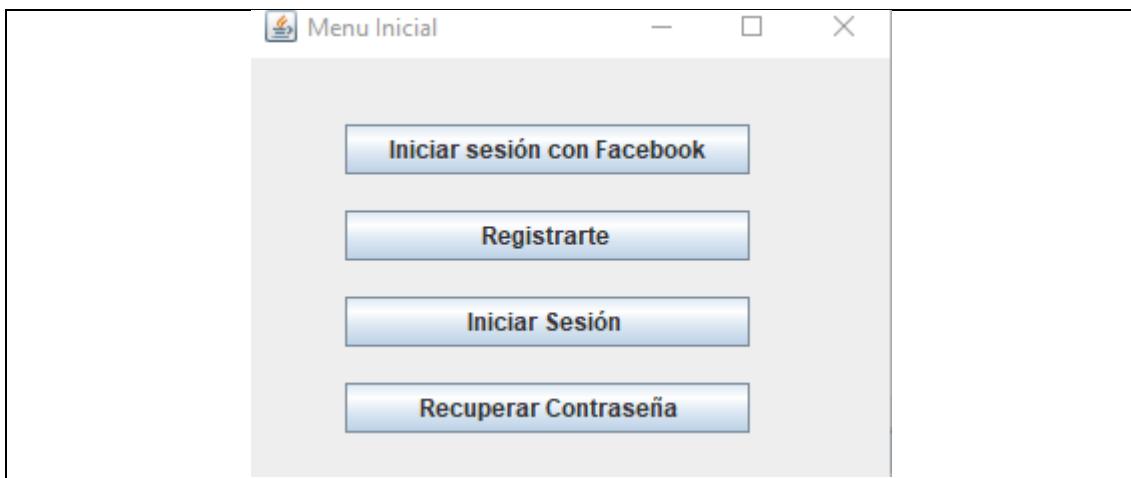


Ilustración 1

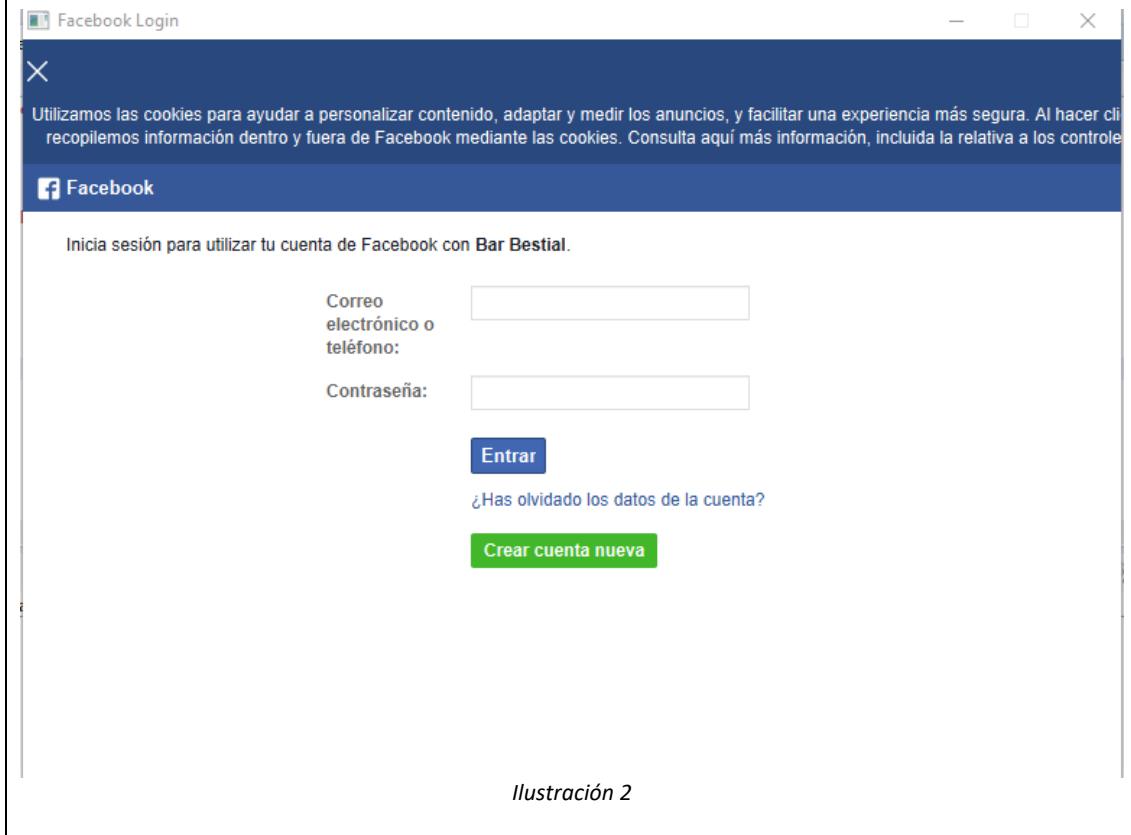
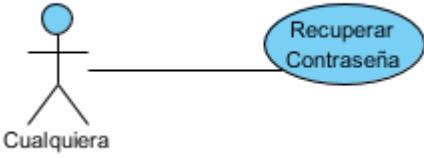


Ilustración 2



Ilustración 3

2.2.4 Recuperación de contraseña

	
Nombre:	Recuperar contraseña
Descripción:	Permite al usuario recuperar su contraseña.
Actores:	Cualquiera.
Precondiciones:	Ninguna.
Requisitos no funcionales:	Ninguno.
Flujo de Eventos:	<ol style="list-style-type: none">1. El usuario pulsa el botón “Recuperar Contraseña” (Ilustración 1).2. Se abrirá una ventana para introducir el email (Ilustración 2). [SI PULSA “CANCELAR”]<ol style="list-style-type: none">3.1. Se vuelve a la pantalla del menú inicial (Ilustración 1). [SI PULSA “ACEPTAR”]<ol style="list-style-type: none">[SI NO SE HAN RELLENADO TODOS LOS CAMPOS]<ol style="list-style-type: none">3.2.1. Se notifica al usuario (Ilustración 4) y al pulsar “Aceptar” se vuelve a la pantalla de cambio de contraseña (Ilustración 2). [SI SE HAN RELLENADO TODOS LOS CAMPOS]<ol style="list-style-type: none">[SI EL EMAIL ES CORRECTO]<ol style="list-style-type: none">3.2.2.1. Se enviará un mensaje al email con la contraseña nueva y se volverá al menú inicial (Ilustración 1). [SI EL EMAIL NO ES CORRECTO]<ol style="list-style-type: none">3.2.2.2. Se notificará al usuario (Ilustración 3) y al pulsar “Aceptar” vuelve a la pantalla de recuperar contraseña (Ilustración 2).
Postcondiciones:	Se ha enviado su contraseña al correo y se guarda la nueva contraseña.
Interfaz Gráfica:	

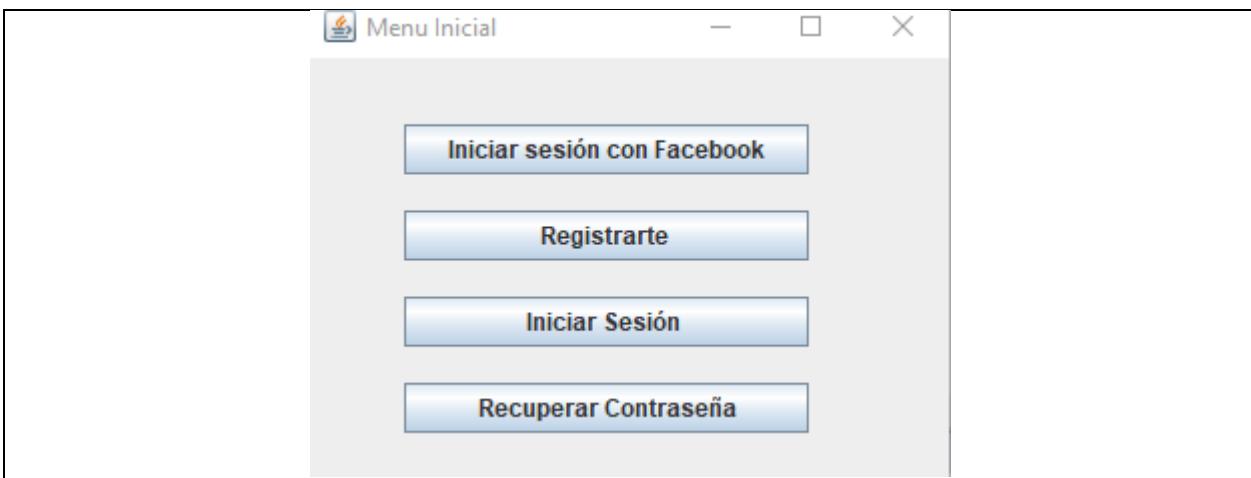


Ilustración 1

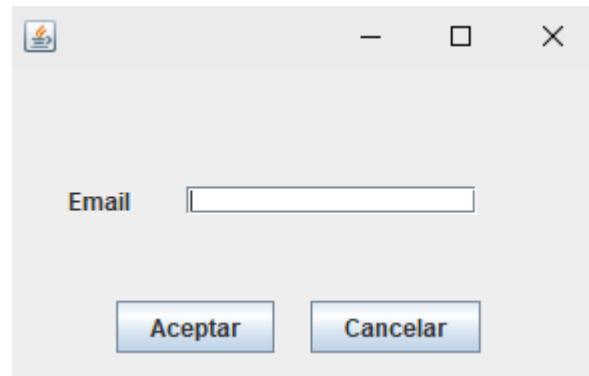


Ilustración 2

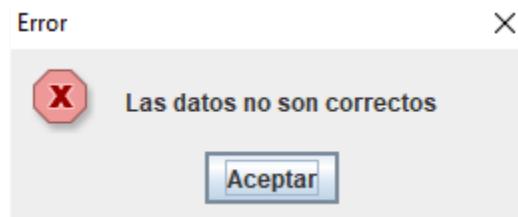


Ilustración 3

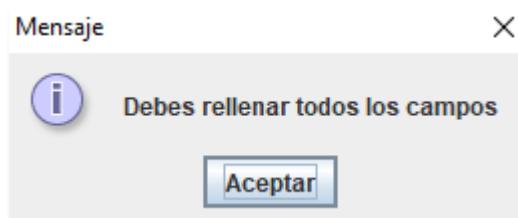
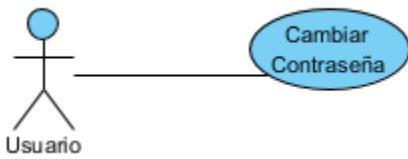


Ilustración 4

2.2.5 Cambio de contraseña


<p>Nombre: Cambiar contraseña</p>
<p>Descripción: Permite al usuario modificar su contraseña.</p>
<p>Actores: Usuario.</p>
<p>Precondiciones: Ninguna.</p>
<p>Requisitos no funcionales: Ninguno.</p>
<p>Flujo de Eventos:</p> <ol style="list-style-type: none"> 1. El usuario identificado selecciona en su menú principal la opción “Cambiar contraseña” (Ilustración 1). 2. Se abrirá una ventana con un formulario. Hay que poner la propia contraseña y la nueva (Ilustración 2). <p>[SI PULSA “CANCELAR”]</p> <ol style="list-style-type: none"> 3.1. Se vuelve a la pantalla principal del juego (Ilustración 1). <p>[SI PULSA “ACEPTAR”]</p> <p>[SI NO SE HAN RELLENADO TODOS LOS CAMPOS]</p> <ol style="list-style-type: none"> 3.2.1. Se notifica al usuario (Ilustración 5) y al pulsar “Aceptar” se vuelve a la pantalla de cambio de contraseña (Ilustración 2). <p>[SI SE HAN RELLENADO TODOS LOS CAMPOS]</p> <p>[SI LA CONTRASEÑA ANTIGUA ES INCORRECTA]</p> <ol style="list-style-type: none"> 3.2.2.1. Se notifica al usuario (Ilustración 4). <p>[SI LA CONTRASEÑA ANTIGUA ES CORRECTA]</p> <p>[SI LAS DOS CONTRASEÑAS NUEVAS COINCIDEN]</p> <ol style="list-style-type: none"> 3.2.2.2.1. Se modifica la contraseña con la nueva contraseña y se vuelve a la pantalla principal del juego (Ilustración 1). <p>[SI LAS DOS CONTRASEÑAS NO COINCIDEN]</p> <ol style="list-style-type: none"> 3.2.2.2.2. Se notifica al usuario (Ilustración 3) y al pulsar “Aceptar” se vuelve a la pantalla de cambio de contraseña (Ilustración 2).
<p>Postcondiciones: Se ha modificado la contraseña del usuario.</p>

Interfaz Gráfica:

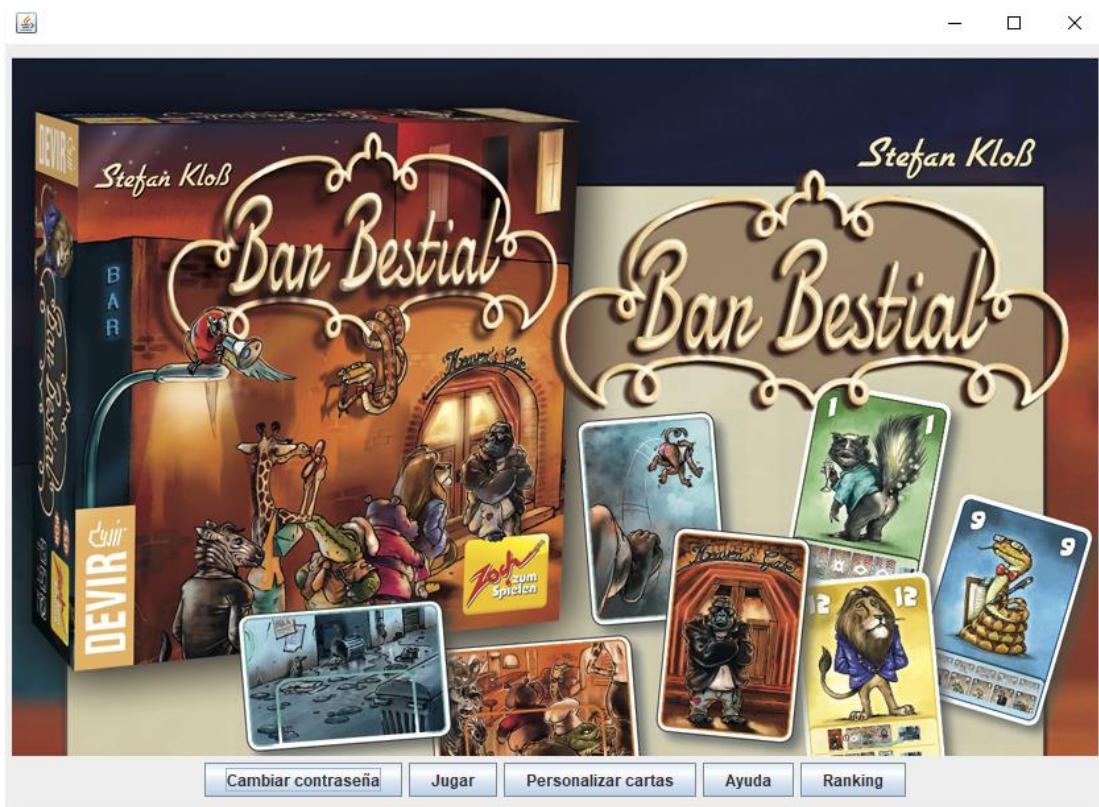


Ilustración 1

Contraseña antigua

Nueva Contraseña

Confirmar Contraseña

Aceptar Cancelar

Ilustración 2

Error

X



Las contraseñas no coinciden

Aceptar

Ilustración 3

Error

X



La contraseña antigua no es correcta

Aceptar

Ilustración 4

Mensaje

X



Debes llenar todos los campos

Aceptar

Ilustración 5

2.2 Visualización del ranking

2.2.1 Ranking mejores partidas usuario

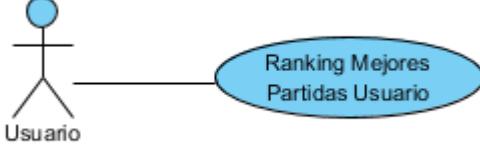
	
Nombre:	Ranking Mejores Partidas Usuario
Descripción: El usuario podrá visualizar el ranking de sus mejores partidas jugadas hasta el momento junto con su puntuación.	
Actores: Usuario.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario selecciona en el menú principal la opción “Ranking” (Ilustración 1)2. Dentro del menú del “Ranking”, el usuario selecciona la opción de “Tus Mejores Partidas” (Ilustración 2). <p>[SI NO TIENE NINGUNA PARTIDA FINALIZADA]</p> <ol style="list-style-type: none">3.1. Se muestra un mensaje de información (Ilustración 3) y al pulsar “Aceptar” se vuelve a la pantalla del ranking (Ilustración 2) <p>[SI TIENE PARTIDAS FINALIZADAS]</p> <ol style="list-style-type: none">3.2. Se muestra la nueva ventana con los datos correspondientes al ranking de “Tus Mejores Partidas” (Ilustración 4).	
Postcondiciones: Ninguna.	
Interfaz Gráfica:	



Ilustración 1



Ilustración 2

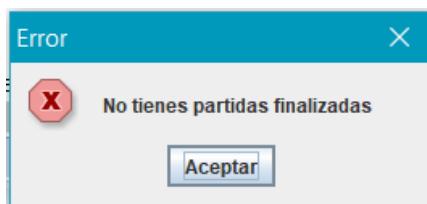
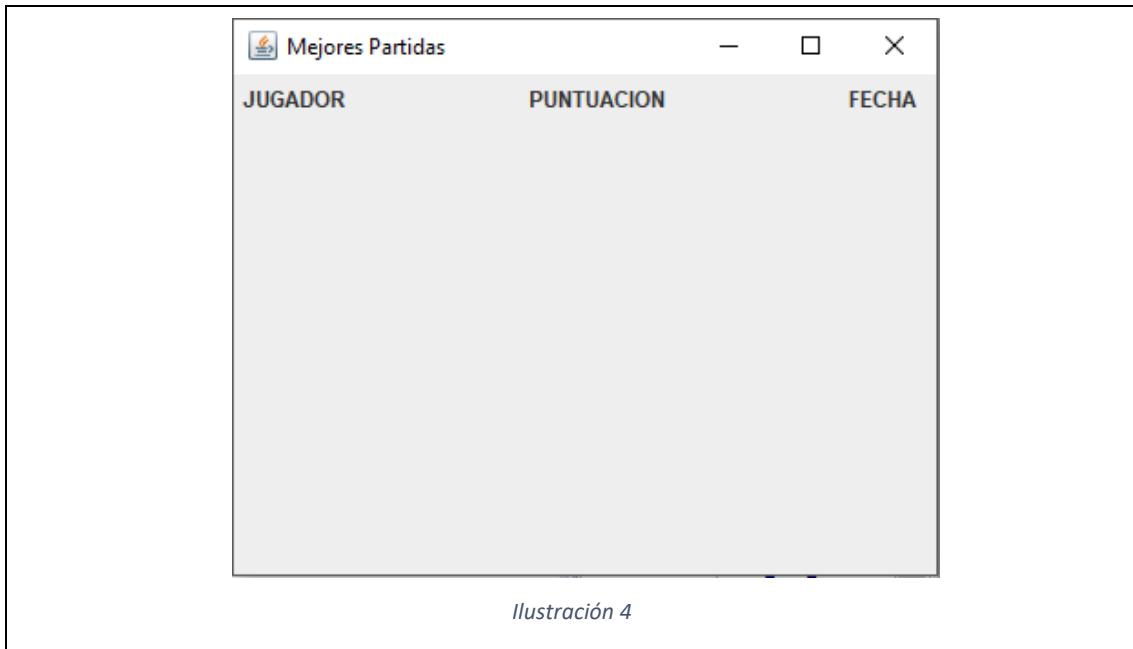


Ilustración 3



2.3.2 Ranking mejor puntuación diaria usuario

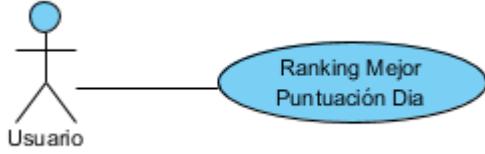
	
Nombre:	Ranking Mejor Puntuación Dia
Descripción: El usuario podrá visualizar el ranking con su mejor puntuación diaria.	
Actores: Usuario.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario selecciona en el menú principal la opción “Ranking” (Ilustración 1)2. Dentro del menú del “Ranking”, el usuario selecciona la opción de “Puntuación Dia” (Ilustración 2). <p>[SI TODAVÍA NO HAY PARTIDAS REGISTRADAS ESE DIA]</p> <ol style="list-style-type: none">3.1. Se muestra un mensaje (Ilustración 3) y al pulsar “Aceptar” se vuelve a la pantalla del ranking (Ilustración 2). <p>[SI EXISTEN PARTIDAS REGISTRADAS ESE DIA]</p> <ol style="list-style-type: none">3.2. Se muestra la nueva ventana con los datos correspondientes al ranking de “Puntuación Día” (Ilustración 4).	
Postcondiciones: Ninguna.	
Interfaz Gráfica:	



Ilustración 1



Ilustración 2

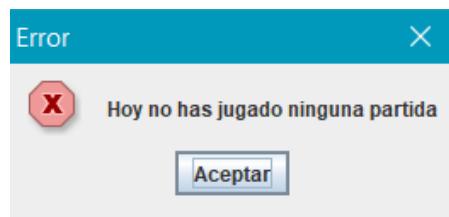


Ilustración 3



Ilustración 4

2.3.3 Ranking mejores partidas

 Usuario  Ranking Mejores Partidas	
Nombre:	Ranking Mejores Partidas
Descripción: El usuario podrá visualizar el ranking histórico con las mejores partidas jugadas hasta el momento.	
Actores: Usuario.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario selecciona en el menú principal la opción “Ranking” (Ilustración 1)2. Dentro del menú del “Ranking”, el usuario selecciona la opción de “Mejores Partidas” (Ilustración 2). <p>[SI NO HAY PARTIDAS REGISTRADAS]</p> <ol style="list-style-type: none">3.1. Se muestra un mensaje (Ilustración 3) y al pulsar “Aceptar” se vuelve a la pantalla del ranking (Ilustración 2). <p>[SI HAY PARTIDAS REGISTRADAS]</p> <ol style="list-style-type: none">3.2. Se muestra la nueva ventana con los datos correspondientes al ranking de “Mejores Partidas” (Ilustración 4).	
Postcondiciones: Ninguna.	
Interfaz Gráfica:	



Ilustración 1

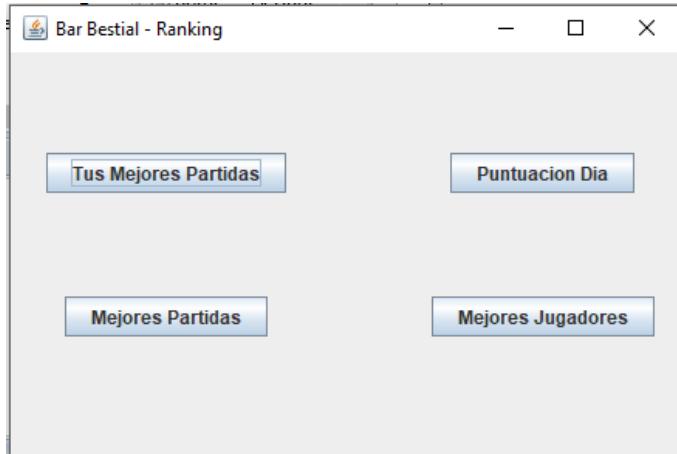


Ilustración 2

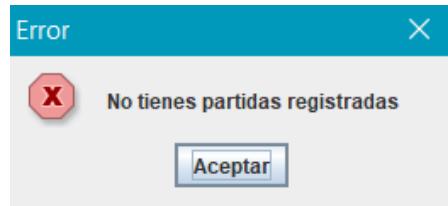


Ilustración 3

JUGADOR	PUNTUACION	FECHA

Ilustración 4

2.3.4 Ranking mejores jugadores

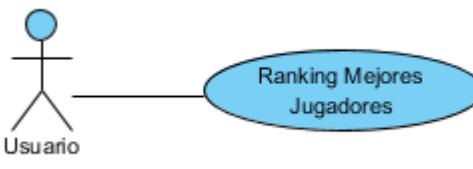
	
Nombre:	Ranking Mejores Jugadores
Descripción: El usuario podrá visualizar el ranking de los mejores jugadores del Bar Bestial con sus puntuaciones medias.	
Actores: Usuario.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario selecciona en el menú principal la opción “Ranking” (Ilustración 1)2. Dentro del menú del “Ranking”, el usuario selecciona la opción de “Mejores Jugadores” (Ilustración 2). <p>[SI NO HAY PARTIDAS REGISTRADAS]</p> <ol style="list-style-type: none">2.1. Se muestra un mensaje (Ilustración 3) y al pulsar “Aceptar” se vuelve a la pantalla del ranking (Ilustración 2). <p>[SI HAY PARTIDAS REGISTRADAS]</p> <ol style="list-style-type: none">2.2. Se muestra la nueva ventana con los datos correspondientes al ranking de “Mejores Jugadores” (Ilustración 4).	
Postcondiciones: Ninguna.	
Interfaz Gráfica:	



Ilustración 1



Ilustración 2



Ilustración 3

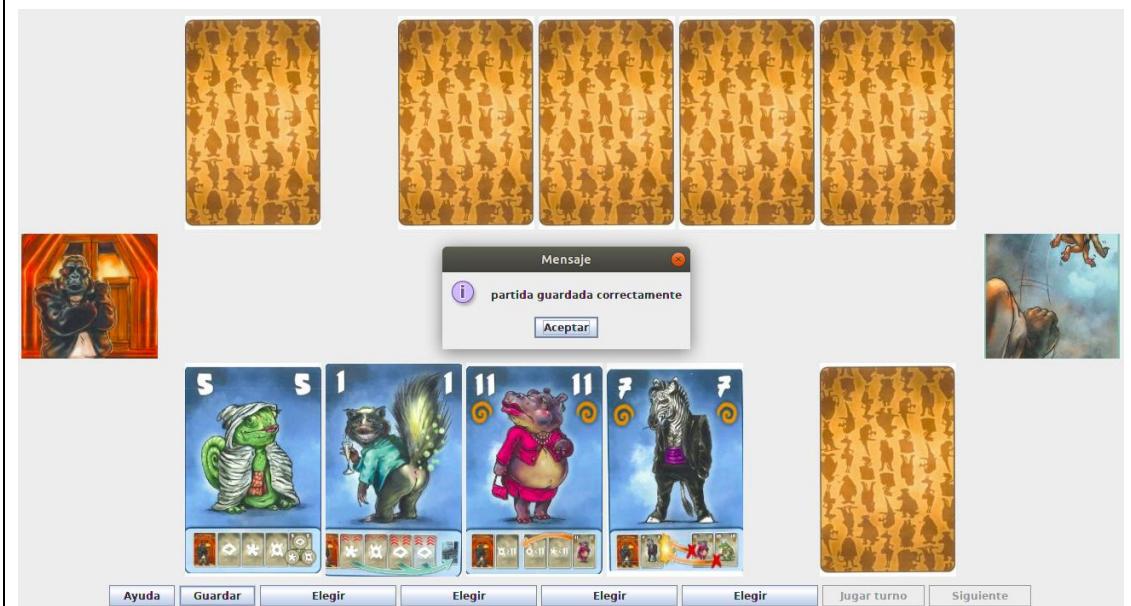


Ilustración 4

2.4 Guardar/Reanudar Partida

2.4.1 Guardar partida

Nombre: Guardar Partida
Descripción: El jugador puede guardar la partida en cualquier momento para retomarla más tarde.
Actores: Jugador.
Precondiciones: Ninguna.
Requisitos no funcionales: Ninguna.
<p>Flujo de Eventos:</p> <ol style="list-style-type: none"> 1. El jugador selecciona en la ventana de juego (Ilustración 1) el botón de guardar. [SI HA ALCANZADO EL MÁXIMO DE PARTIDAS GUARDADAS] <ol style="list-style-type: none"> 2.1 Aparece un mensaje avisando que no puede guardar más partidas. 3.1 El jugador pulsa en aceptar (Ilustración 3) y vuelve a la ventana de juego. [SI NO HA ALCANZADO EL MÁXIMO PARTIDAS GUARDADAS] <ol style="list-style-type: none"> 2.2. Aparece la pantalla para introducir el nombre de la partida (Ilustración 5). <ol style="list-style-type: none"> [SI NO INTRODUCE UN NOMBRE] <ol style="list-style-type: none"> 3.2.1 Aparece un mensaje de error (Ilustración 6) y al pulsar “Aceptar” se vuelve a la ventana de guardar nombre. [SI HA INTRODUCIDO UN NOMBRE] <ol style="list-style-type: none"> [SI YA HAY UNA PARTIDA CON ESE NOMBRE] <ol style="list-style-type: none"> 3.2.2.1 Se muestra un mensaje de error (Ilustración 7) y al pulsar “Aceptar” se vuelve a la pantalla de guardar nombre. [SI NO HAY UNA PARTIDA CON ESE NOMBRE] <ol style="list-style-type: none"> 3.2.2.2 Se guarda la partida y aparece un mensaje confirmando que la partida ha sido guardada (Ilustración 2). 4.2.2.2 El jugador pulsa en aceptar y vuelve a la ventana principal de juego (Ilustración 4).
Postcondiciones: La partida queda guardada si tiene huecos de guardado.
Interfaz Gráfica:



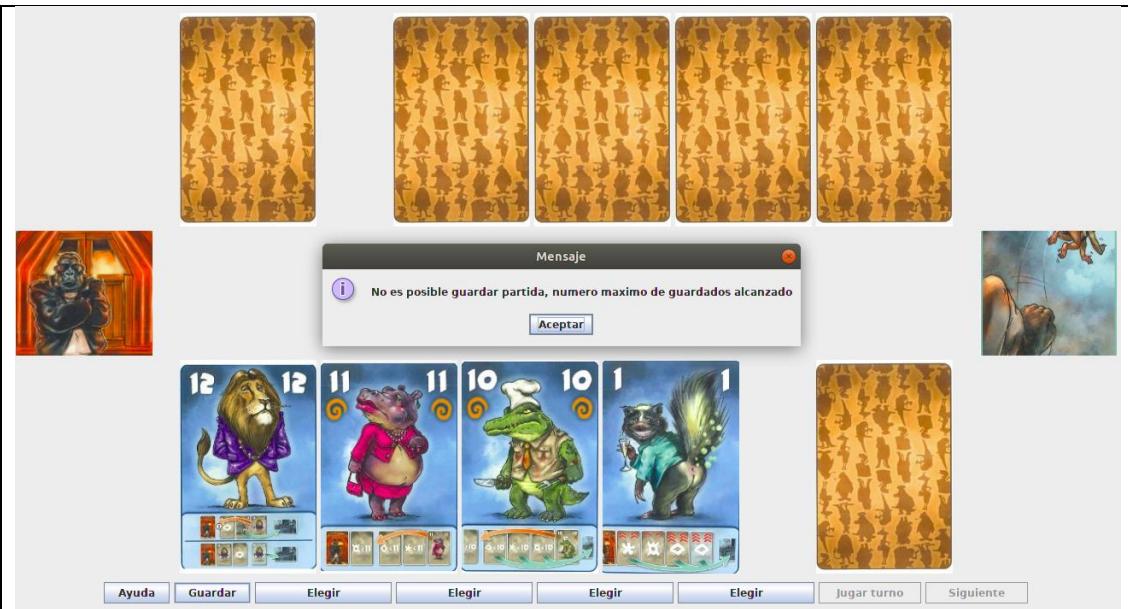


Ilustración 3



Ilustración 4

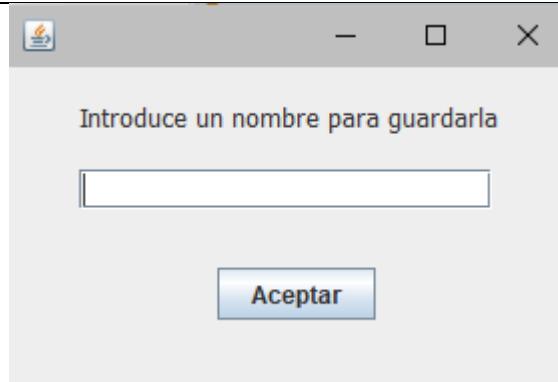


Ilustración 5

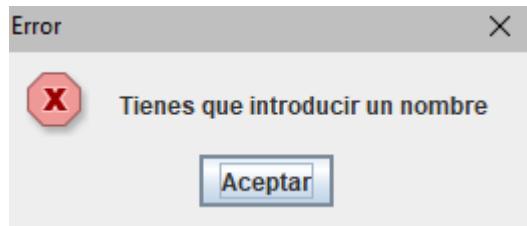


Ilustración 6

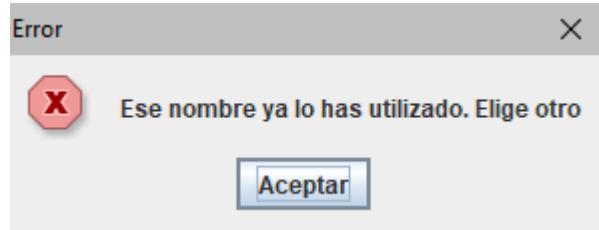


Ilustración 7

2.4.2 Reanudar partida

	
Nombre:	Reanudar Partida
Descripción: Permite al usuario reanudar una partida guardada anteriormente.	
Actores: Usuario.	
Precondiciones: El usuario tiene que tener al menos una partida guardada.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario identificado selecciona en su menú principal la opción “Jugar” (Ilustración 1).2. El usuario selecciona la opción “Reanudar Partida” (Ilustración 2).3. Se muestra una nueva ventana que le dará la opción de elegir la partida que desea reanudar (Ilustración 3). <p>[SI PULSA “ACEPTAR” SIN SELECCIONAR UNA PARTIDA]</p> <ol style="list-style-type: none">4.1. Aparece un mensaje indicando que tiene que seleccionar una partida (Ilustración 4). <p>[SI PULSA “ACEPTAR” HABIENDO SELECCIONADO UNA PARTIDA]</p> <ol style="list-style-type: none">4.2. Se abrirá la ventana de juego con la partida que el usuario ha escogido para jugar (Ilustración 5).	
Postcondiciones: Se ha reanudado la partida seleccionada y se ha borrado de las partidas guardadas.	
Interfaz Gráfica:	



Ilustración 1



Ilustración 2

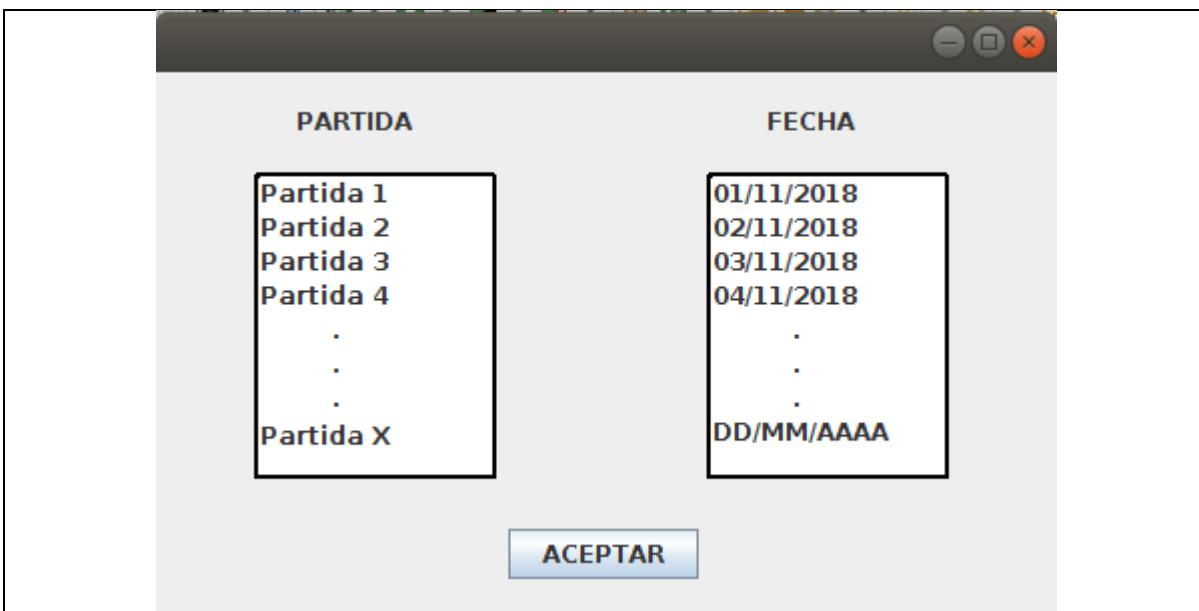


Ilustración 3

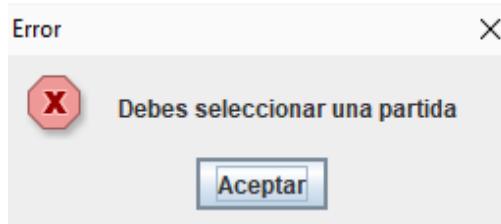


Ilustración 4



Ilustración 5

2.4.3 Iniciar Partida

	
Nombre:	Iniciar Partida
Descripción: Permite al usuario iniciar una partida nueva desde cero.	
Actores: Usuario.	
Precondiciones: Ninguna.	
Requisitos no funcionales: Ninguno.	
Flujo de Eventos: <ol style="list-style-type: none">1. El usuario identificado selecciona en su menú principal la opción “Jugar” (Ilustración 1).2. El usuario selecciona la opción “Nueva Partida” (Ilustración 2).<ol style="list-style-type: none">3.1. Se muestra al usuario una ventana con las distintas personalizaciones que tiene (Ilustración 3). [SI PULSA “ACEPTAR” SIN SELECCIONAR UNA BARAJA]<ol style="list-style-type: none">4.1. Se abrirá la ventana de juego con una partida nueva y la baraja sin personalizar (Ilustración 4).	
Postcondiciones: Se ha iniciado una nueva partida con la baraja elegida.	
Interfaz Gráfica:	



Ilustración 1



Ilustración 2

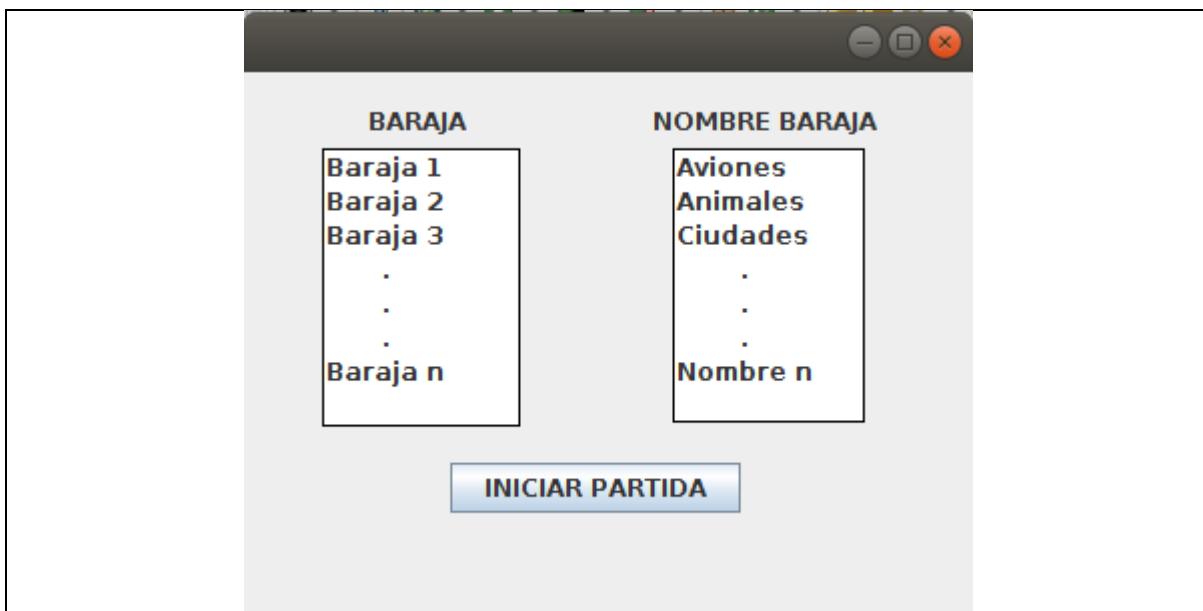


Ilustración 3

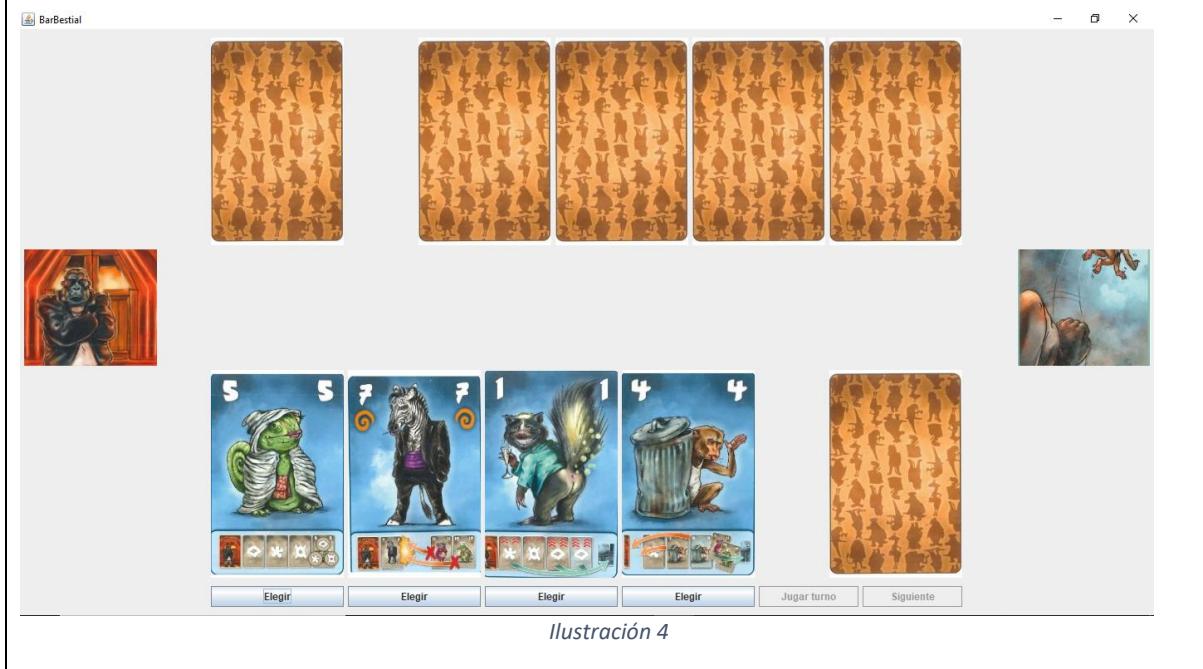


Ilustración 4

2.5 Ayuda

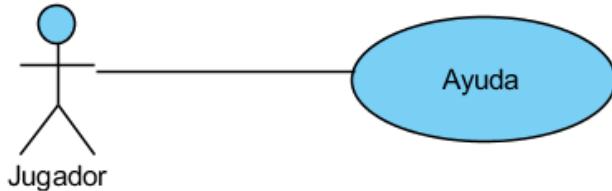

Nombre: Ayuda
Descripción: Al solicitar la ayuda permite devolver una carta del bar bestial del rival al campo.
Actores: Jugador.
Precondiciones: Ninguna.
Requisitos no funcionales: Ninguno.
Flujo de Eventos: 1. El Jugador selecciona en la ventana de juego (Ilustración 1) el botón de “Ayuda”. [SI NO TIENE PUNTOS DE AYUDA] 2.1. Saldrá un mensaje por pantalla (Ilustración 2). 3.1. El jugador pulsa “Aceptar” y vuelve a la pantalla de juego (Ilustración 1). [SI TIENE PUNTOS] [HA USADO 2 VECES LA AYUDA] 2.2. Saldrá un mensaje por pantalla (Ilustración 3). 3.2. El jugador pulsa “Aceptar” y vuelve a la pantalla de juego (Ilustración 1). [NO HA USADO 2 VECES LA AYUDA] [EL RIVAL TIENE CARTAS EN EL BAR] 2.3. Se abre una ventana en la que salen las cartas que puede elegir (Ilustración 4). 3.3. El jugador pulsa “Seleccionar” en la carta que quiere. 4.3. La carta se coloca al final de la fila. [EL RIVAL NO TIENE CARTAS EN EL BAR] 2.4. Consumo el punto de ayuda.
Postcondiciones: Si se ejecuta la ayuda gastara un punto de ayuda.
Interfaz Gráfica:



Ilustración 1

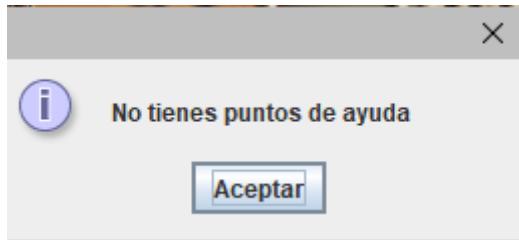


Ilustración 2

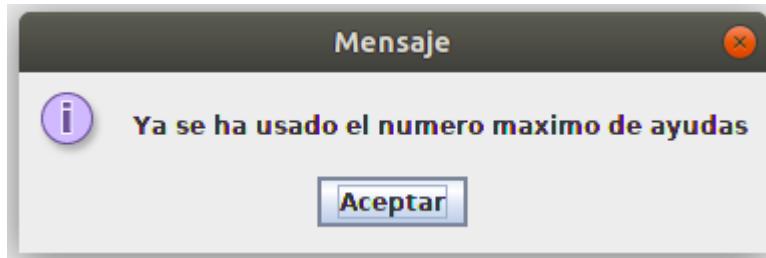


Ilustración 3

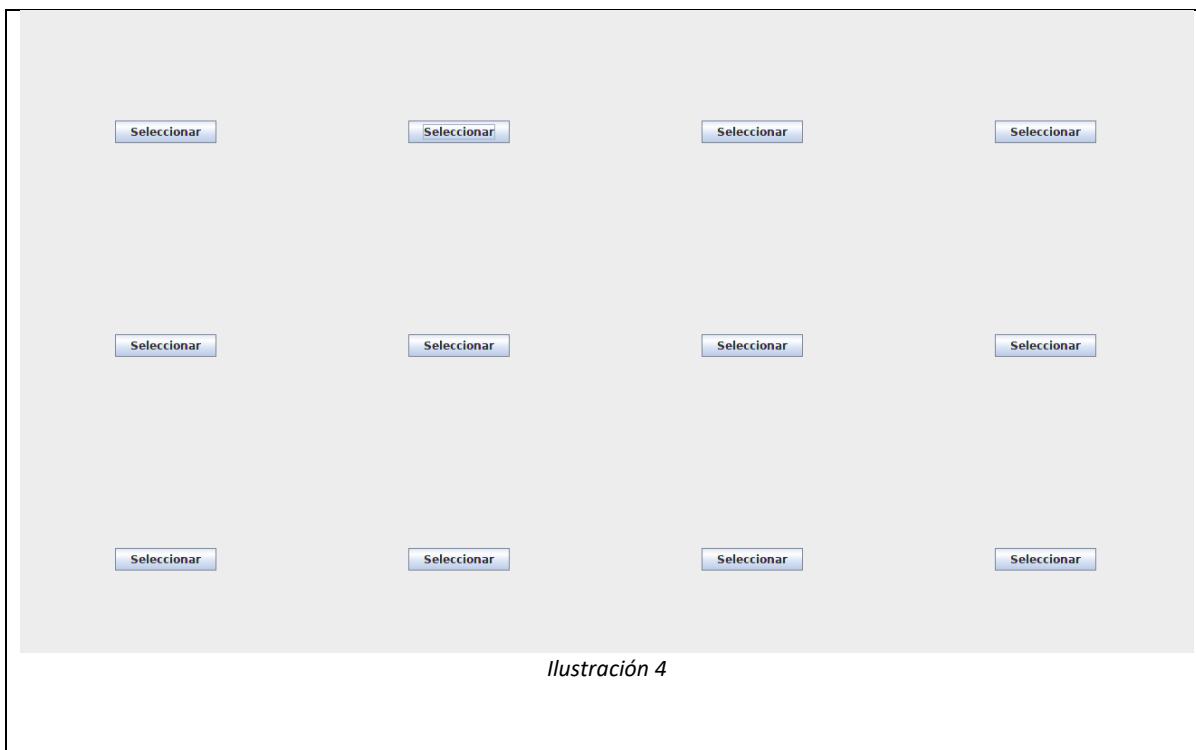


Ilustración 4

2.6 Personalización

	
Nombre:	Personalizar
<p>Descripción: Permite al usuario crear configuraciones de forma que un jugador pueda utilizar imágenes personalizadas para sus cartas y elegir con cuál de sus configuraciones quiere jugar.</p>	
<p>Actores: Usuario.</p>	
<p>Precondiciones: Ninguna.</p>	
<p>Requisitos no funcionales: Ninguno.</p>	
<p>Flujo de Eventos:</p> <ol style="list-style-type: none">1. El usuario identificado selecciona en su menú principal la opción “Personalizar Cartas” (Ilustración 1).2. Se abrirá una ventana en la que el usuario introduce el nombre de la baraja (Ilustración 5). [SI EL USUARIO NO HA INTRODUCIDO UN NOMBRE]<ol style="list-style-type: none">3.1. Se muestra un mensaje de error (Ilustración 6) y vuelve a la ventana de escoger nombre para la baraja (Ilustración 5) [SI EL USUARIO HA INTRODUCIDO UN NOMBRE] [SI EL NOMBRE DE LA BARAJA NO ESTÁ REPETIDO]<ol style="list-style-type: none">3.2. Se abrirá una nueva ventana en la que el usuario podrá elegir la carta que se desea modificar (Ilustración 2).4.2. Una vez elegida la carta, se abrirá el explorador de archivos para que seleccione la carta que desea introducir (Ilustración 3). [SI NO HA PULSADO NINGUNA CARTA Y PULSA GUARDAR]<ol style="list-style-type: none">5.1 Se muestra un mensaje de error (“Ilustración 8”), y se volverá a la ventana de elegir carta. [SI HA SELECCIONADO UNA CARTA ENTONCES]<ol style="list-style-type: none">5.2. Se vuelve a la ventana de “Personalización” y el usuario pulsa el botón guardar. [SI NO HA PERSONALIZADO LAS 12 CARTAS]<ol style="list-style-type: none">6.1. Se muestra un mensaje (Ilustración 4) y al pulsar “Aceptar” se vuelve a la pantalla de personalizar (Ilustración 2). [SI HA PERSONALIZADO LAS 12 CARTAS]<ol style="list-style-type: none">7.1. Se guarda la baraja y se regresa a la ventana del Menú Principal.	

[SI EL NOMBRE DE LA BARAJA ESTÁ REPETIDO]

3.3. Se muestra un mensaje de error (Ilustración 7) y vuelve a la ventana de escoger nombre para la baraja (Ilustración 5)

Postcondiciones: Se guarda la baraja personalizada.

Interfaz Gráfica:

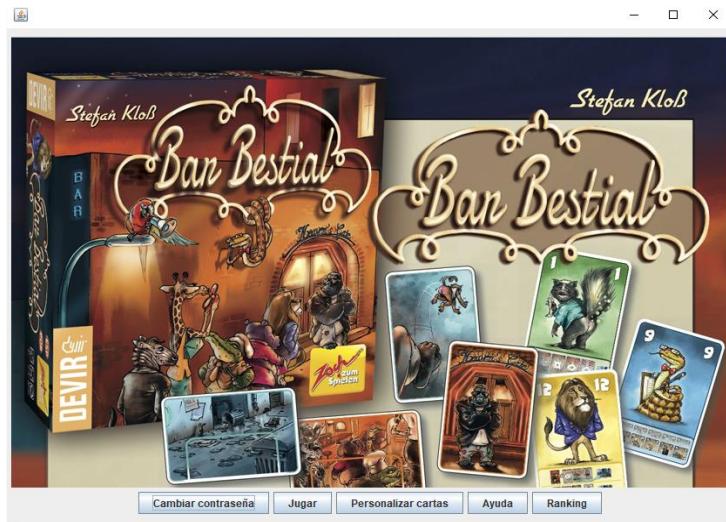


Ilustración 1



Ilustración 2

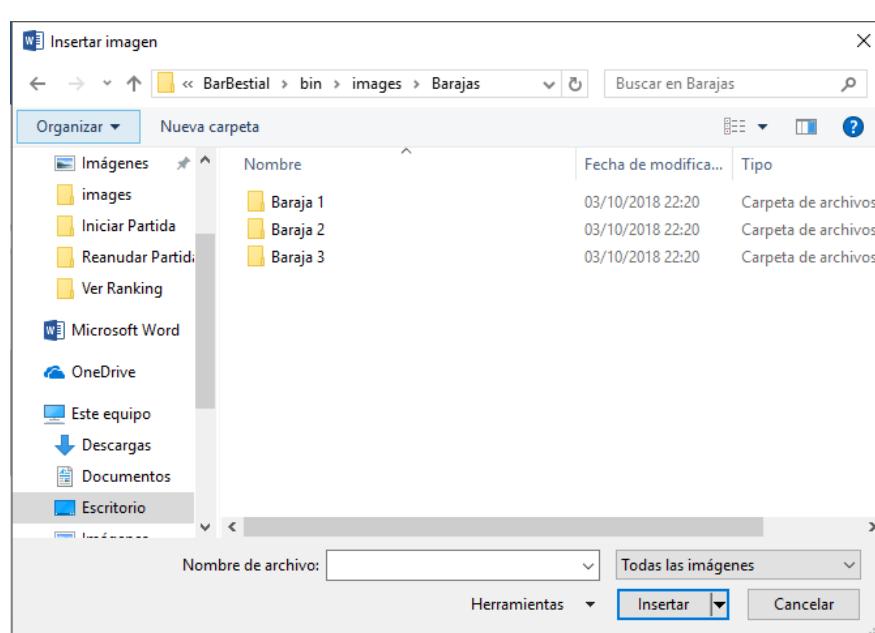
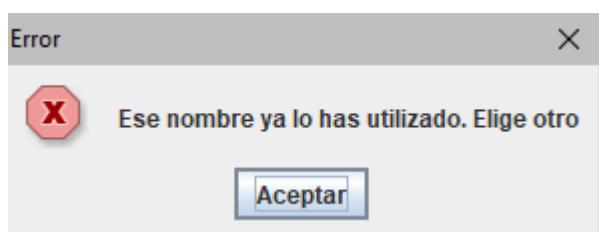
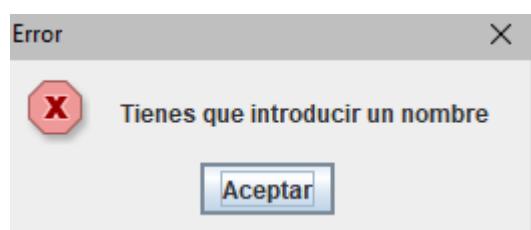
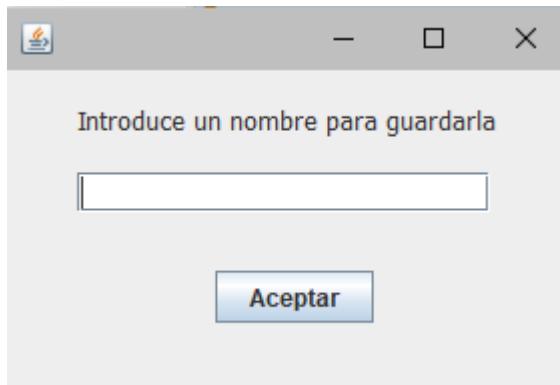
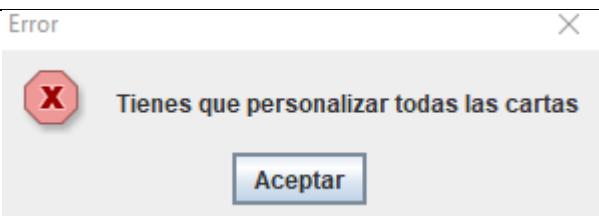


Ilustración 3



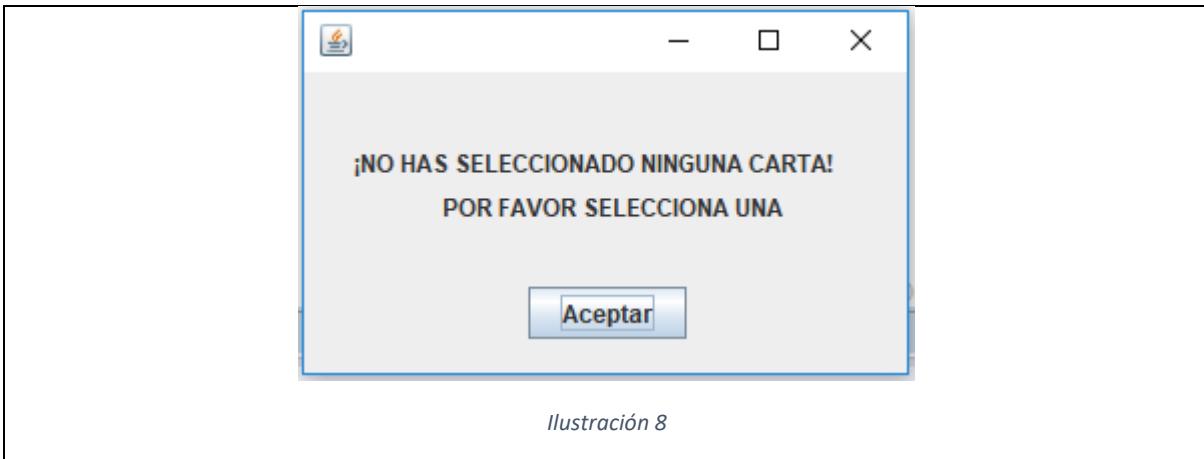
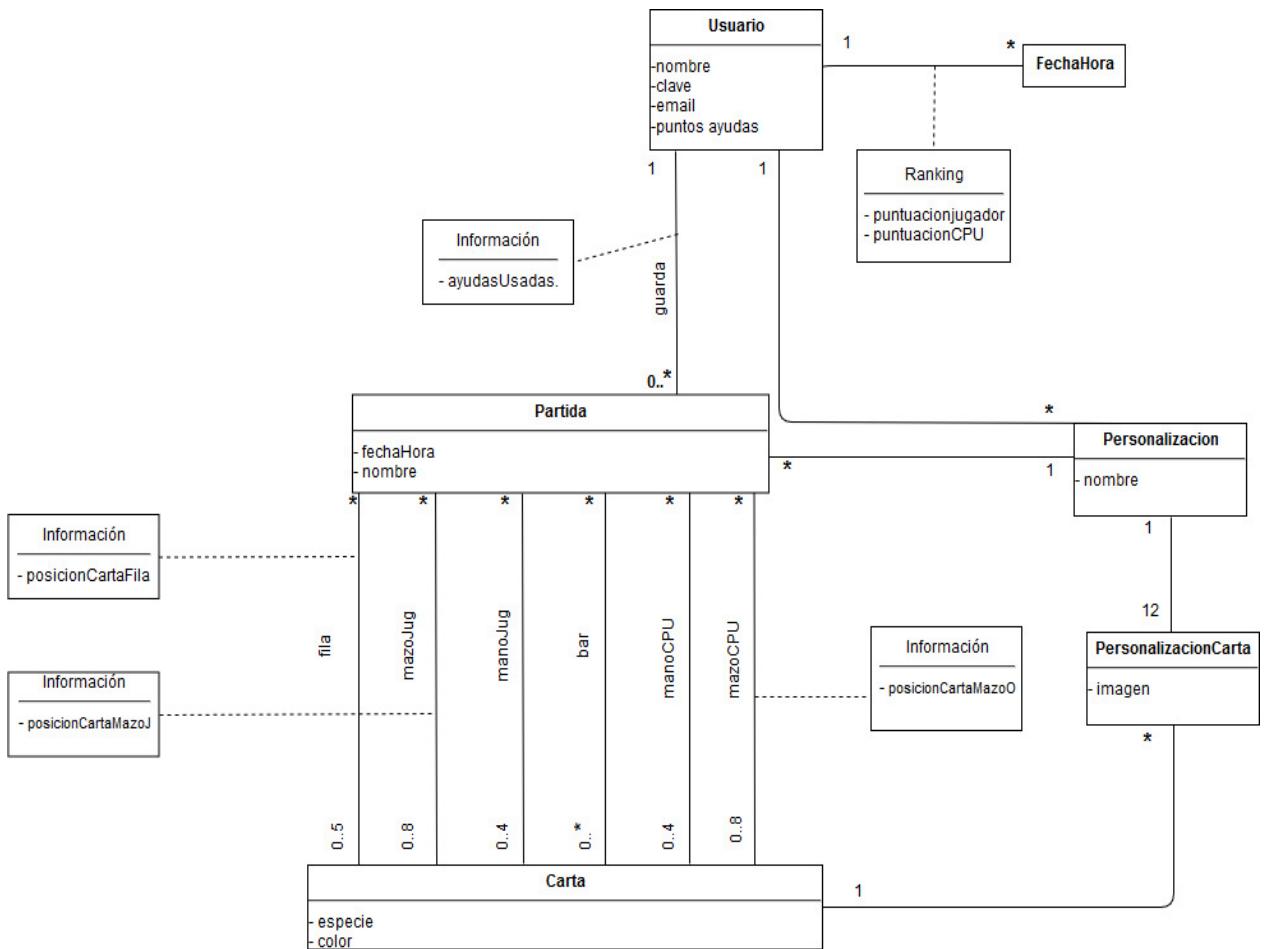


Ilustración 8

3. Modelo de Dominio

En este apartado se describe el diagrama de modelo de dominio planteado por los miembros del grupo.



ENTIDADES:

- **Usuario:** Es la persona que está identificada en el juego. Tiene capacidad para iniciar todas las partidas que quiera, capacidad de reanudar sus propias partidas siempre que quiera, cambiar su propia contraseña, personalizar las cartas... Tiene un nombre, una clave, un email y puntos de ayuda.
- **Partida:** Es el sistema al que accede usuario cuando inicia o reanuda partida. Puede ser guardada por el jugador y reanudada por el mismo jugador sólamente. Contiene el mazo del jugador, el mazo de la CPU, la mano del jugador, la mano de la CPU, la fila del bar y las cartas dentro del bar.
- **Carta:** Puede ser personalizada por el usuario, tiene un número y una imagen. Forma parte de las listas de cartas que forman el mazo del jugador, el mazo de la CPU, la mano del jugador, la mano de la CPU, la fila del bar y las cartas dentro del bar.
- **Personalización:** Hace referencia a las personalizaciones que el usuario puede hacer. Tiene como atributo un nombre y las personalizacionesCarta.
- **PersonalizacionCarta:** Tiene como atributo una imagen, ya que hace referencia a la imagen de la carta que el usuario ha cambiado.

RELACIONES:

- **fila:** Hace referencia a la fila de una partida, y esta tiene un atributo (posicionCartaFila) para que así, en caso de haber reanudado una partida, se cargue la fila en el orden en la que estaba guardada.
- **mazoJug:** Hace referencia a las cartas que tiene el jugador en el mazo, y tiene un atributo (posicionCartaMazoJ) para que en caso de reanudar una partida, se reanude con el mazo en el mismo orden.
- **manoJug:** Son las cartas que tiene el jugador en la mano para jugar un turno.
- **Bar:** Hace referencia a las cartas que han entrado en la carta “Bar Bestial”.
- **manoCPU:** Son las cartas que tiene la CPU en la mano para jugar un turno (el usuario no podrá verlas).
- **mazoCPU:** Hace referencia a las cartas que tiene la CPU en el mazo, y tiene un atributo (posicionCartaMazoO) para que en caso de reanudar una partida, se reanude con el mazo en el mismo orden.
- **Guarda:** Hace referencia a la posibilidad que tiene el usuario de guardar una partida si lo desea.

4. Plan de Pruebas

4.1 Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuarios

4.1.1 Registro

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Formato del email distinto a “a@a.com”	El usuario quedará registrado con ese email	El usuario se registra
2	Email ya registrado	Mensaje de error	Mensaje de error
3	Nombre de usuario existente	Mensaje de error	Mensaje de error
4	Contraseñas no coinciden	Mensaje de error	Mensaje de error
5	Algún campo sin rellenar	Mensaje de error	Mensaje de error
6	Todos los datos correctos	Almacenamiento del usuario con todos sus datos en la BD. Acceso al Sistema.	Usuario registrado en la BD Pantalla principal del juego
7	Seleccionar “Registrarte”	Pantalla de registro	Pantalla de registro
8	Seleccionar “Cancelar”	Pantalla de menú inicial	Pantalla de menú inicial

4.1.2 Identificación

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Usuario incorrecto y contraseña incorrecta	Mensaje de error	Mensaje de error
2	Usuario incorrecto y contraseña correcta	Mensaje de error	Mensaje de error
3	Usuario correcto y contraseña incorrecta	Mensaje de error	Mensaje de error
4	Usuario correcto y contraseña correcta	Acceso al Sistema	Pantalla principal del juego
5	Algún campo sin llenar	Mensaje de error	Mensaje de error
6	Seleccionar “Iniciar Sesión”	Pantalla de inicio de sesión	Pantalla de inicio de sesión
7	Seleccionar “Cancelar”	Pantalla de menú inicial	Pantalla de menú inicial

4.1.3 Recuperación de contraseña

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Email introducido no existe	Mensaje de error	Mensaje de error
2	Campo sin llenar	Mensaje de error	Mensaje de error
3	Datos correctos	Se envía un email al usuario con la nueva contraseña Se vuelve al menú inicial	Si el formato es correcto, se envía un email con la contraseña y se cambia en la BD. Si el formato no es correcto, aparece un mensaje de error. Se vuelve a la pantalla de menú inicial.

4	Seleccionar “Recuperar Contraseña”	Pantalla recuperación de contraseña	Pantalla recuperación de contraseña
5	Seleccionar “Cancelar”	Pantalla de menú inicial	Pantalla de menú inicial

4.1.4 Cambio de contraseña

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	La contraseña introducida no coincide con la contraseña antigua	Mensaje de error	Mensaje de error
2	Las contraseñas nuevas no coinciden	Mensaje de error	Mensaje de error
3	Algún campo sin llenar	Mensaje de error	Mensaje de error
4	Contraseña Antigua correcta Nuevas Contraseñas coinciden	Se modifica la antigua contraseña en la BD por la nueva contraseña introducida por el usuario. Se vuelve a la pantalla principal.	Se cambia la contraseña en la BD y se vuelve a la pantalla principal del juego
5	Seleccionar “Cambiar Contraseña”	Pantalla de cambio de contraseña	Pantalla de cambio de contraseña
6	Seleccionar “Cancelar”	Pantalla principal del juego La contraseña no se cambia	Pantalla principal del juego La contraseña no se ha cambiado

4.2 Visualización del ranking

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Se selecciona la opción “Ranking Mejores Partidas Usuario”	Se visualizará una pantalla con las mejores partidas de ese usuario con sus puntuaciones	Pantalla con las mejores partidas del usuario y sus respectivas puntuaciones. Ordenado de mayor a menor.
2	Se selecciona la opción “Ranking Mejor Puntuación Dia”	Se visualizará una pantalla con la mejor puntuación de ese día junto con el jugador que la haya obtenido	Pantalla con la mejor puntuación del día actual y el nombre del jugador que la obtuvo. Ordenado de mayor a menor.
3	Se selecciona la opción “Ranking Mejores Partidas”	Se visualizará una pantalla con las mejores partidas jugadas desde el inicio junto con los jugadores correspondientes, las puntuaciones que obtuvieron y la fecha de la partida	Pantalla con las mejores partidas desde el inicio con información de sus jugadores, puntuaciones y fecha. Ordenado de mayor a menor.
4	Se selecciona la opción “Ranking Mejores Jugadores”	Se visualizará una pantalla con los jugadores con mejor puntuación media, considerando las partidas desde el inicio	Pantalla con los mejores jugadores y sus respectivas medias. Ordenado de mayor a menor.
5	Se selecciona la opción “Ranking Mejores Partidas Usuario” y el usuario todavía no tiene una partida finalizada	Se visualizará un mensaje indicando que todavía no tiene partidas en el ranking	Mensaje de Error “No tienes partidas finalizadas”.
6	Se selecciona la opción “Ranking Mejor”	Se visualizará un mensaje indicando que	Mensaje de Error “No hay partidas finalizadas en este

	Puntuación Día ” y todavía no hay partidas registradas para ese dia	todavía no hay partidas registradas ese dia	día”
7	Se selecciona la opción “Ranking Mejores Partidas” y todavía no hay ninguna partida registrada	Se visualizará un mensaje indicando que todavía no hay partidas registradas	Mensaje de Error “No hay partidas finalizadas”
8	Se selecciona la opción “Ranking Mejores Jugadores” y todavía no hay ninguna partida registrada	Se visualizará un mensaje indicando que todavía no hay partidas registradas	Mensaje de Error “No hay partidas finalizadas”

4.3 Guardar/Reanudar Partida

4.3.1 Guardar Partida

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Seleccionar “Guardar” y alcanzado el número máximo de partidas que se pueden guardar	Mensaje indicando que no se pueden guardar más partidas	
2	Seleccionar “Guardar” y no alcanzado el número máximo de partidas que se pueden guardar	Pantalla indicando que tiene que introducir un nombre	
3	Introduce un nombre y no existe otra partida con ese nombre	La partida queda guardada Mensaje indicando que la partida se ha guardado	
4	Introduce un nombre y existe otra partida con ese nombre	Mensaje de error indicando que ya existe una partida con ese nombre	
5	No introduce ningún nombre y pulsa “Aceptar”	Mensaje de error indicando que tiene que introducir un nombre	
6	Selecciona “Guardar” y el bar está vacío	Se guarda la partida y en la tabla “Bar” de la base de datos no se guarda nada	
7	Selecciona “Guardar” y la fila está vacía	Se guarda la partida y en la tabla “Fila” de la base de datos no se guarda nada	
8	Selecciona “Guardar” y la mano del jugador está vacía	No se puede guardar la partida porque no tiene cartas en la mano.	
9	Selecciona “Guardar” y el mazo del jugador está	Se guarda la partida y en la tabla “MazoJug”	

	vacío	de la base de datos no se guarda nada	
10	Selecciona “Guardar” y el mazo del ordenador está vacío	Se guarda la partida y en la tabla “MazoOrd” de la base de datos no se guarda nada	
11	Selecciona “Guardar” y la mano del ordenador está vacía pero la del jugador no	Se guarda la partida y en la tabla “ManoOrd” de la base de datos no se guarda nada	

4.3.2 Reanudar Partida

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Seleccionar “Jugar”	Menú con las opciones de reanudar partida o iniciar una nueva	
2	Seleccionar “Reanudar Partida”	Menú con las diferentes partidas guardadas y sus respectivas fechas	
3	Seleccionar la partida que se quiere reanudar y pulsar “Aceptar”	Ventana principal del juego con los datos de la partida elegida cargados	
4	Pulsar “Aceptar” sin tener una partida seleccionada	Mensaje de error indicando que hay que seleccionar una partida	

4.3.3 Iniciar Partida

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Seleccionar “Jugar”	Menú con las opciones de reanudar partida o iniciar una nueva	
2	Seleccionar “Nueva Partida”	Menú con las diferentes barajas disponibles y sus respectivos nombres	
3	Seleccionar la baraja que se quiere utilizar y pulsar “Aceptar”	Ventana principal del juego con los datos de una nueva partida cargada con la baraja elegida	
4	Pulsar “Aceptar” sin tener una baraja seleccionada	Ventana principal del juego con los datos de una nueva partida cargada y la baraja sin personalizar	

4.4 Ayuda

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Pedir usar la ayuda sin disponer de puntos	Mensaje avisando al usuario "no hay puntos"	
2	Pedir usar la ayuda habiendo usado la ayuda 2 veces	Mensaje avisando al usuario "Ya usadas las 2 ayudas"	
3	Pedir usar la ayuda teniendo puntos,no usada la ayuda 2 veces, y el rival tiene cartas en el bar con la fila no llena	La carta seleccionada se coloca en la última posición disponible de la fila, gasta punto de ayuda	
4	Pedir usar la ayuda teniendo puntos,no usada la ayuda 2 veces, y el rival no tiene cartas en el bar	Gasta punto de ayuda	

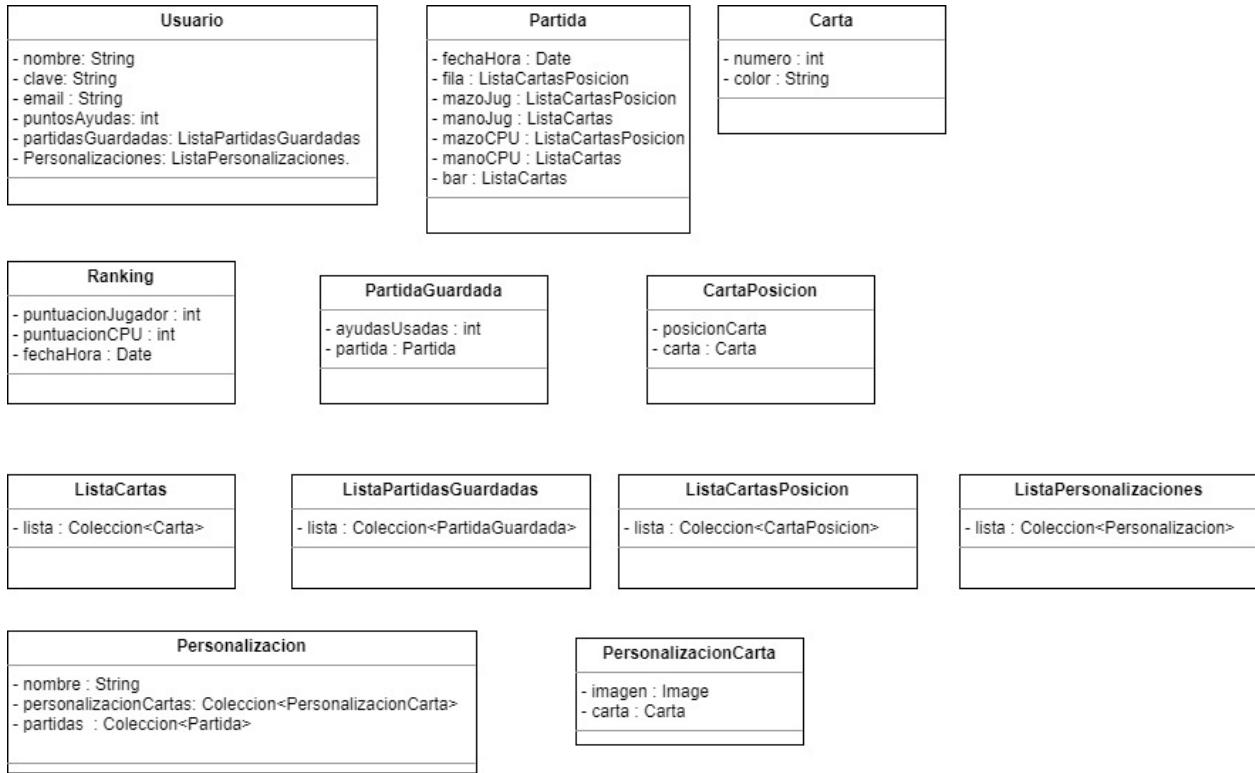
4.5 Personalización

Código de Prueba	Descripción	Resultado esperado	Resultado obtenido
1	Selecciona “Personalizar cartas”	Aparece la pantalla para introducir un nombre	Aparece la pantalla para introducir un nombre
2	Selecciona una carta a modificar	Se abre el explorador de archivos	Se abre el explorador de archivos
3	Selecciona una imagen del explorador de archivos para insertar en la carta de la baraja.	La imagen se cambiará en la imagen seleccionada de la baraja y el botón quedará inhabilitado.	Se ha cambiado la imagen en la baraja, y el botón se ha inhabilitado.
4	Selecciona “Guardar” sin haber personalizado todas las cartas	Aparece un mensaje de error y vuelve a la pantalla de personalización de baraja	Aparece un mensaje de error y vuelve a la pantalla de personalización de baraja
5	Selecciona “Guardar” habiendo personalizado todas las cartas	Se ha creado la baraja y vuelve a la ventana del “Menú Principal”	Se ha creado la baraja y vuelve a la ventana del “Menú Principal”
6	Escribe el nombre de la baraja y pulsa aceptar	Se guarda el nombre de la baraja y su path en la BD y se abre la pantalla de personalización de la baraja.	Se guarda el nombre de la baraja y su path en la BD y se abre la pantalla de personalización de la baraja.
7	Si no escribe nada en la ventana de seleccionar el nombre de la baraja y pulsa aceptar	Devuelve un mensaje de error y regresa a la pantalla de selección del nombre	Mensaje de error y vuelta a la ventana “Selección Nombre”
8	Sale sin aceptar el nombre	Vuelve a la pantalla principal del juego	Vuelve a la pantalla principal del juego
9	Sale sin guardar	No se guarda ningún cambio	Se borra la baraja creada en la carpeta de las barajas y los datos del path y el nombre de la BD.

10	Si el nombre de la baraja ya estaba guardado	Devuelve un mensaje de error y regresa a la pantalla de “Insertar Nombre Baraja”	Devuelve un mensaje de error y regresa a la pantalla de “Insertar Nombre Baraja”
11	Si el usuario selecciona un formato diferente a JPG o PNG para insertar en la carta	Devuelve un mensaje de error y regresa a la pantalla de “Seleccionar Carta”	

5. Diagrama de clases

5.1 Diagrama de clases y base de datos a partir del modelo de dominio



Este diagrama de clases se ha obtenido a partir del modelo de dominio y sólo trabaja con objetos. Nosotros hemos decidido modificar el de año pasado añadiendo las nuevas funcionalidades y una base de datos.

Algunos atributos como el atributo “email” de la clase Usuario y el atributo “ayudasUsadas” de la clase PartidaGuardada se han añadido al diagrama de clases final como atributos de las clases JugadorReal y Partida respectivamente. También se han añadido las clases Personalizacion y PersonalizacionCarta.

Los demás datos que no se han puesto en el diagrama de clases final, se guardarán en la base de datos.

BASE DE DATOS:

Usuario (email, nombre, clave, puntosAyudas)

Partida (nombrePartida, fechaHora, emailUsuario, ayudasUsadas, nombrePersonalizacion)

Carta (especie,color)

Ranking (id, fecha, emailUsuario, puntosJug, puntosCPU)

Fila (nombrePartida,especie,color, posicion)

MazoJug (nombrePartida,especie,color, posicion)

MazoOrd (nombrePartida,especie,color, posicion)

ManoJug (nombrePartida.especie,color)

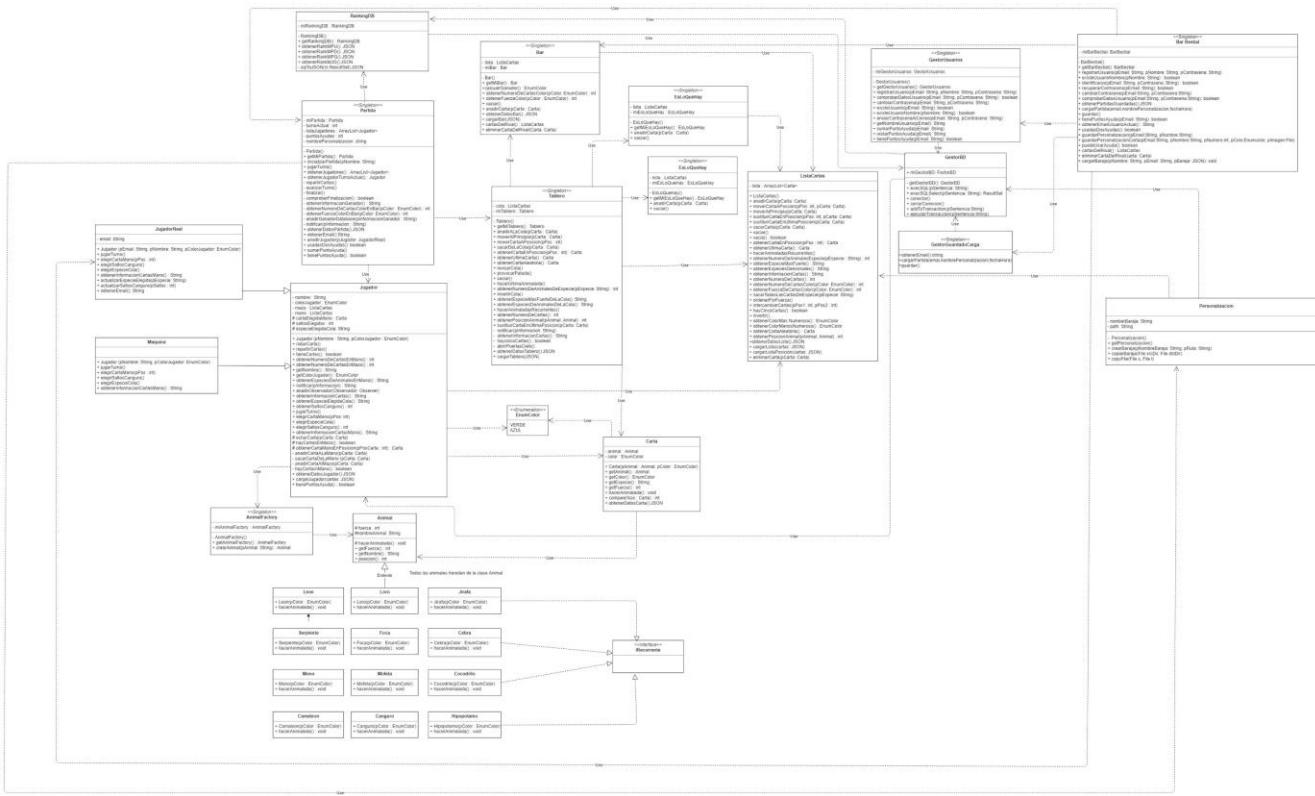
ManoOrd (nombrePartida.especie,color)

Bar (nombrePartida.especie,color)

Personalización (emailUsuario, nombrePersonalizacion, path)

PersonalizacionCarta(nombrePersonalizacion,emailUsuario, especie, color, imagen)

5.2 Diagrama de clases final



ENLACE A EL DIAGRAMA DE CLASES

https://drive.google.com/open?id=1XNxToaj3I8GE0BAR_v_-gW_29GcQf7L

Se ha añadido a la clase Partida el atributo “puntosAyudas”, que representa el número de veces que el usuario ha utilizado la ayuda en esa partida y el atributo “nombrePersonalización”. Este último se usará para saber que tipo de personalización ha usado el usuario en el caso de guardar dicha partida.

Los métodos de la clase “RankingDB” se han modificado debido a que los rankings que hay que visualizar con las nuevas funcionalidades son diferentes.

Se ha añadido una clase “GestorBD”, la cual será la encargada de conectarse con la base de datos y ejecutar las instrucciones que se le envían con el método “execSQL(pSentencia: string)”.

La clase “GestorUsuarios” que se ha añadido al diagrama de clases sirve para obtener datos de los usuarios/jugadores como el email, la contraseña, los puntos de ayuda que tiene etc.

Se ha añadido también una clase “GestorGuardaCarga”, la cual será la encargada de:

- Guardar una partida que un usuario ha iniciado, guardado el orden del mazo, el orden de la mano, el orden de la fila y la baraja con la que el usuario ha iniciado la partida.
- Reanudar una partida que un usuario había iniciado anteriormente, cargado la partida con el mismo orden del mazo, de la mano, de la fila y la baraja con la que el usuario había iniciado la partida

Se han añadido al diagrama la clase “Personalizacion”, la cual representa la personalizaciones que el jugador está usando en la partida.

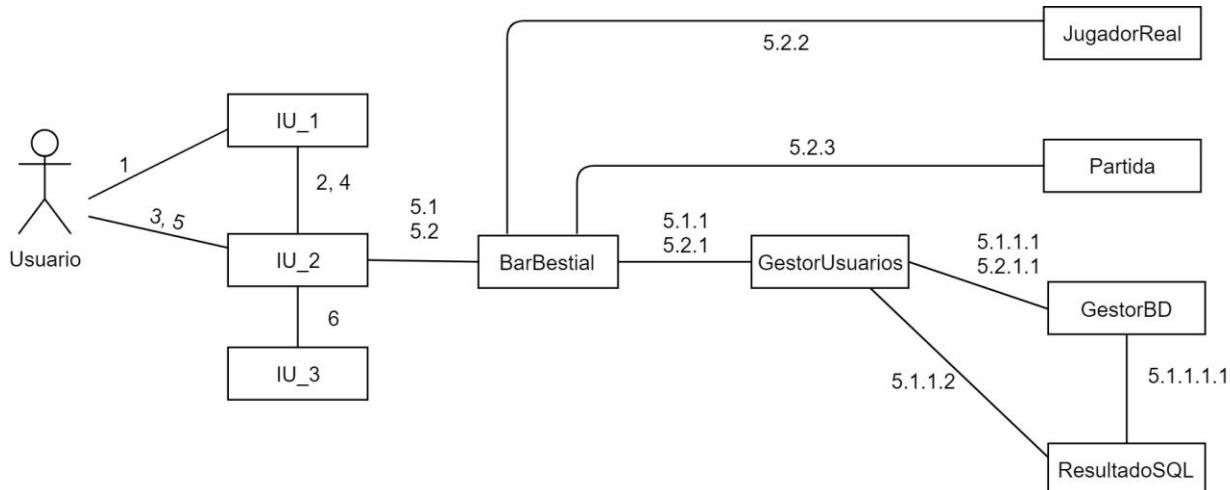
5. Diagramas de comunicación

5.1 Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuarios

* El número de las interfaces hace referencia a las interfaces que aparecen en el flujo de eventos correspondiente.

5.1.1 Registro

Diagrama de comunicación

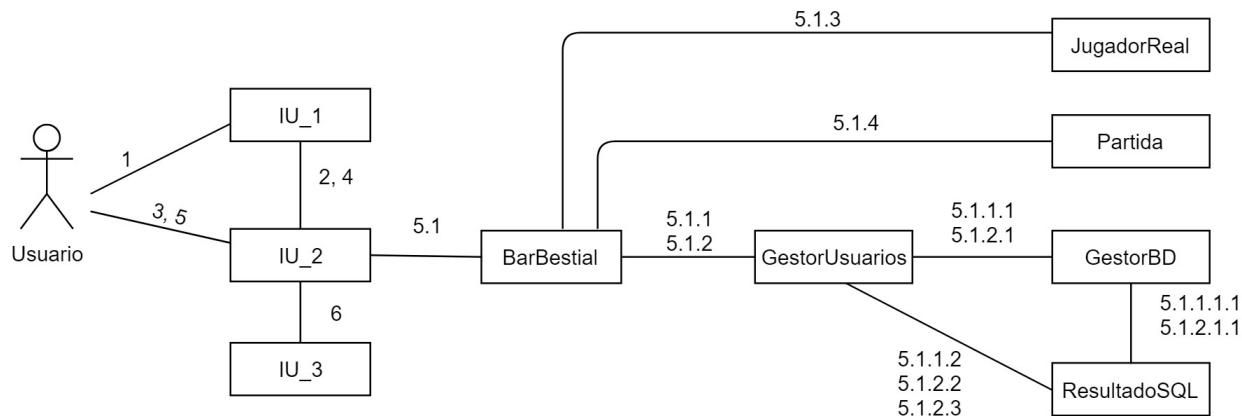


Flujo de eventos

1. El usuario pulsa "Registrarse"
2. new IU_2()
3. El usuario pulsa "Cancelar"
4. new IU_1()
5. El usuario rellena los datos y pulsa "Aceptar"
 - 5.1. existeUsuarioNombre(nombre): boolean
 - 5.1.1. existeUsuarioNombre(nombre): boolean
 - 5.1.1.1. execSQL("SELECT * FROM USUARIO WHERE nombre=%nombre%")
 - 5.1.1.1. new()
 - 5.1.1.2. next()
 - 5.2. registrarUsuario(email, nombre, contrasena)
 - 5.2.1. registrarUsuario(email, nombre, contrasena)
 - 5.2.1.1. execSQL("INSERT INTO USUARIO(email, nombre, clave, puntosAyudas) VALUES (%email%, %nombre%, %contrasena%,0)")
 - 5.2.2. new JugadorReal(email, nombre, azul) = elJugador
 - 5.2.3. anadirJugador(elJugador)
6. new IU_3()

5.1.2 Identificación

Diagrama de comunicación



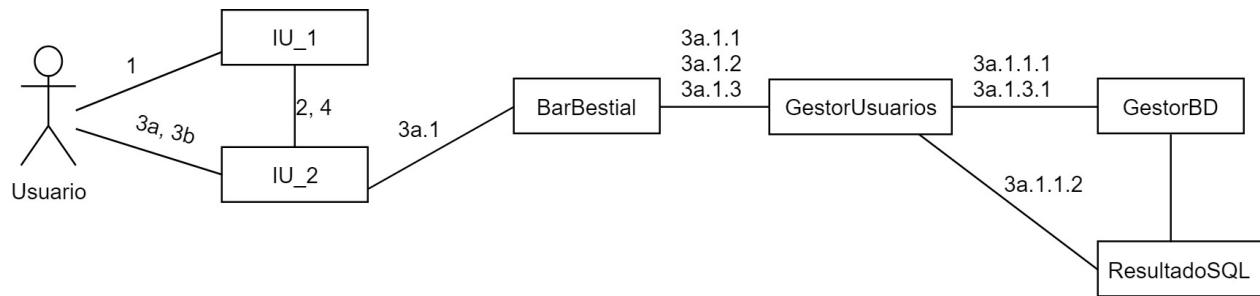
Flujo de eventos

1. El usuario pulsa "Iniciar Sesión".
2. new IU_2 ()
3. El usuario pulsa "Cancelar"
4. new IU_1 ()
5. El usuario rellena los datos y pulsa "Aceptar"
 - 5.1. identificarse(email, contrasena): boolean
 - 5.1.1. comprobarDatosUsuario (email, contrasena): boolean
 - 5.1.1.1. execSQL ("SELECT * FROM USUARIO WHERE email =% email% AND clave = %contrasena%")
 - 5.1.1.1.1. new()
 - 5.1.1.1.2. next()
 - 5.1.2. getNombreUsuario(email): String
 - 5.1.2.1. execSQL("SELECT nombre FROM USUARIO WHERE email=%email%")
 - 5.1.2.1.1. new()
 - 5.1.2.2. next()
 - 5.1.2.3. getString("nombre")
 - 5.1.3. new JugadorReal(email, nombre, azul) = elJugador
 - 5.1.4. anadirJugador(elJugador)
 6. new IU_3 ()

5.1.3 Recuperación de contraseña

* Los métodos que se utilizan en “*enviarContrasenaAlCorreo(email contrasena)*” son métodos que se obtienen con JavaMail API.

Diagrama de comunicación



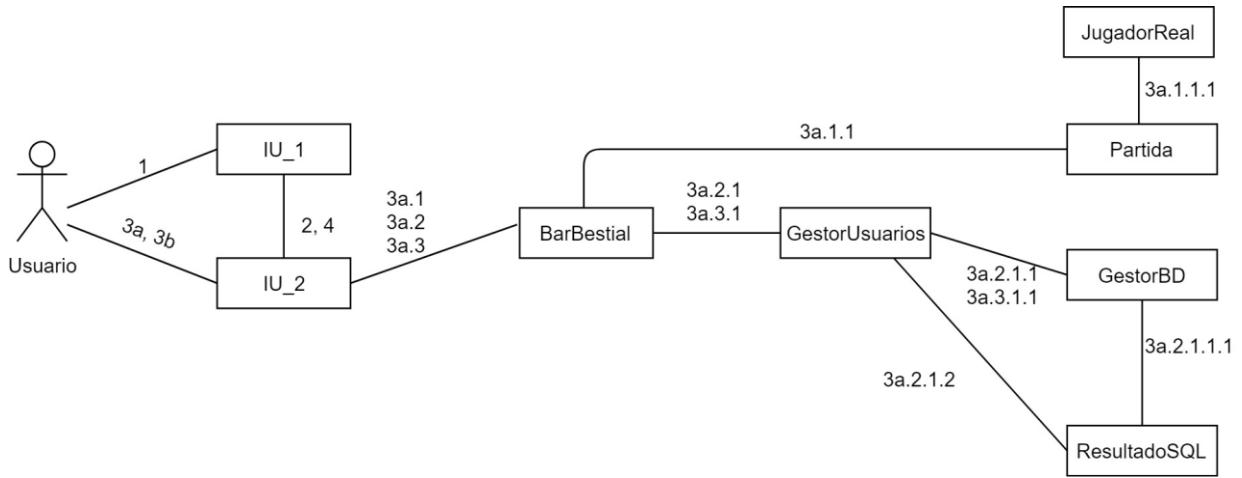
Flujo de eventos

1. El usuario pulsa "Recuperar Contraseña"
2. new IU_2 ()
- 3a. El usuario introduce los datos y pulsa "Aceptar"
 - 3a.1. recuperarContrasena(email): boolean
 - 3a.1.1. existeUsuario(email): boolean
 - 3a.1.1.1. execSQL("SELECT * FROM USUARIO WHERE email=%email%")
 - 3a.1.1.2. new()
 - 3a.1.1.3. next()
 - 3a.1.2. enviarContrasenaAlCorreo(email, contrasena) *
 - 3a.1.3. cambiarContrasena(email, contrasena) *
 - 3a.1.3.1. execSQL("UPDATE USUARIO SET clave=%contrasenaNueva% WHERE email=%email%")
 - 3b. El usuario pulsa "Cancelar"
 4. new IU_1()

* La contraseña se crea aleatoriamente y es la que se le envía al usuario mediante su correo.

5.1.4 Cambio de contraseña

Diagrama de comunicación



Flujo de eventos

1. El usuario pulsa "Cambiar contraseña"
2. new IU_2 ()
- 3a. El usuario rellena los datos y pulsa "Aceptar"
 - 3a.1. obtenerEmailUsuarioActual(): String
 - 3a.1.1. obtenerEmail(): String
 - 3a.1.1.1. obtenerEmail(): String
 - 3a.2. comprobarDatosUsuario (email, contrasenaAntigua): boolean
 - 3a.2.1. comprobarDatosUsuario(email, contrasenaAntigua): boolean
 - 3a.2.1.1. execSQL ("SELECT * FROM USUARIO WHERE email=%email% AND clave=%contrasena%")
 - 3a.2.1.1.1. new()
 - 3a.2.1.2. next()
 - 3a.3. cambiarContrasena(email, contrasenaNueva)
 - 3a.3.1. cambiarContrasena(email, contrasenaNueva)
 - 3a.3.1.1. execSQL("UPDATE USUARIO SET clave=%contrasenaNueva% WHERE email=%email%")
- 3b. El usuario pulsa "Cancelar"
4. new IU_1 ()

5.2 Visualización del ranking

5.2.1 Ranking mejores partidas usuario

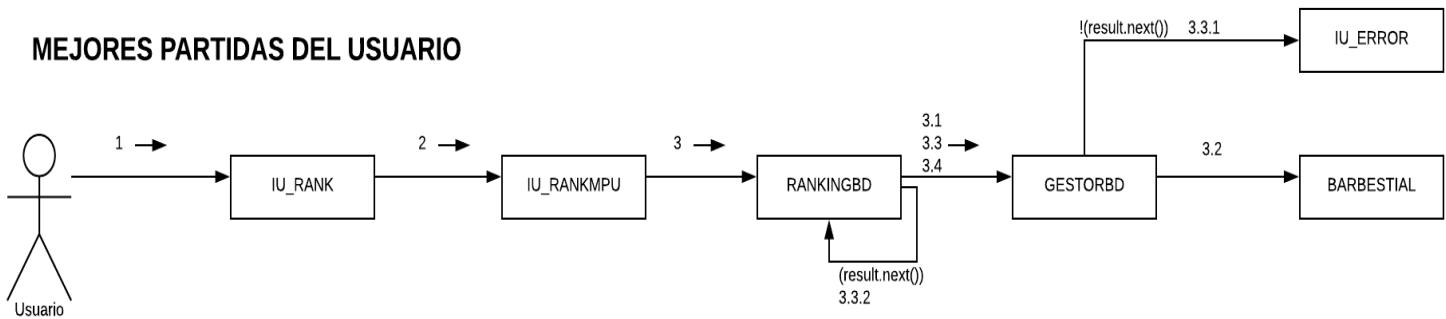


Diagrama de Comunicación de la Visualización de las Mejores Partidas del Usuario

Flujo de Eventos:

- 1: Selecciona la opción Ranking en el menú principal.
- 2: Selecciona la opción de visualizar el Ranking de Mejores Partidas del Usuario.
3. obtenerRankMPU():JSON
 - 3.1. conectar()
 - 3.2. obtenerEmailUsuarioActual():String
 - 3.3. stmt.executeQuery(pSentencia1:String):ResultSet
 - 3.3.1 Si !result.next()
Se muestra un mensaje de ERROR.
 - 3.3.2 Si result.next()
crearJSON(rs:ResultSet):JSON
 - 3.4. cerrarConexion()

JSON → { datos : { fecha : string, puntosjug : double, emailusuario : string } }

Sentencia1 → "SELECT emailUsuario,puntosJug,fecha FROM rankingdb WHERE emailUsuario = "+pEmail+" ORDER BY puntosJug DESC"

5.2.2 Ranking mejor puntuación del día

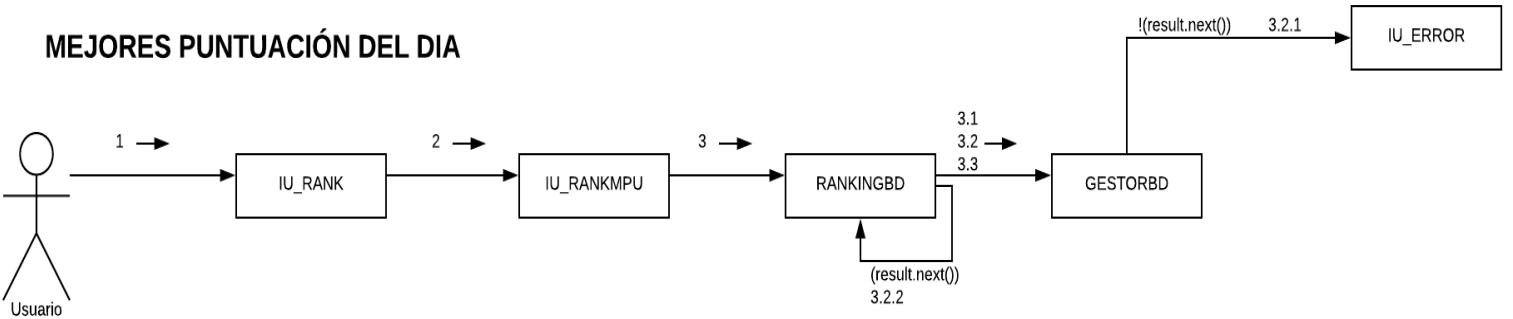


Diagrama de Comunicación de la Visualización de la Mejor Puntuación del Día del Usuario

Flujo de Eventos:

- 1: Selecciona la opción Ranking en el menú principal.
- 2: Selecciona la opción de visualizar el Ranking de Mejor Puntuación del Día.
3. obtenerRankMPD():JSON
 - 3.1. conectar()
 - 3.2. stmt.executeQuery(pSentencia1:String):ResultSet
 - 3.2.1 Si !result.next()

Se muestra un mensaje de ERROR.
 - 3.2.2 Si result.next()

crearJSON(rs:ResultSet):JSON
 - 3.3. cerrarConexion()

JSON → { datos : { puntosjug : double, emailusuario : string } }

Sentencia1 → "SELECT emailUsuario,puntosJug FROM rankingDB WHERE fecha ="+modifiedDate+" ORDER BY puntosJug DESC LIMIT 0,1"

5.2.3 Ranking mejores partidas

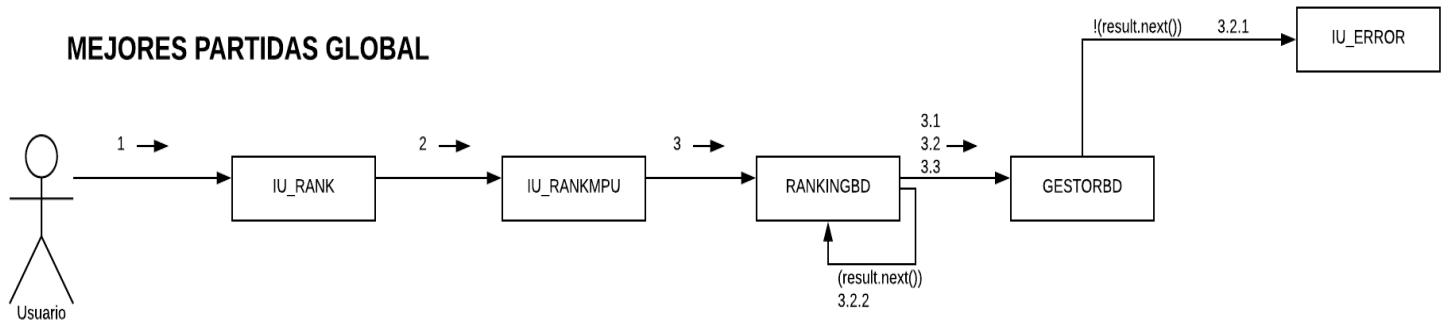


Diagrama de Comunicación de la Visualización de las Mejores Partidas Globales

Flujo de Eventos:

- 1: Selecciona la opción Ranking en el menú principal.
- 2: Selecciona la opción de visualizar el Ranking de Mejores Partidas.
3. obtenerRankMPG():JSON
 - 3.1. conectar()
 - 3.2. stmt.executeQuery(pSentencia1:String):ResultSet
 - 3.2.1 Si !result.next()

Se muestra un mensaje de ERROR.
 - 3.2.2 Si result.next()

crearJSON(rs:ResultSet):JSON
 - 3.3. cerrarConexion()

JSON → { datos : { fecha : string, puntosjug : double, emailusuario : string } }

Sentencia1 → "SELECT emailUsuario,puntosJug,fecha FROM rankingdb ORDER BY puntosJug DESC"

5.2.4 Ranking mejores jugadores

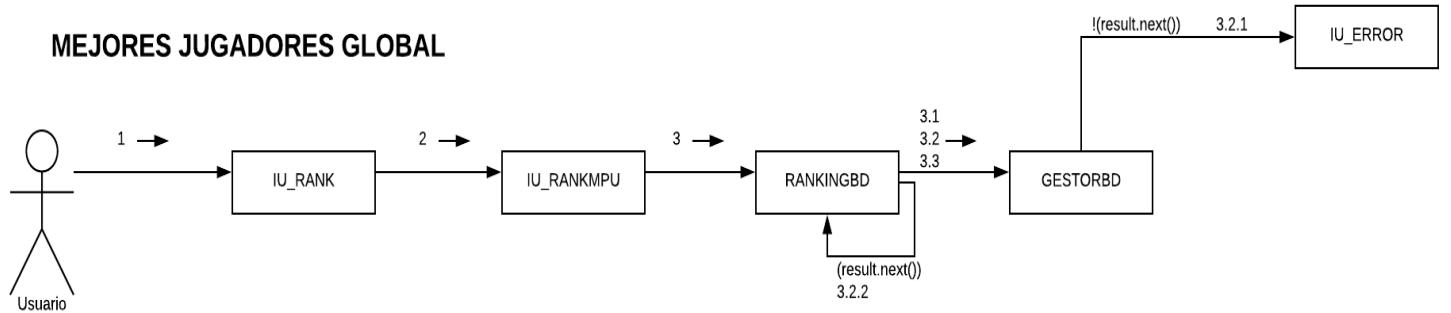


Diagrama de Comunicación de la Visualización de los Mejores Jugadores Globales

Flujo de Eventos:

- 1: Selecciona la opción Ranking en el menú principal.
- 2: Selecciona la opción de visualizar el Ranking de Mejores Jugadores.
3. obtenerRankMJG():JSON
 - 3.1. conectar()
 - 3.2. stmt.executeQuery(pSentencia1:String):ResultSet
 - 3.2.1 Si !result.next()

Se muestra un mensaje de ERROR.
 - 3.2.2 Si result.next()

crearJSON(rs:ResultSet):JSON
 - 3.3. cerrarConexion()

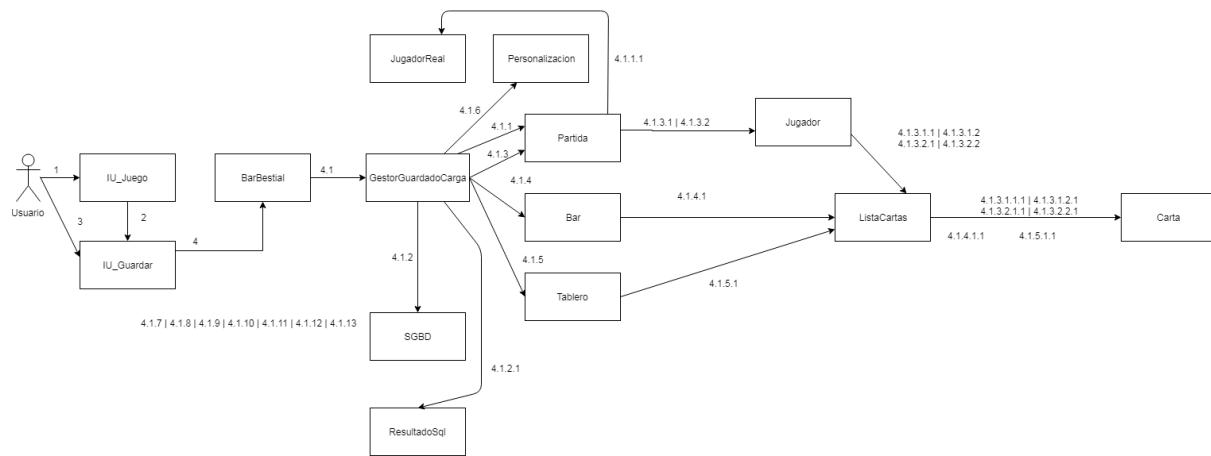
JSON → { datos : { avg(puntosjug) : double, emailusuario : string } }

Sentencia1 → "SELECT emailusuario, AVG(puntosJug) FROM rankingdb GROUP BY emailusuario ORDER BY AVG(puntosJug) DESC"

5.3 Guardar/Reanudar partida

5.3.1 Guardar partida

Diagrama de Comunicación:



Flujo De Eventos:

- 1: El usuario pulsa en el botón guardar
- 2: new_IuGuardar()
- 3: El usuario introduce el nombre con el que guardar la partida
- 4: guardar(nombrePartida:string)
 - 4.1: guardar(nombrePartida:string)
 - 4.1.1: obtenerEmail():string
 - 4.1.1.1: obtenerEmail():string
 - 4.1.2: execSql("SELECT nombrePartida FROM Partida WHERE nombrePartida=%nombrePartida% AND email=%emailUsuario%")
 - 4.1.2.1: new()
 - 4.1.3: obtenerDatosPartida():JSON /**tipo4
 - 4.1.3.1:obtenerDatosJugador():JSON /**tipo3
 - 4.1.3.1.1:obtenerDatosLista():JSON /**tipo2
 - 4.1.3.1.1.1:obtenerDatosCarta():JSON /**tipo1
 - 4.1.3.1.1.2:obtenerDatosLista():JSON /**tipo2
 - 4.1.3.1.1.2.1:obtenerDatosCarta():JSON /**tipo1
 - 4.1.3.1.2:obtenerDatosJugador():JSON /**tipo3
 - 4.1.3.2:obtenerDatosJugador():JSON /**tipo3
 - 4.1.3.2.1:obtenerDatosLista():JSON /**tipo2
 - 4.1.3.2.1.1:obtenerDatosCarta():JSON /**tipo1
 - 4.1.3.2.2:obtenerDatosLista():JSON /**tipo2
 - 4.1.3.2.2.1:obtenerDatosCarta():JSON /**tipo1
 - 4.1.4: obtenerDatosBar():JSON /**tipo2
 - 4.1.4.1:obtenerDatosLista():JSON /**tipo2

```

    4.1.4.1.1:obtenerDatosCarta():JSON /**tipo1
4.1.5: obtenerDatosTablero():JSON /**tipo2
    4.1.5.1:obtenerDatosLista():JSON /**tipo2
        4.1.5.1.1:obtenerDatosCarta():JSON /**tipo1
4.1.6: getNombre():String
4.1.7: execSql(" INSERT INTO Partida
VALUES(%nombrePartida%,%fechaHora,%emailUsuario,%ayudasUsadas,%nombrePersonalizacion)")

4.1.8: execSql(" INSERT INTO Fila VALUES(%nombrePartida,%especie%,%color,%posicion)")
4.1.9: execSql(" INSERT INTO MazoJug VALUES(%nombrePartida,%especie,%color,%posicion)")
4.1.10: execSql(" INSERT INTO MazoOrd VALUES(%nombrePartida,%especie,%color,%posicion)")
4.1.11: execSql(" INSERT INTO ManoJug VALUES(%nombrePartida,%especie,%color)")
4.1.12: execSql(" INSERT INTO ManoOrd VALUES(%nombrePartida,%especie,%color)")
4.1.13: execSql(" INSERT INTO Bar VALUES(%nombrePartida,%especie,%color)")

```

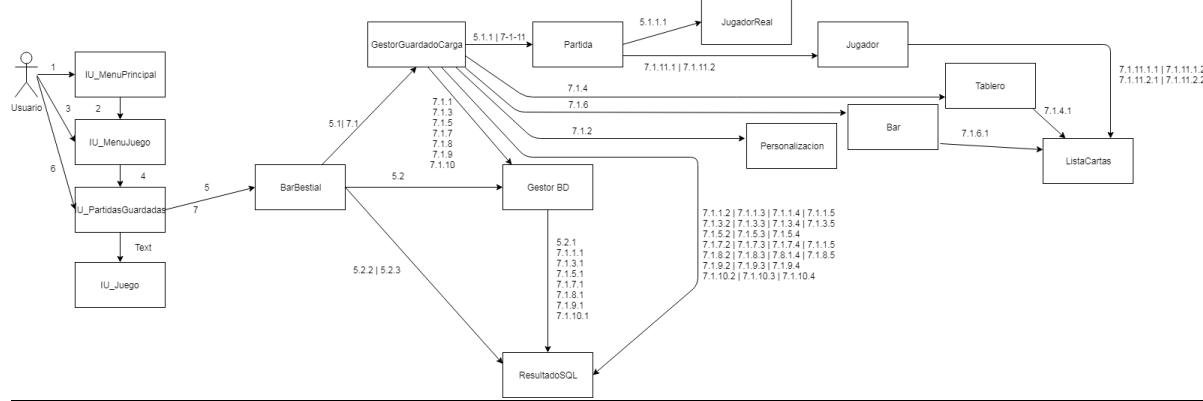
```

JSON tipo1 [especie: string, color: EnumColor]
JSON tipo2 [{tipo1:JSON,posición: int},{JSON tipo1,posición: int}....]
JSON tipo3 [tipo2:JSON,tipo2:JSON]
JSON tipo4 [tipo3: JSON , tipo3:JSON, ayudasUsadas:int]

```

5.3.2 Reanudar partida

Diagrama de comunicación



Flujo de eventos:

- 1: El usuario pulsa “Jugar”.
 - 2: newIU_MenuJugar.
 - 3: El usuario pulsa “Reanudar”.
 - 4: newIU_PartidasGuardadas()
 - 5: obtenerPartidasGuardadas():JSON /** tipo1
 - 5.1: obtenerEmail():string
 - 5.1.1: obtenerEmail():string
 - 5.1.1.1: obtenerEmail():string
 - 5.2 execSQL(“SELECT nombrePartida,nombrePersonalizacion FROM Partida WHERE emailUsuario=%email%”)
 - 5.2.1: new()
 - 5.2.2: next()
 - 5.2.3: getDate(“nombrePartida”)
 - 5.2.4: getString(“nombrePersonalizacion”)
 - 6: El usuario selecciona una partida.
 - 7: cargarPartida(email,nombrePersonalizacion,nombrePartida)
 - 7.1:cargarPartida(email,nombrePersonalizacionl,nombrePartida)
 - 7.1.2:cargarPersonalizacion(nombrePersonalizacion)
 - 7.1.3: execSQL(“SELECT especie,color,posicion FROM Fila WHERE emailUsuario=%email% AND fechaHora=%fechaHora%”)
 - 7.1.3.1: new()
 - 7.1.3.2: next()
 - 7.1.3.3: getString(“especie”)
 - 7.1.3.4: getEnum(“color”)
 - 7.1.3.5:getInt(“posicion”)

7.1.4:cargarTablero(JSON) /** JSON tipo3
 7.1.4.1: cargarListaPosicion(JSON) /** JSON tipo3
 7.1.5:execSQL("SELECT especie,color FROM Bar WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")
 7.1.5.1: new()
 7.1.5.2: next()
 7.1.5.3: getString("especie")
 7.1.5.4: getEnum("color")
 7.1.6: cargarBar(JSON) /** JSON tipo 4
 7.1.6.1: cargarLista(JSON) /** JSON tipo3
 7.1.7: execSQL("SELECT especie,color,posicion FROM MazoJugWHERE emailUsuario=%email% AND fechaHora=%fechaHora%")
 7.1.7.1: new()
 7.1.7.2: next()
 7.1.7.3: getString("especie")
 7.1.7.4: getEnum("color")
 7.1.7.5: getInt("posicion")

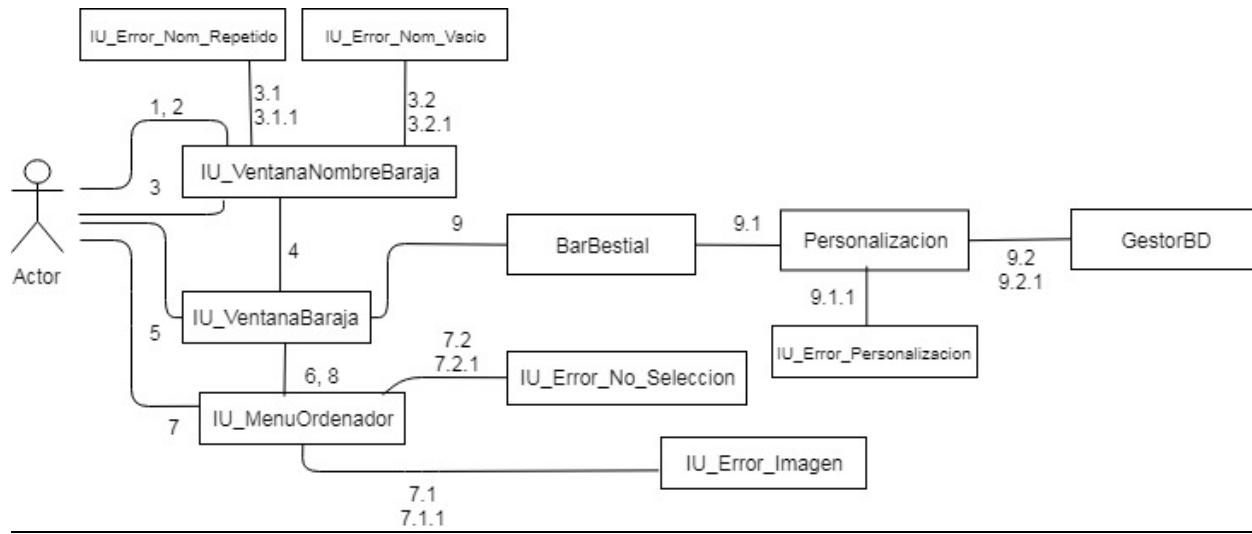
 7.1.8: execSQL("SELECT especie,color,posicion FROM MazoOrdWHERE emailUsuario=%email% AND fechaHora=%fechaHora%")
 7.1.8.1: new()
 7.1.8.2: next()
 7.1.8.3: getString("numero")
 7.1.8.4: getEnum("color")
 7.1.8.5: getInt("posicion")
 7.1.9: execSQL("SELECT especie,color FROM ManoJugWHERE emailUsuario=%email% AND fechaHora=%fechaHora%")
 7.1.9.1: new()
 7.1.9.2: next()
 7.1.9.3: getString("numero")
 7.1.9.4: getEnum("color")
 7.1.10: execSQL("SELECT especie,color FROM ManoOrd WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")
 7.1.10.1: new()
 7.1.10.2: next()
 7.1.10.3: getString("numero")
 7.1.10.4: getEnum("color")
 7.1.11: cargarJugador(JSON) /** JSON tipo5
 7.1.11.1: cargarJugador(JSON) /** JSON tipo6
 7.1.11.1.1: cargarLista(JSON) /** JSON tipo4

7.1.11.1.2: cargarListaPosicion(JSON) /** JSON tipo3
7.1.11.2: cargarJugador(JSON) /** JSON tipo6
7.1.11.2.1: cargarLista(JSON) /** JSON tipo4
7.1.11.2.1: cargarListaPosicion(JSON) /** JSON tipo3

8: newIU_Juego()
JSON tipo1 [{nombrePartida: String,nombrePersonalizacion: String},{...}...]
JSON tipo2 [{numero:int,color: EnumColor, imagen: Image},{..}...]
JSON tipo3 [{numero: int, color: EnumColor, posicion: int},{...}...]
JSON tipo4 [{numero: int, color: EnumColor},{...}...]
JSON tipo5 [{tipo3: JSON,tipo4: JSON},{tipo3: JSON,tipo4: JSON}]
JSON tipo6 [{tipo3: JSON,tipo4: JSON}]

5.5 Personalización

Diagrama de Comunicación:



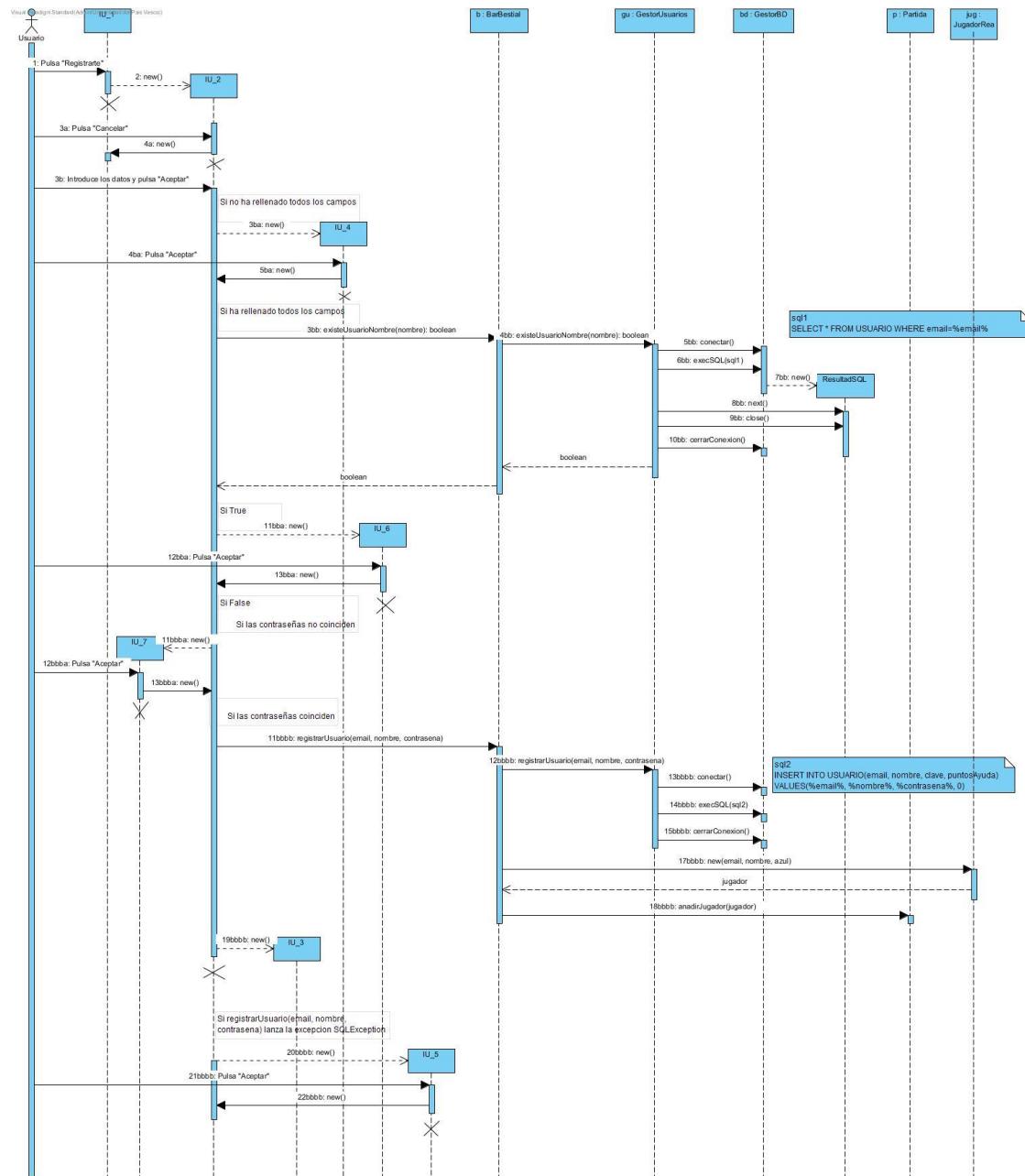
Flujo de Eventos:

1. El usuario pulsa “Personalizar”
2. new IU_VentanaNombreBaraja()
3. El usuario introduce el nombre de la baraja y pulsa Aceptar.
 - 3.1. Si el nombre de la baraja ha sido introducido anteriormente.
 - 3.1.1 new IU_Error_Nom_Repetido()
 - 3.2. Si el nombre de la baraja está vacío.
 - 3.2.1 new IU_Error_Nom_Vacio()
4. New IU_VentanaBaraja() //Esta ventana es en la que aparecen las 12 cartas para seleccionar
5. El usuario selecciona la carta que desee modificar de la baraja
6. new IU_MenuOrdenador() //Se abre el explorador de archivos
7. El usuario selecciona una imagen de su ordenador y pulsa Aceptar
 - 7.1 Si la imagen no está en tipo JPG o PNG
 - 7.1.1. new IU_Error_Imagen
 - 7.2 Si el usuario no ha seleccionado una carta
 - 7.2.1 new IU_Error_No_Seleccion
8. new IU_VentanaBaraja()
9. El usuario pulsa el botón Guardar
 - 9.1 Si no se han personalizado las 12 cartas
 - 9.1.1 IU_Error_Personalizacion
 - 9.2 Si todo ha ido correctamente
 - 9.2.1. executeSQL (INSERT INTO Personalizacion (nombrePersonalizacion, path) VALUES (pNombrePersonalizacion, pPath));

6. Diagramas de secuencia

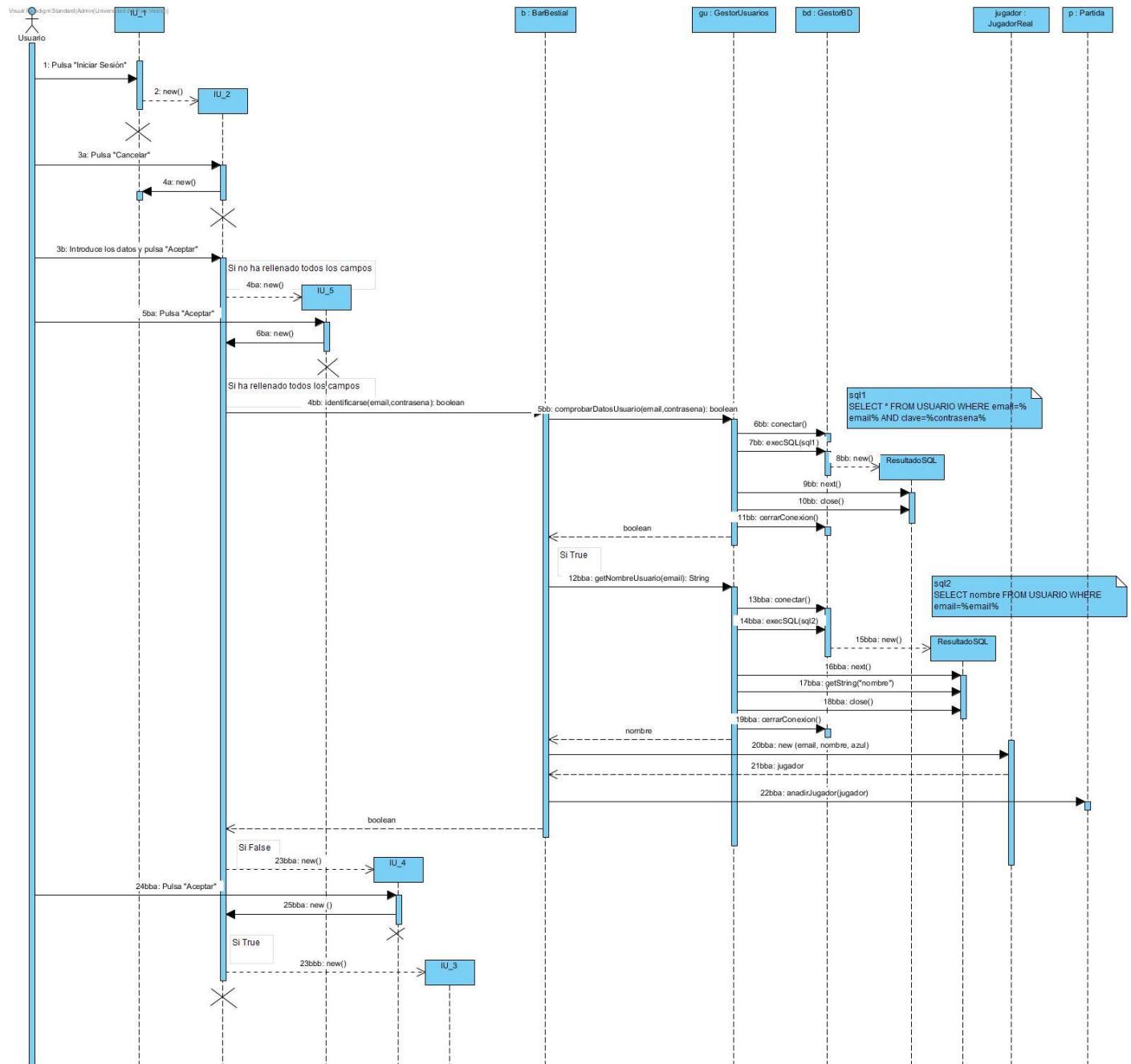
6.1 Proceso de registro e identificación, recuperación de contraseña y cambio de contraseña para todos los usuarios

6.1.1 Registro



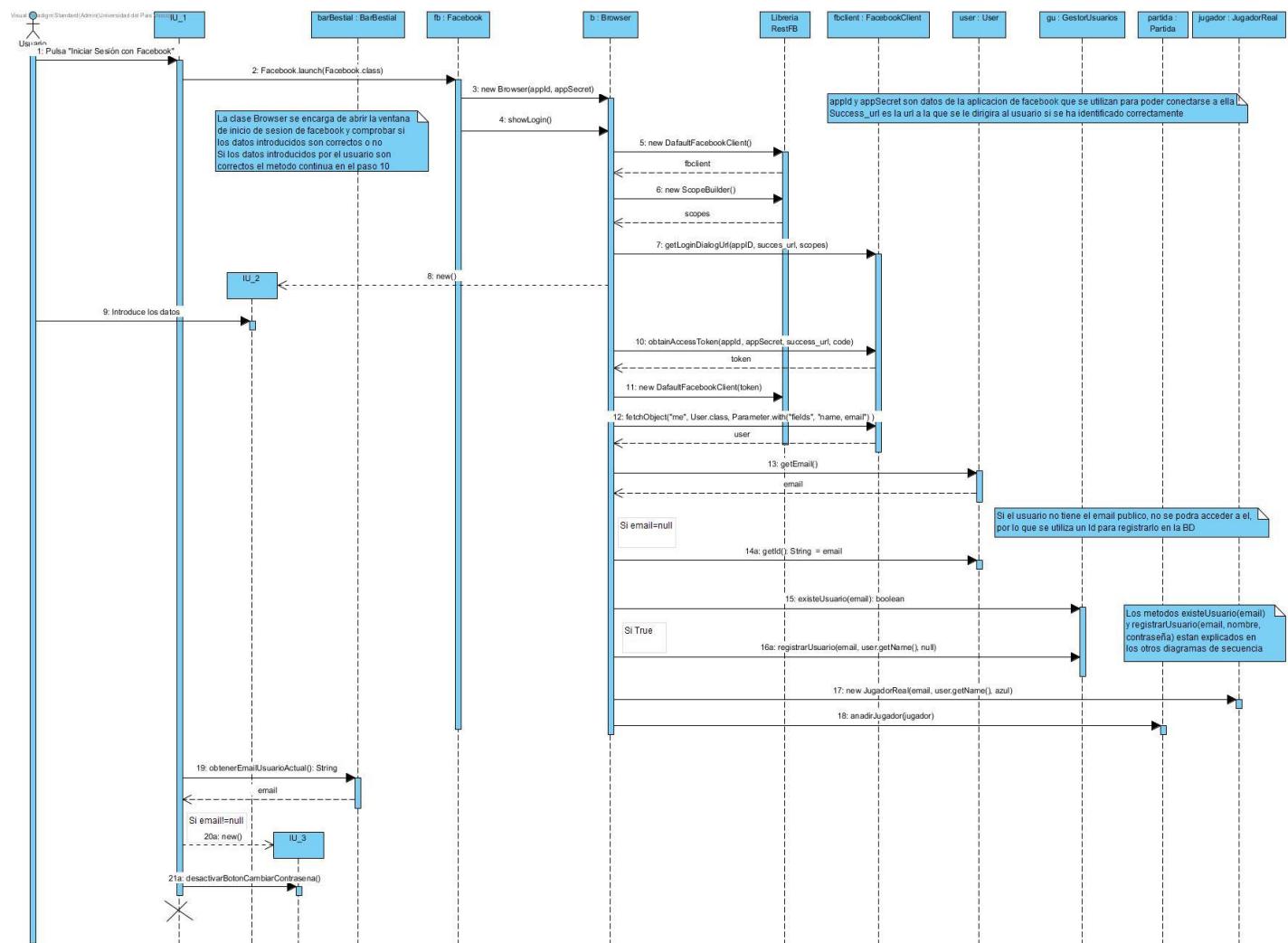
Link a la imagen: https://drive.google.com/open?id=1war5iW7QN_tSiR909G1CXnqn5FrC2s8t

6.1.2 Identificación



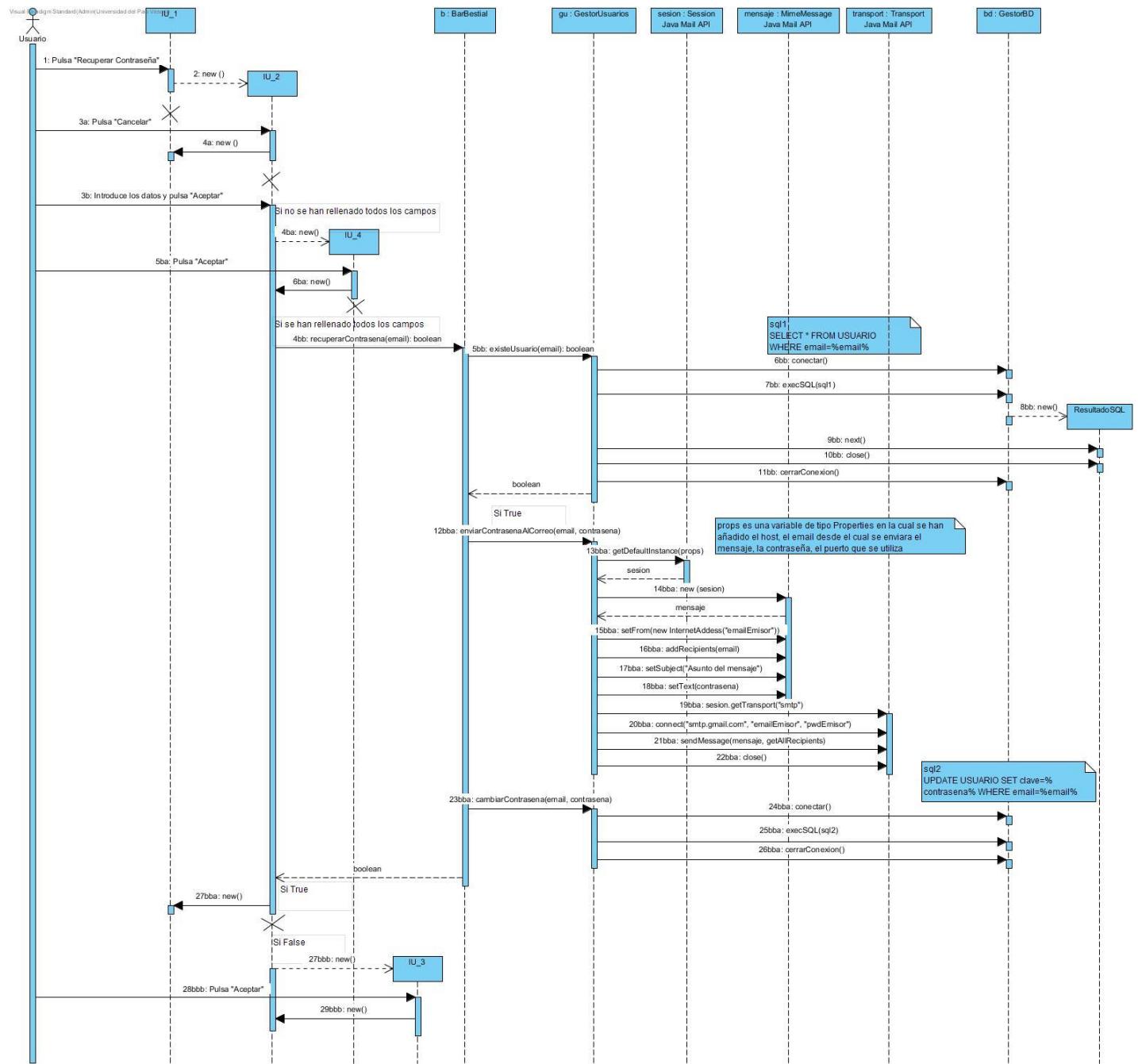
Link a la imagen: https://drive.google.com/open?id=1pMP9CKuApszpa4G_yN3fdUbPqEnLA1kG

6.1.3 Identificación RRSS



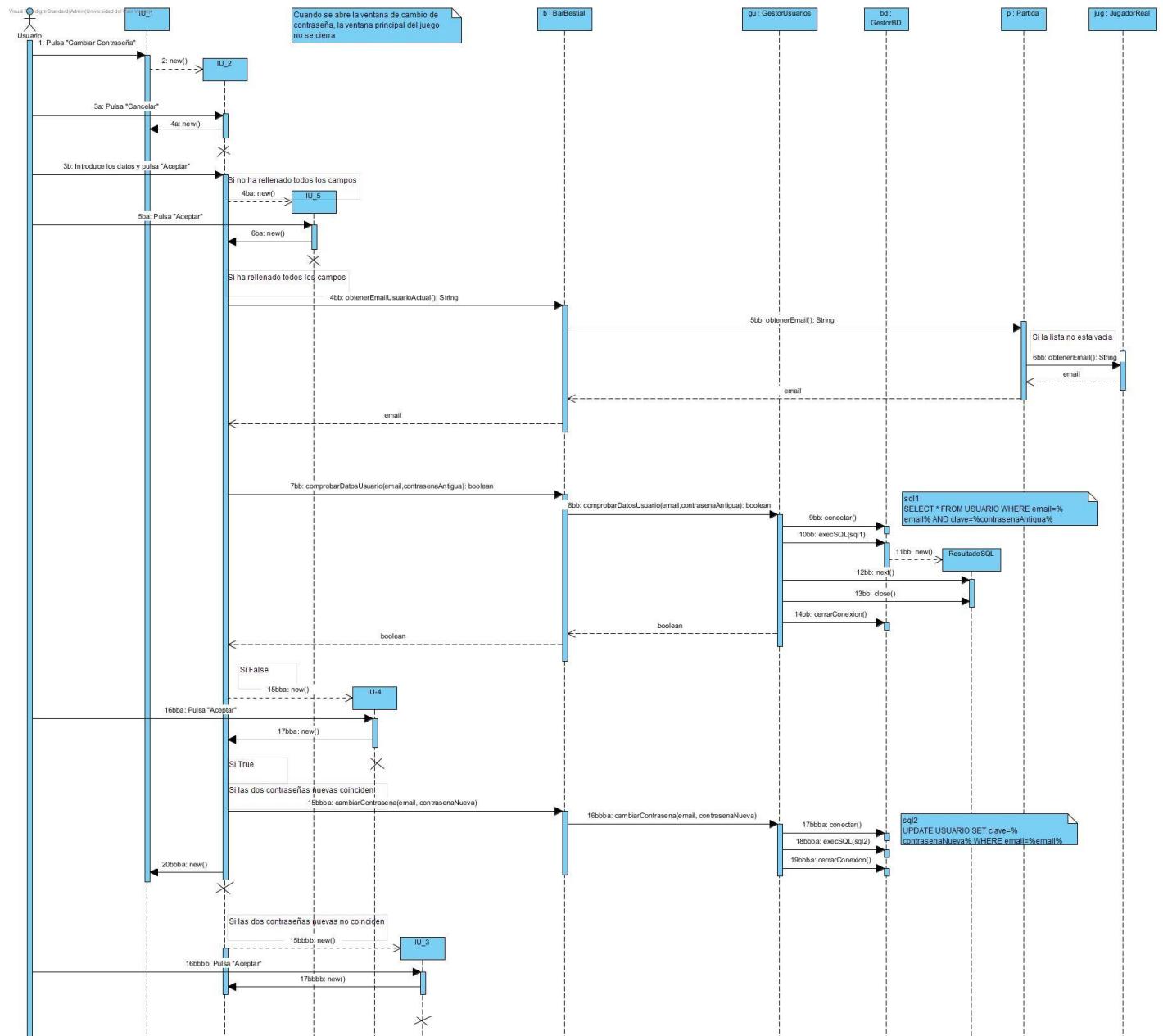
Link a la imagen: <https://drive.google.com/open?id=13QoqFKIQUTrbOkGWBf2jxl8fPL48PFMA>

6.1.4 Recuperación de contraseña



Link a la imagen: https://drive.google.com/open?id=14qtz_oX-sQBX2oEYaXZYEdnfMwrmeDkN

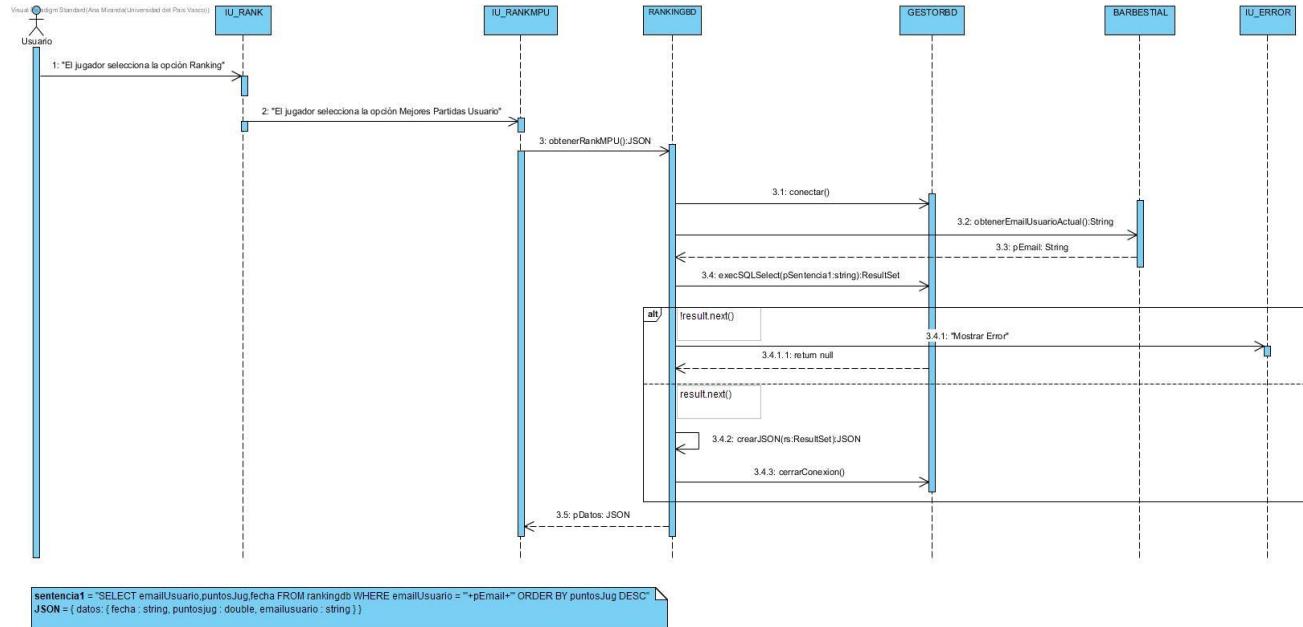
6.1.5 Cambio de contraseña



Link a la imagen: <https://drive.google.com/open?id=1GR7gh1wPuMIGH7XAvh61W321LY6kKpM2>

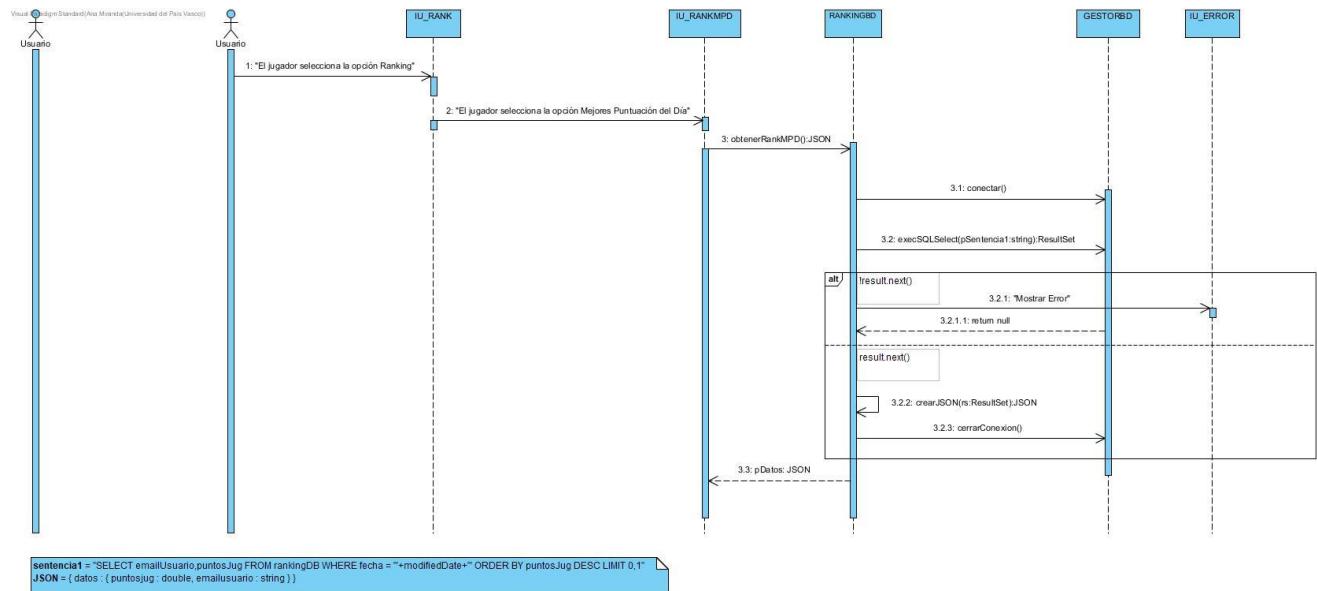
6.2 Visualización del ranking

6.2.1 Ranking mejores partidas usuario



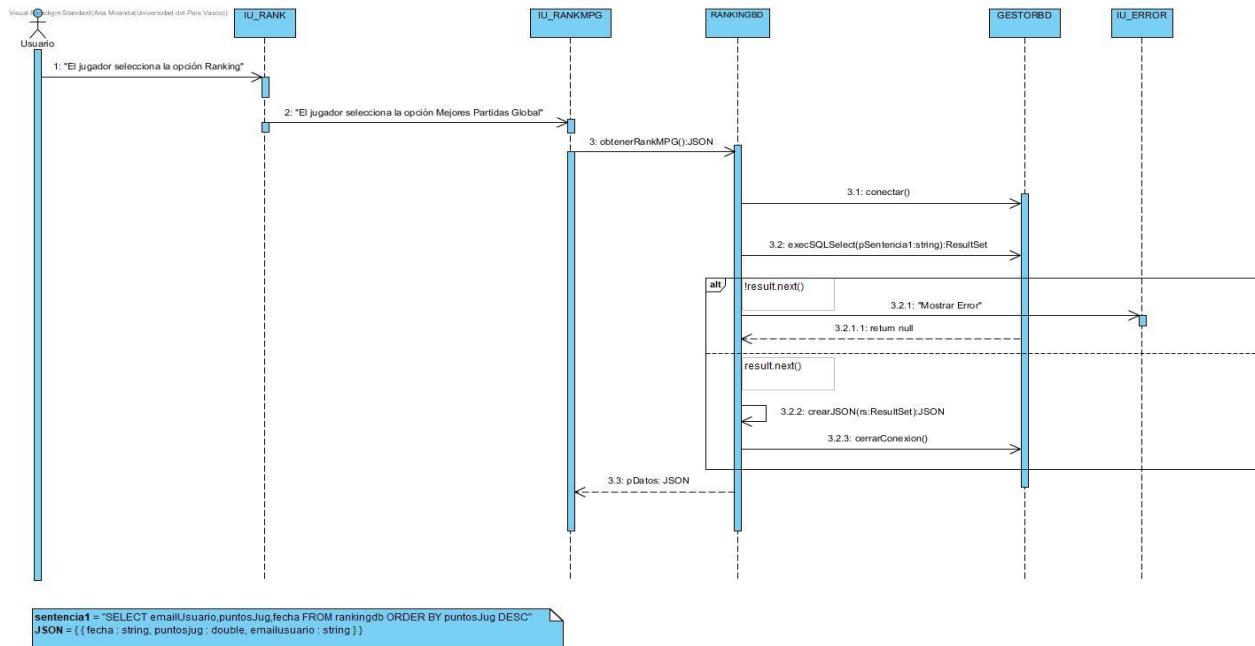
Link a la imagen: <https://drive.google.com/open?id=19LHj2vfFCZtISf2ebRGfr8TKRdXpjHYe>

6.2.2 Ranking mejor puntuación diaria usuario



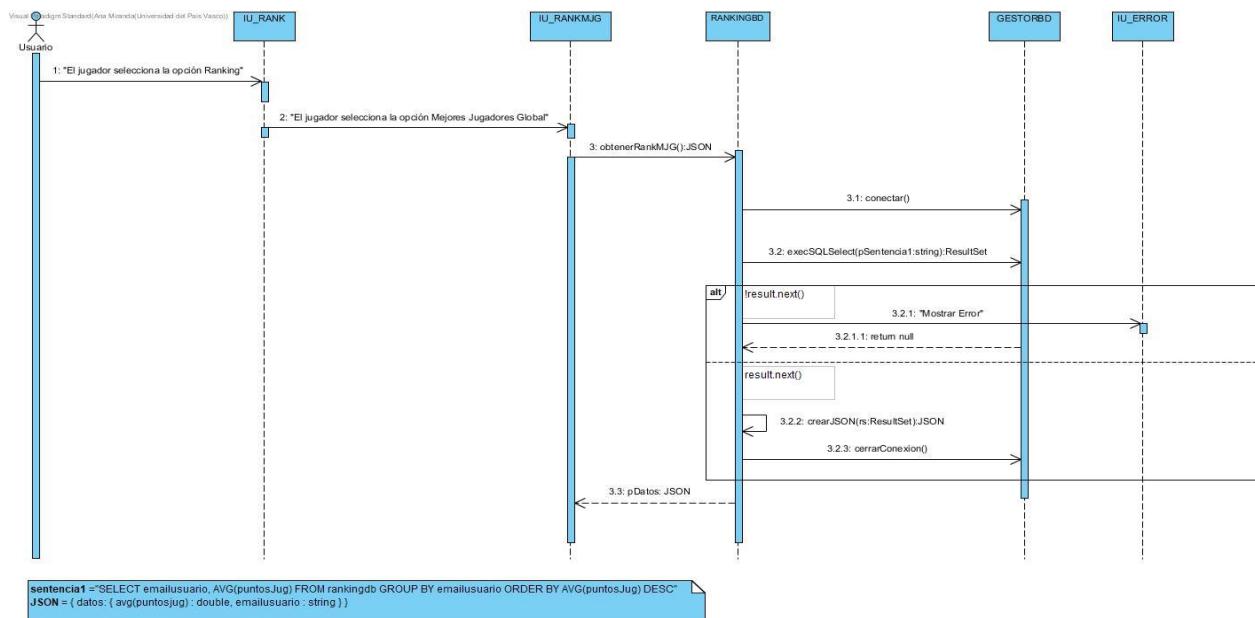
Link a la imagen: https://drive.google.com/open?id=1uD_6IxNbpf2WHv2qalS-1OJL8VsE8bXq

6.2.3 Ranking mejores partidas



Link a la imagen: <https://drive.google.com/open?id=1R-OhtnnRiXzc82WhbSbXIIHZnfCWBVRM>

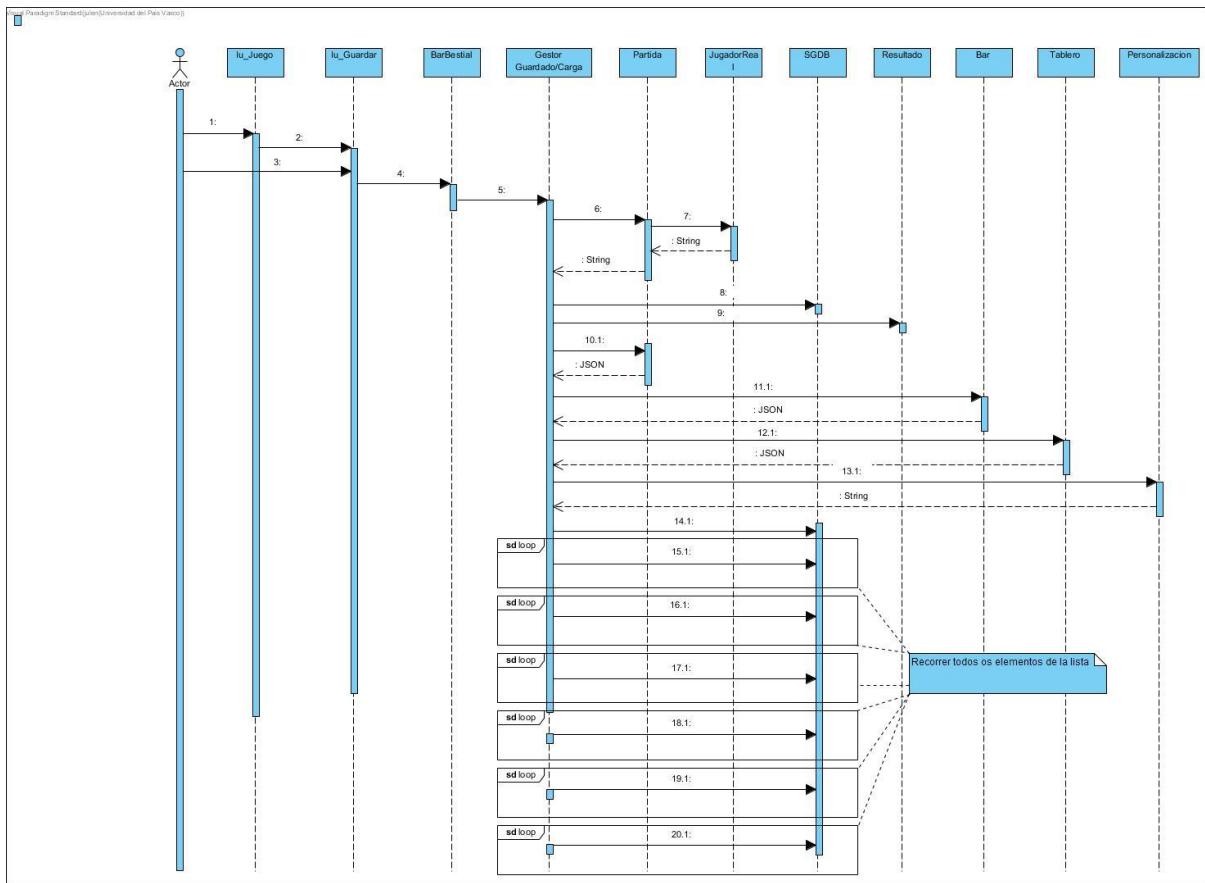
6.2.4 Ranking mejores jugadores



Link a la imagen: <https://drive.google.com/open?id=1Xb7c0VI3GNfyj9LX7H9MgZ3osAmuILqk>

6.3 Guardar/Reanudar partida

6.3.1 Guardar partida



1: El usuario pulsa en el botón guardar

2: new_IuGuardar()

3: El usuario introduce el nombre con el que guardar la partida y pulsa “aceptar”.

4: guardar(nombrePartida:string)

5: guardar(nombrePartida:string)

6: obtenerEmail():string

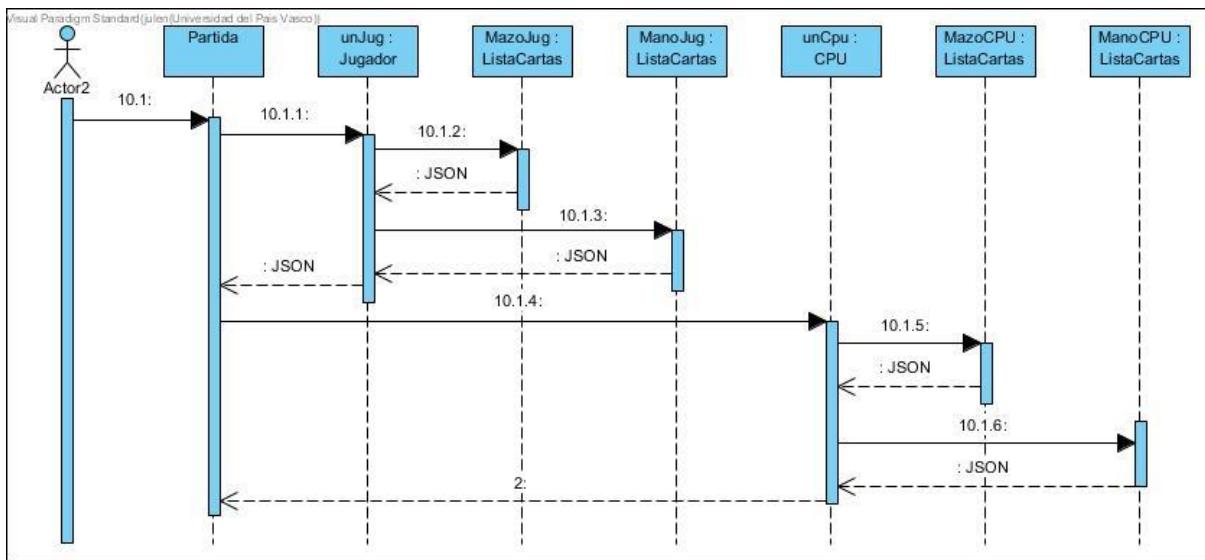
7: obtenerEmail():string

8: execSql("SELECT nombrePartida FROM Partida WHERE nombrePartida=%nombrePartida% AND email=%emailUsuario%")

9: new()

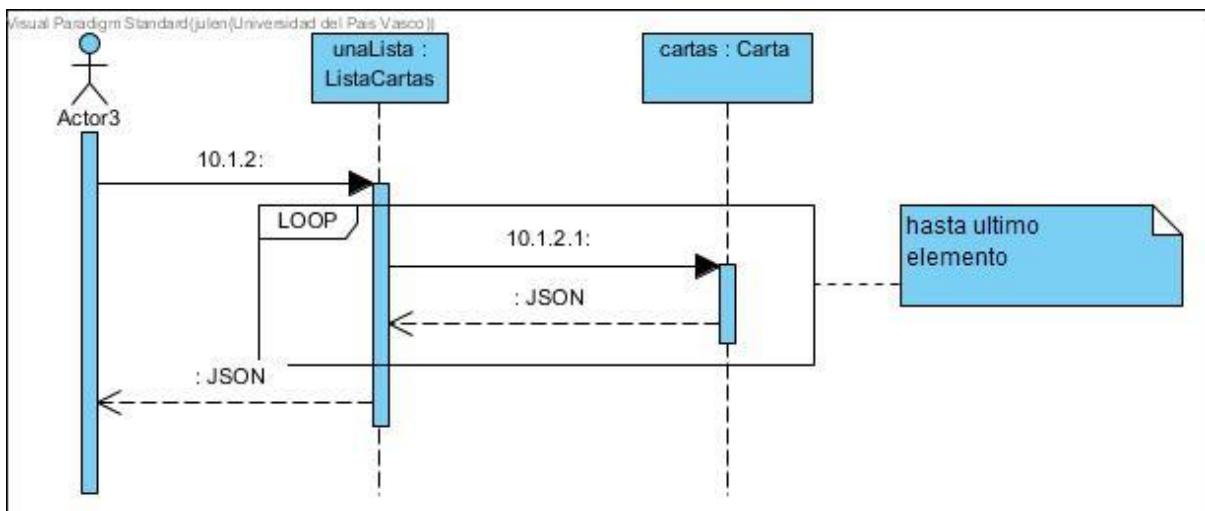
[nombre no en uso]

10.1: obtenerDatosPartida():JSON /*tipo4



10.1.1:obtenerDatosJugador():JSON /**tipo3

10.1.2:obtenerDatosLista():JSON /**tipo2



10.1.2.1:obtenerDatosCarta():JSON /**tipo1

10.1.3:obtenerDatosLista():JSON /**tipo2

10.1.3.1:obtenerDatosCarta():JSON /**tipo1

10.1.4:obtenerDatosJugador():JSON /**tipo3

10.1.5:obtenerDatosLista():JSON /**tipo2

10.1.5.1:obtenerDatosCarta():JSON /**tipo1

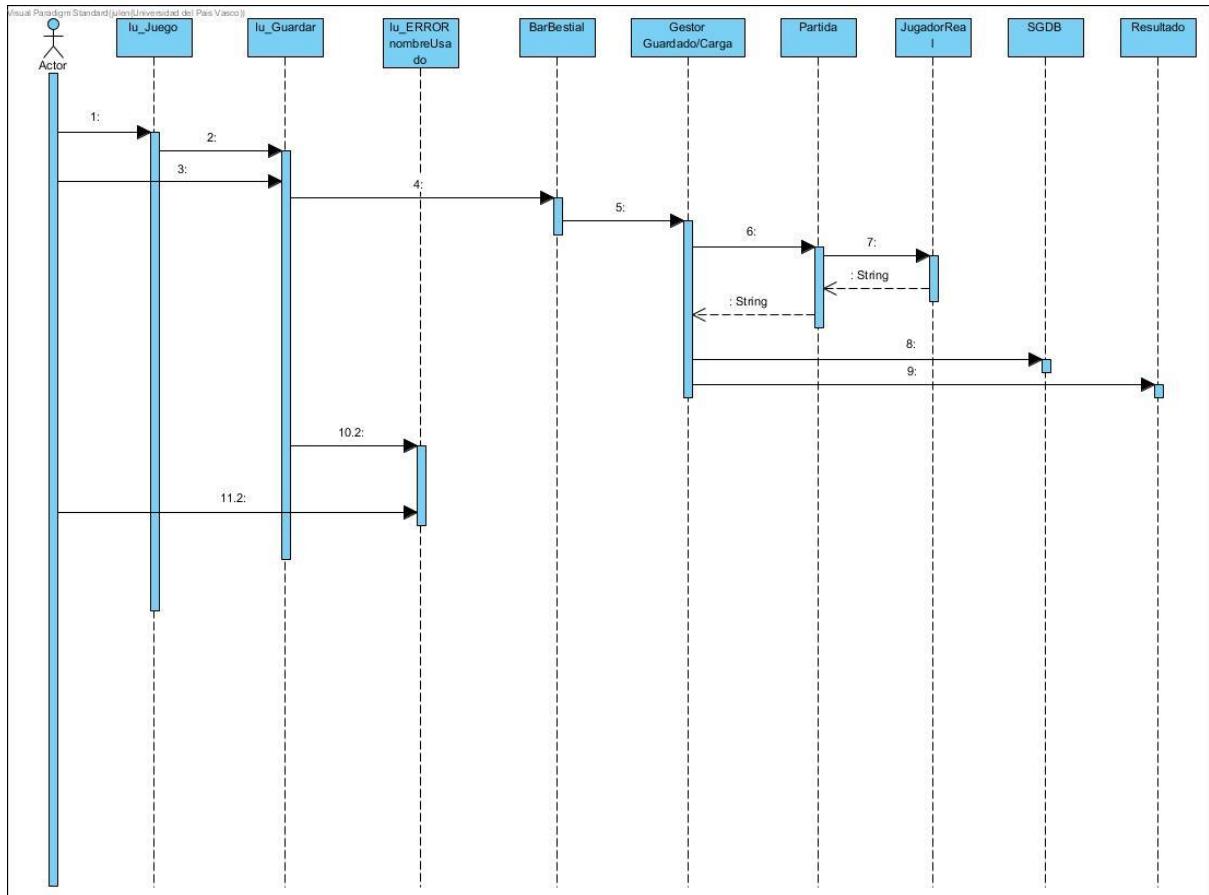
10.1.6:obtenerDatosLista():JSON /**tipo2

10.1.6.1:obtenerDatosCarta():JSON /**tipo1

11.1: obtenerDatosBar():JSON /**tipo2

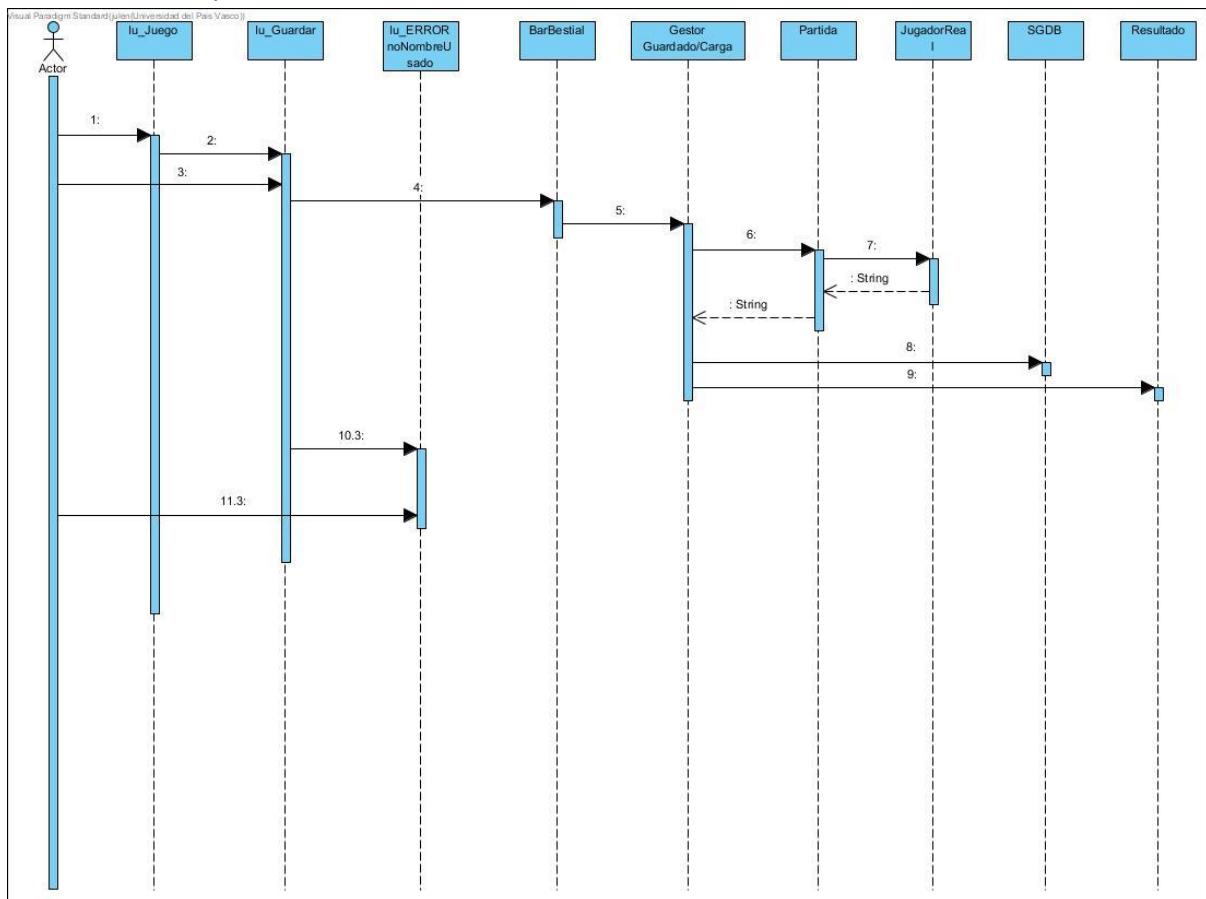
4.1.4.1:obtenerDatosLista():JSON /**tipo2

4.1.4.1.1:obtenerDatosCarta():JSON /**tipo1
 12.1: obtenerDatosTablero():JSON /**tipo2
 4.1.5.1:obtenerDatosLista():JSON /**tipo2
 4.1.5.1.1:obtenerDatosCarta():JSON /**tipo1
 13.1: getNombre():String
 14.1: execSql(" INSERT INTO Partida
 VALUES(%nombrePartida%,%fechaHora,%emailUsuario,%ayudasUsadas,%nombrePersonalizacion)"
)
 15.1: execSql(" INSERT INTO Fila VALUES(%nombrePartida%,%especie%,%color%,%posicion%)")
 16.1: execSql(" INSERT INTO MazoJug
 VALUES(%nombrePartida%,%especie%,%color%,%posicion%)")
 17.1: execSql(" INSERT INTO MazoOrd
 VALUES(%nombrePartida%,%especie%,%color%,%posicion%)")
 18.1: execSql(" INSERT INTO ManoJug VALUES(%nombrePartida%,%especie%,%color%)")
 19.1: execSql(" INSERT INTO ManoOrd VALUES(%nombrePartida%,%especie%,%color%)")
 20.1: execSql(" INSERT INTO Bar VALUES(%nombrePartida%,%especie%,%color%)")
 [nombre en uso]



10.2: new_Iu_ERRORnombreUsado()

11.2: el usuario pulsa “aceptar” y vuelve a la pantalla de guardar
[nombre no puesto]



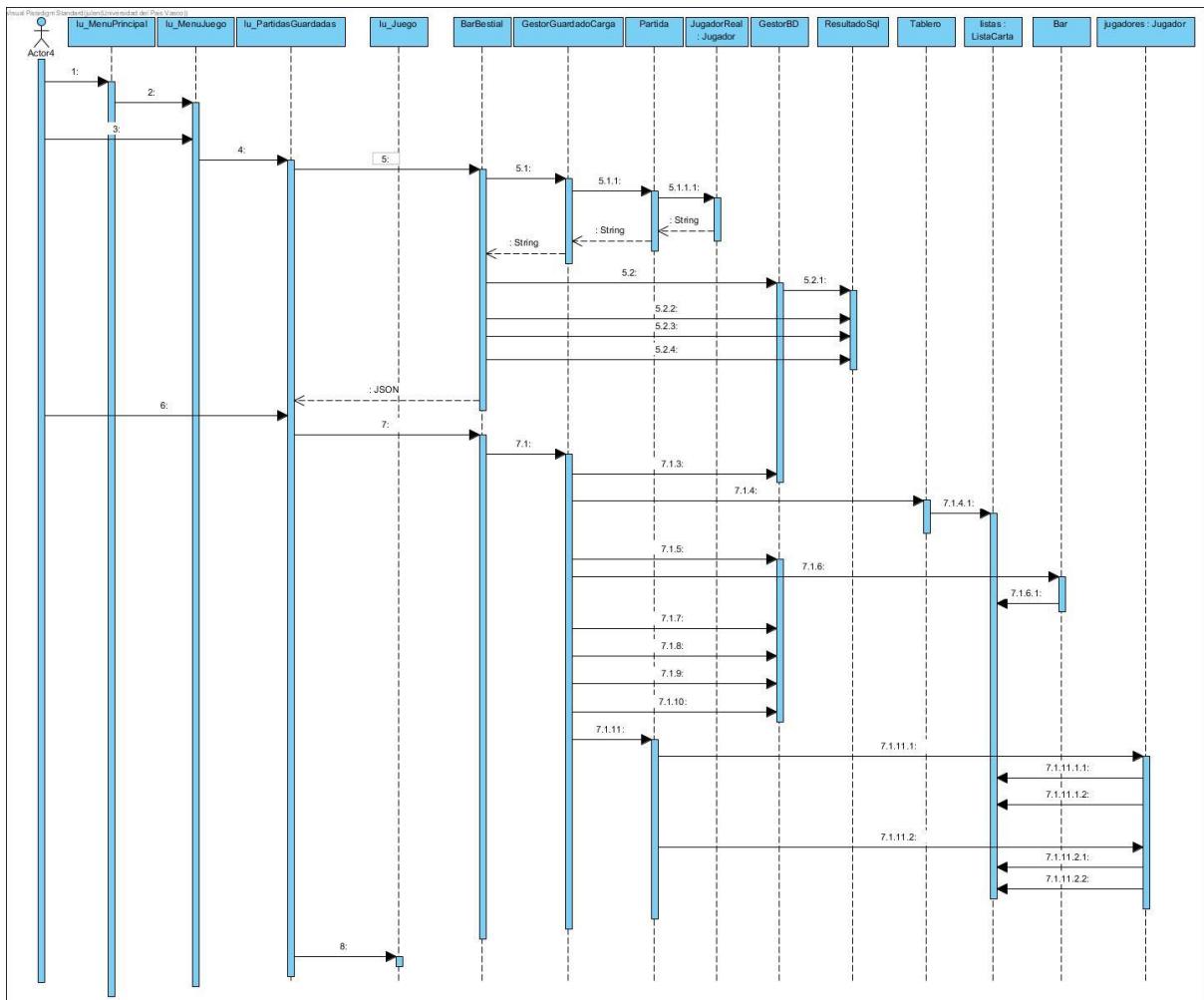
10.3: new_Iu_ERRORnoNombreUsado()

11.3: el usuario pulsa “aceptar” y vuelve a la pantalla de guardar

```

JSON tipo1 {especie: string, color: EnumColor}
JSON tipo2 [{tipo1:JSON,posición: int},{JSON tipo1,posición: int}....]
JSON tipo3 {tipo2:JSON,tipo2:JSON}
JSON tipo4 {tipo3: JSON , tipo3:JSON, ayudasUsadas:int}
  
```

6.3.2 Reanudar partida



1: El usuario pulsa “Jugar”.

2: newIU_MenuJugar.

3: El usuario pulsa “Reanudar”.

4: newIU_PartidasGuardadas()

5: obtenerPartidasGuardadas():JSON /** tipo1

5.1: obtenerEmail():string

5.1.1: obtenerEmail():string

5.1.1.1: obtenerEmail():string

5.2 execSQL(“SELECT nombrePartida,nombrePersonalizacion FROM Partida WHERE emailUsuario=%email%”)

5.2.1: new()

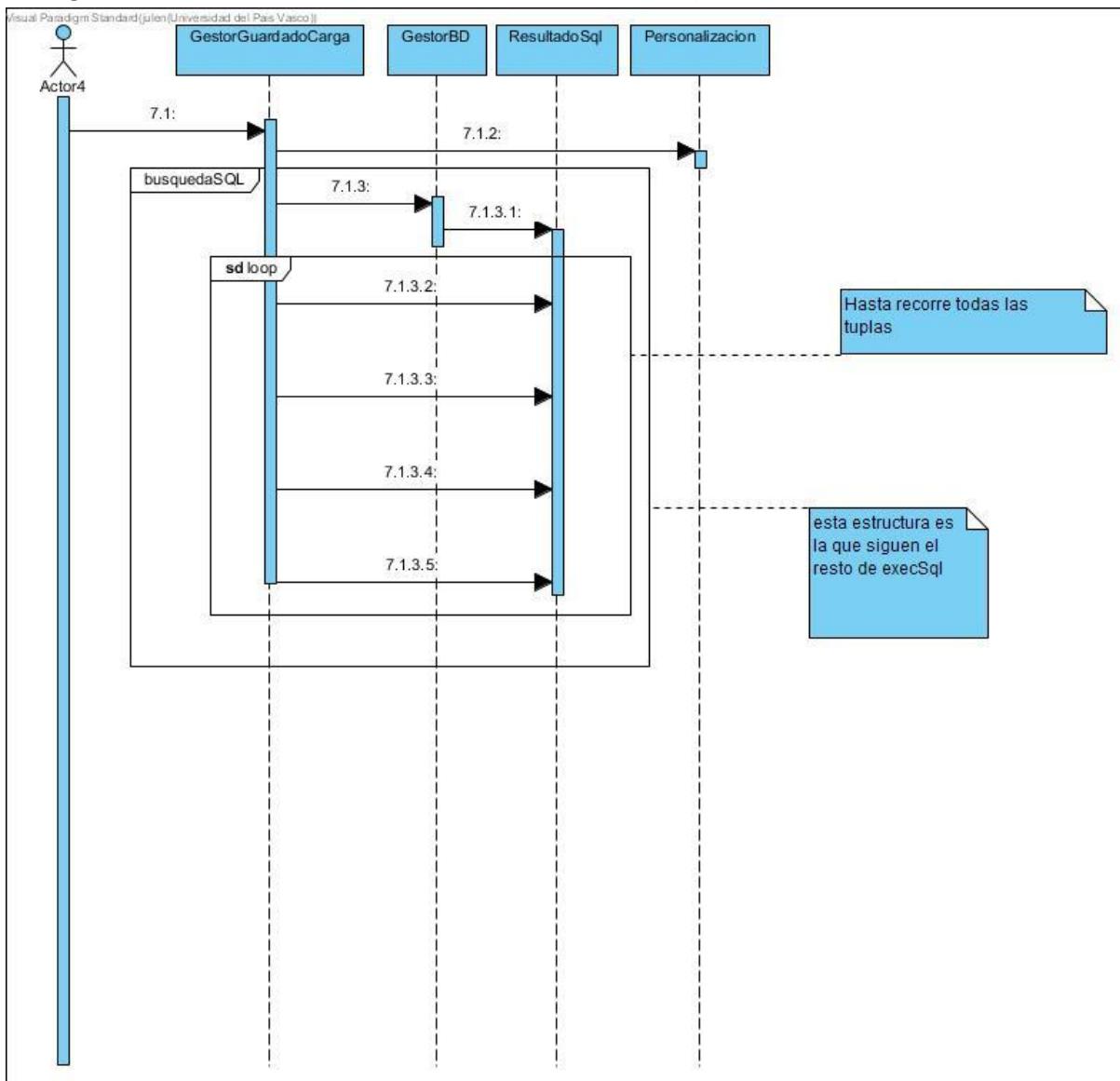
5.2.2: next()

5.2.3: getDate(“nombrePartida”)

5.2.4: getString(“nombrePersonalizacion”)

6: El usuario selecciona una partida.

7: cargarPartida(email,nombrePersonalizacion,nombrePartida)



7.1:cargarPartida(email,nombrePersonalizacion,nombrePartida)

7.1.2:cargarPersonalizacion(nombrePersonalizacion)

7.1.3: execSQL("SELECT especie,color,posicion FROM Fila WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")

7.1.3.1: new()

7.1.3.2: next()

7.1.3.3: getString("especie")

7.1.3.4: getEnum("color")

7.1.3.5: getInt("posicion")

7.1.4:cargarTablero(JSON) /** JSON tipo3

7.1.4.1: cargarListaPosicion(JSON) /** JSON tipo3

7.1.5: execSQL("SELECT especie,color FROM Bar WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")

- 7.1.5.1: new()
- 7.1.5.2: next()
- 7.1.5.3: getString("especie")
- 7.1.5.4: getEnum("color")

7.1.6: cargarBar(JSON) /** JSON tipo 4

- 7.1.6.1: cargarLista(JSON) /** JSON tipo3

7.1.7: execSQL("SELECT especie,color,posicion FROM MazoJug WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")

- 7.1.7.1: new()
- 7.1.7.2: next()
- 7.1.7.3: getString("especie")
- 7.1.7.4: getEnum("color")
- 7.1.7.5: getInt("posicion")

7.1.8: execSQL("SELECT especie,color,posicion FROM MazoOrd WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")

- 7.1.8.1: new()
- 7.1.8.2: next()
- 7.1.8.3: getString("numero")
- 7.1.8.4: getEnum("color")
- 7.1.8.5: getInt("posicion")

7.1.9: execSQL("SELECT especie,color FROM ManoJug WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")

- 7.1.9.1: new()
- 7.1.9.2: next()
- 7.1.9.3: getString("numero")
- 7.1.9.4: getEnum("color")

7.1.10: execSQL("SELECT especie,color FROM ManoOrd WHERE emailUsuario=%email% AND fechaHora=%fechaHora%")

- 7.1.10.1: new()
- 7.1.10.2: next()
- 7.1.10.3: getString("numero")
- 7.1.10.4: getEnum("color")

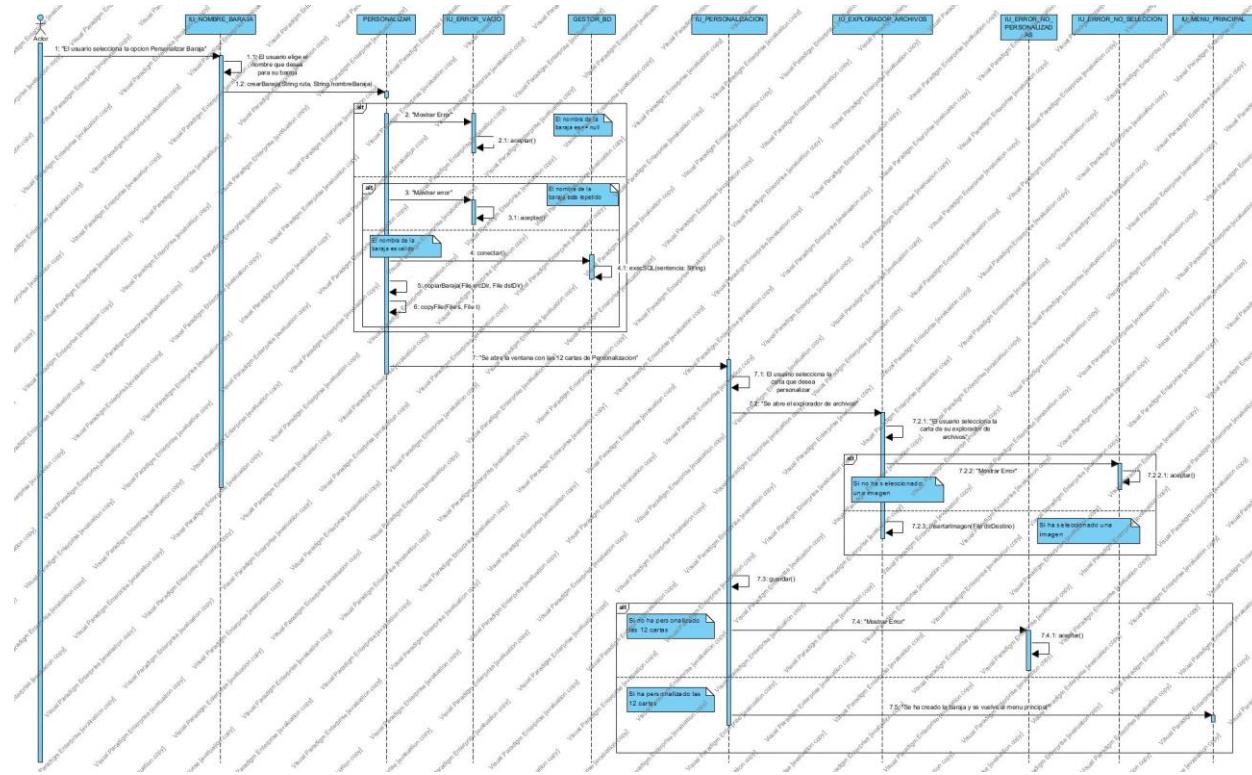
7.1.11: cargarJugador(JSON) /** JSON tipo5

- 7.1.11.1.: cargarJugador(JSON) /** JSON tipo6
- 7.1.11.1.1: cargarLista(JSON) /** JSON tipo4
- 7.1.11.1.2: cargarListaPosicion(JSON) /** JSON tipo3

7.1.11.2: cargarJugador(JSON) /** JSON tipo6
7.1.11.2.1: cargarLista(JSON) /** JSON tipo4
7.1.11.2.1: cargarListaPosicion(JSON) /** JSON tipo3
8: newIU_Juego()

JSON tipo1 [{nombrePartida: String,nombrePersonalizacion: String},{...}...]
JSON tipo2 [{numero:int,color: EnumColor, imagen: Image},{...}...]
JSON tipo3 [{numero: int, color: EnumColor, posicion: int},{...}...]
JSON tipo4 [{numero: int, color: EnumColor},{...}...]
JSON tipo5 [{tipo3: JSON,tipo4: JSON},{tipo3: JSON,tipo4: JSON}]
JSON tipo6 [{tipo3: JSON,tipo4: JSON}]

6.4 Personalización



[Enlace al diagrama en PNG](#)

[Enlace al diagrama en VPP](#)

Flujo de eventos

1. El usuario selecciona la opción Personalizar Baraja
 - 1.1. El usuario elige el nombre que desea para su baraja
 - 1.2. crearBaraja(String ruta, String nombreBaraja)

[SI EL NOMBRE DE LA BARAJA ES VACÍO]
2. Mostrar error
 - 2.1. aceptar()

[SI EL NOMBRE DE LA BARAJA ESTÁ REPETIDO]
3. Mostrar error
 - 3.1. aceptar()

[SI EL NOMBRE DE LA BARAJA ES VALIDO]
4. conectar()
 - 4.1. execSQL(sentencia: String)
5. copiarBaraja(File srcDir, File dstDir)
6. copyFile(File s, File t)

7. Se abre la ventana con las 12 cartas de Personalización

 7.1. El usuario selecciona la carta que desea personalizar

 7.2. Se abre el explorador de archivos

 7.2.1. El usuario selecciona la carta de su explorador de archivos

 [SI NO HA SELECCIONADO UNA IMAGEN]

 7.2.2. Mostrar Error

 7.2.2.1. aceptar()

 [SI HA SELECCIONADO UNA IMAGEN]

 7.2.3. insertarImagen(File dirDestino)

 7.3. guardar()

 [SI NO HA PERSONALIZADO LAS 12 CARTAS]

 7.4. Mostrar Error

 7.4.1. aceptar()

 [SI HA PERSONALIZADO LAS 12 CARTAS]

 7.5. Se ha creado la baraja y se vuelve al menú principal

Sentencia: "INSERT INTO Personalizar(nombreBaraja, path) VALUES (+nombreBaraja+, +path+)"

7. Problemas en la implementación

Surgieron varios problemas a la hora de integrar el proyecto entero ya que, por razones desconocidas, a algunas personas no les funcionaba correctamente pero a otras sí. Finalmente llegamos a la conclusión de que sqlite nos daba algunos problemas por lo que decidimos hacer la base de datos en MySQL.

Al comienzo de la integración surgieron también problemas con la ortografía debido a los caracteres acentuados y a la letra “ñ” pero se solucionaron con facilidad cambiando la “ñ” por la “n”.

Además, al utilizar GitHub, también tuvimos problemas con los push y pull entre los usuarios, dándose el caso de haber cambiado el classpath y no funcionar absolutamente nada del proyecto.

Llegando a la finalización del proyecto, por falta de tiempo y ser época de exámenes, ha sido prácticamente imposible realizar la automatización de las pruebas de algunas funcionalidades, a pesar de, en la mayoría de los casos, haber comprobado el funcionamiento correctamente antes de realizar la entrega final. Los planes de prueba de dichas funcionalidades tampoco han sido completadas.

8. Conclusiones

Una vez finalizada la práctica podemos concluir que hemos trabajado correctamente en grupo a pesar de tener algunos contratiempos (como la baja de uno de los componentes del grupo en el tramo final del proyecto).

En líneas generales, no hemos tenido problemas con las fechas de entrega ya que prácticamente en todas las ocasiones el trabajo estaba finalizado con anterioridad. El único momento de estrés ha surgido en la parte final debido a los problemas en la integración de todas las funcionalidades. A pesar de eso, finalmente hemos podido lograr nuestros objetivos en la mayoría de las funcionalidades a implementar.

Gracias a esta práctica, hemos aprendido a trabajar individualmente para, posteriormente, realizar una integración con el resto de compañeros y de esta forma, trabajar en grupo. Además, hemos ido aplicando los diferentes conocimientos adquiridos en la asignatura lo cual ha sido útil a la hora de estudiar ya que nos ha servido como reto.