3-4-2020

**PROYECTOS INFORMÁTICOS II**

**MEMORIA DEL PROYECTO**  
 **MONOPOLY**

GRUPO A

DANIEL – AYOUB – RAÚL

**ÍNDICE**

[1. Control de documentación 2](#_Toc36810182)

[2. Historial de modificaciones 2](#_Toc36810183)

[3. Casos de uso 3](#_Toc36810184)

[4. Modelo Objeto-Relación 4](#_Toc36810185)

[5. Análisis de requisitos 5](#_Toc36810186)

[6. Planificación en fase de ejecución e hitos 0](#_Toc36810187)

[7. Diagramas de actividad 1](#_Toc36810188)

[8. Diagramas de secuencia 8](#_Toc36810189)

[9. Diagrama de componentes 15](#_Toc36810190)

# Control de documentación

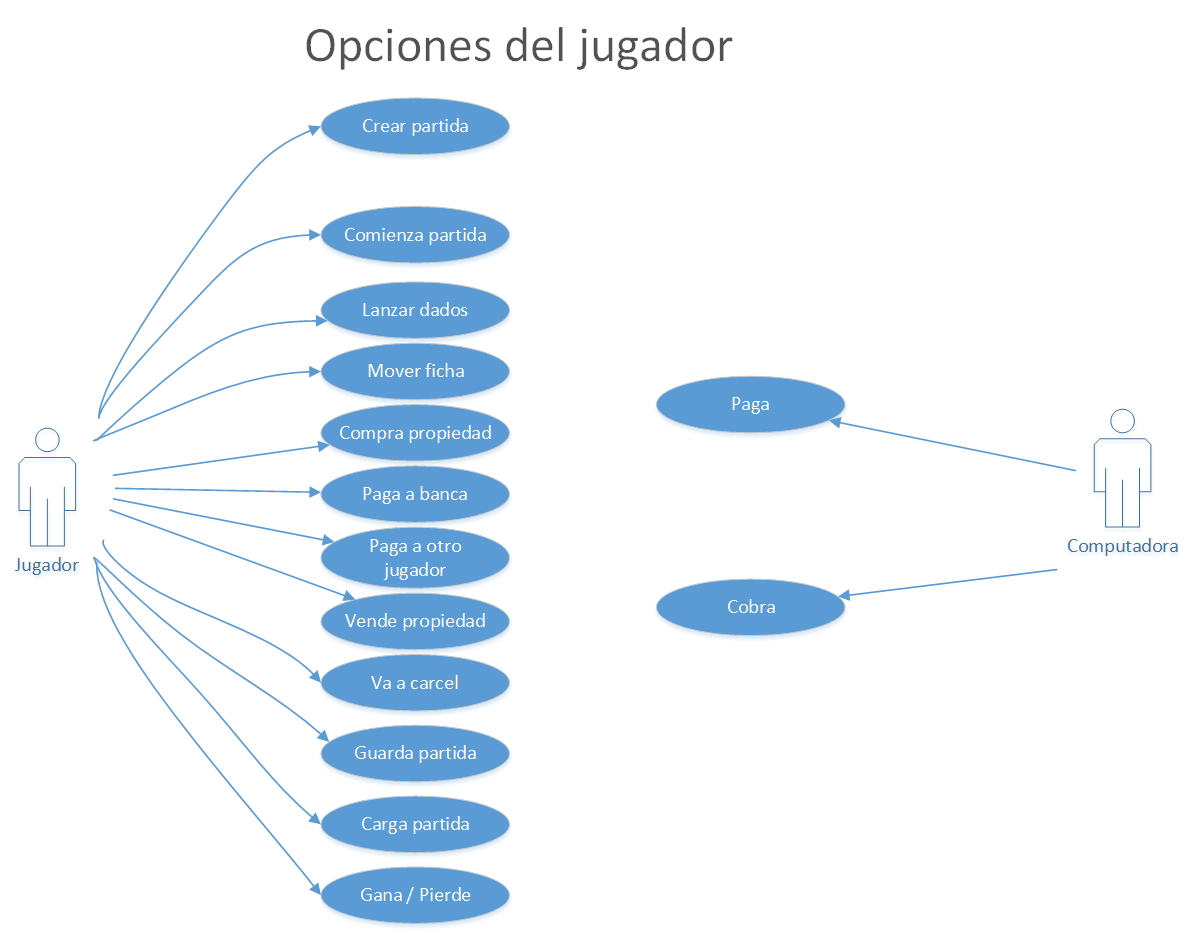
|  |  |  |
| --- | --- | --- |
| **Elaborado por:** | **Revisado por:** | **Aprobado por:** |
| Alumnos de la asignatura (desarrollo) | Alumnos de la asignatura (revisado) | Equipo de la Universidad |
| **Nombre:** | **Nombre:** | **Nombre:** |
| Daniel Ortega  Ayoub Dachour  Raúl García | Daniel Ortega  Ayoub Dachour  Raúl García | [Jordi José](https://opencampus.uols.org/user/view.php?id=1347&course=872) |
| **Fecha:** | **Fecha:** | **Fecha:** |
| 03/04/2020 |  |  |
| **Firma:** | **Firma:** | **Firma:** |
|  |  |  |

# Historial de modificaciones

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSIÓN** | **FECHA DE VIGENCIA** | **MODIFICADO POR:** | **MODIFICACIÓN REALIZADA** |
| 1.0 | 03/04/2020 | Raúl García | Creación de documento |

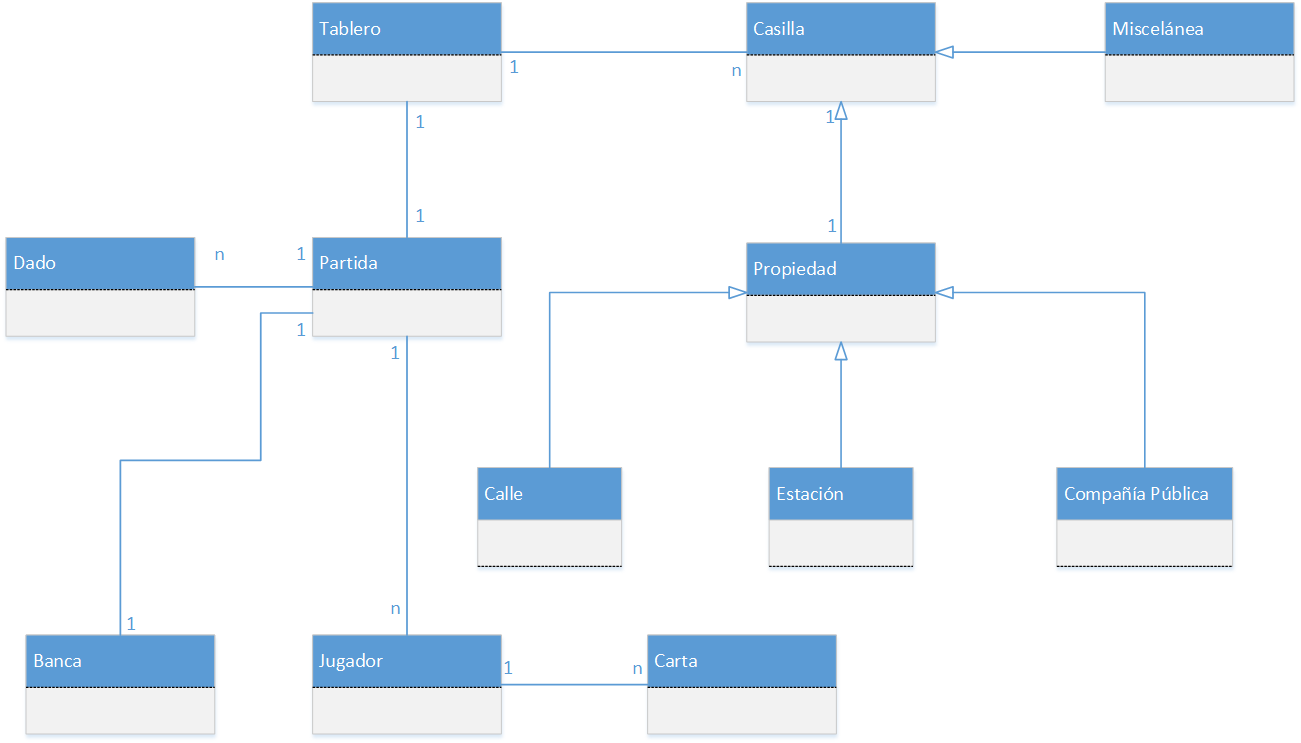
# Casos de uso

A continuación se expone casos de uso genéricos, con las principales funcionalidades que dispondrá la aplicación. En el futuro se estudiará realizar una evolución/modificación de estos e implementación de diagrama de clases.



# Modelo Objeto-Relación

En este apartado vamos a exponer el modelo que, aunque no implementaremos ninguna base de datos, nos ayudará a comprender la relación entre objetos de la aplicación. Este modelo es una aproximación a alto nivel que puede sufrir cambios a lo largo del desarrollo del proyecto.



# Diagrama de Clases

# Análisis de requisitos

A modo de recordatorio, se dejarán todos los requisitos analizados a tener en cuenta por el grupo de desarrollo en esta fase de ejecución, ya que se deberán aplicar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Tipo** | **Prioridad** | **Capa** | **Aplicado o Comentarios** |
| Programación MVC | No funcional | Alta | Todas |  |
| Programación siguiendo buenas prácticas (comentarios, estructura…) | No funcional | Alta | Todas |  |
| Soporte de 2 a 8 jugadores | Funcional | Alta | Controlador |  |
| Incluir Robots con una IA simplista | Funcional | Baja | Controlador |  |
| Deberá soportar importación de partidas mediante ficheros | Funcional | Alta | Modelo |  |
| Deberá soportar guardado de partidas mediante ficheros | Funcional | Alta | Modelo |  |
| Deberá soportar carga de partidas mediante ficheros | Funcional | Alta | Modelo |  |
| El jugador puede crear una partida nueva | Funcional | Alta | Controlador |  |
| Se deberá aportar el código fuente del programa | No funcional | Alta | Todas |  |
| Consola de texto para visualizar la información de las acciones | Funcional | Alta | Vista |  |
| El jugador debe poder mover la ficha | Funcional | Alta | Controlador |  |
| El jugador debe poder tirar el dado | Funcional | Alta | Controlador |  |
| El jugador debe poder comprar una propiedad | Funcional | Alta | Controlador |  |
| El jugador debe poder comprar una propiedad | Funcional | Alta | Controlador |  |
| El orden de salida será aleatorio | Funcional | Alta | Controlador |  |
| El jugador deberá ver la interacción del juego (Tablero y consola de texto) | Funcional | Alta | Vista |  |
| Incluir cartas de suerte y comunidad | Funcional | Alta | Modelo |  |
| Jugador podrá visualizar cartas suerte y comunidad | Funcional | Alta | Vista |  |
| Sólo se podrá adquirir una propiedad por turno ya sea mediante la compra o la negociación con otro jugador. Las casillas no compradas no saldrán a subasta. | Funcional | Media | Controlador |  |
| Se usan dos dados para el juego | Funcional | Alta | Controlador |  |
| Un jugador pierde cuando entra en bancarrota y tiene un saldo negativo. | Funcional | Alta | Controlador |  |
| Los jugadores recibirán 200€ cada vez que pasen por la casilla de salida mientras queden propiedades sin dueño. | Funcional | Alta | Controlador |  |
| Si se sacan dobles se vuelve a tirar, si se sacan dobles 3 veces seguidas el jugador irá a la cárcel. | Funcional | Alta | Controlador |  |
| Cuando un jugador pierde, sus propiedades pasan al jugador con quien ha contraído la deuda o a la banca en caso que la deuda sea con ella (en este caso las casillas pasarán a estar en venta y las edificaciones se destruirán). | Funcional | Media | Controlador |  |
| Las multas las gestiona la banca creando un fondo de dinero que será obtenido por el primer jugador que caiga en parking gratuito. | Funcional | Media | Controlador |  |
| Cuando un jugador va a la cárcel se mueve a la casilla (11). Deberá estar tres turnos y puede tirar dados y salir con dobles o pagar 50€ de multa. Si está tres turnos en la cárcel deberá pagar 50€ de multa y tirar dados. (No se volverá a tirar aunque haya sacado dobles). | Funcional | Media | Controlador |  |
| El número total de casas es de 32 y el de hoteles 12. | Funcional | Media | Controlador |  |
| Cada jugador empieza con 1500€ | Funcional | Media | Modelo |  |

De igual forma, se irá detallando si se ha realizado cada caso de uso en la columna “Aplicado o comentarios”

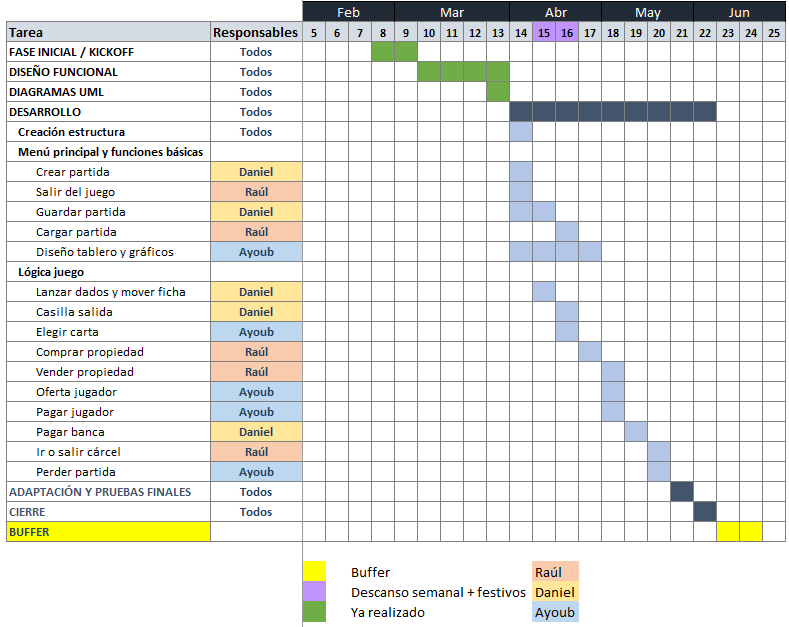
La siguiente tabla pretende plasmar los casos de uso a desarrollar entrando más en detalle en las características de cada uno de ellos.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Nombre | CU Rel. | Descripción | Trigger | Requisitos Previos | Consecuencias | Excepciones | Aplicado o comentarios |
| 1 | Crear partida |  | Creará la partida con la configuración inicial seleccionada | El usuario selecciona la opción crear partida | La aplicación se ha de abrir y ejecutar correctamente | Se genera una nueva partida |  |  |
| 2 | Empezar partida | 1, 14 | Empezará la partida tras crearla o cargarla, en este último caso será desde la última acción realizada.  Si la partida es nueva, el usuario empieza con 1500 euros. | El usuario selecciona la opción empezar partida | La partida ha debido ser creada o cargada | Carga la nueva partida o una ya guardada previamente |  |  |
| 3 | Lanzar Dados | 2 | El usuario podrá lanzar los dados, que serán dos y obtendrá un resultado y en función de este podrá avanzar o se desbloquearán varias acciones. | El usuario hace clic en la opción lanzar dados | La partida tiene que estar empezada y todo lo que ello requiere | El usuario avanzará el número de casillas que resulte de tirar los dados. En caso de estar en la cárcel, dobles equivalen a salir de ella. |  |  |
| 4 | Mover ficha | 3 | En función del número obtenido en lanzar dados, se deberá mover X casillas la ficha | Al lanzar los dados, se llamará a esta función | El usuario ha debido de tirar los dados | Se moverán X casillas | Está en cárcel no moverá ficha a no ser que haya sacado dobles |  |
| 5 | Comprar propiedad | 3,4 | El usuario puede comprar una casa de un color si cae en la casilla correspondiente. El sistema indicará el precio de la propiedad y el usuario elegirá si comprar o no.  El usuario también podrá comprar una propiedad a otro usuario. Por lo que podría hacer una oferta. (relacionado con Caso de uso 8) | El usuario elegirá si comprar propiedad o no. | Se deberá verificar que el usuario tenga saldo para pagar. La partida ha tenido que ser empezada y los dados lanzados. |  | El usuario ya tiene la casa. Por lo que hay que pasar al caso de uso de pagar.  El usuario no tiene saldo |  |
| 6 | Elegir carta | 3,4 | El jugador podrá elegir dos cartas, de la suerte y de la comunidad. Sacan Estas cartas cuando se posan sobre las casillas correspondientes y se debe seguir las instrucciones que haya en ellas. | La ficha del jugador cae en una casilla que requiere coger carta | Que haya empezado la partida y el jugador pueda tirar los dados. Significa que no esté en bancarrota ni en la carcel | Se debe tomar una carta y se deberá restar una a la baraja |  |  |
| 7 | Pagar Banca | 3,4 | En el caso de caer en una casilla que requiera pagar multa, el jugador tendrá que pagar a la banca un importe X | Caer en casilla que requiera pagar | Que haya empezado la partida y el jugador pueda tirar los dados. Significa que no esté en bancarrota ni en la carcel | Se le restara un importe X al usuario |  |  |
| 8 | Oferta a jugador |  | El jugador puede hacer una oferta para comprar una propiedad a otro jugador. También existe la opción de intercambiar casas, por lo que esta oferta puede ser un cambio por otra propiedad o comprar por un importe a introducir | Usar la función de hacer oferta: El jugador deberá introducir la cantidad o la propiedad a intercambiar | Si la oferta es de dinero, verificará el importe del jugador. Si la oferta es de una casa, se deberá tener como mínimo una. Además de que el jugador no puede estar en bancarrota ni en la carcel | Si el jugador B acepta, quitarle la propiedad y/o el importe al jugador A |  |  |
| 9 | Vender propiedad |  | El jugador puede ofrecer la venta de su propiedad a otro jugador.  Se deberá introducir el importe por el que se desea vender. | Usar la función de vender propiedad: El jugador deberá seleccionar la propiedad y por cuanto dinero la desea vender | Verificar que la propiedad realmente exista y también que el jugador no esté en la cárcel o bancarrota | Quitarle la propiedad una vez la venta se haya realizado al jugador A y dárselo al jugador B |  |  |
| 10 | Ir a carcel | 3 | El jugador irá a la cárcel si cae en dicha casilla. Permanecerá en ella hasta que tirar dados sea dobles o se abone la cantidad de 50 euros | Caer en la casilla cárcel |  |  |  |  |
| 11 | Salir carcel | 10,7 | Tirar dados sea dobles, entonces el usuario podrá salir de la cartel en el siguiente turno. También podrá salir si abona 50 euros de multa (pagar a la banca, caso 7) | Tirada de dados dobles o pagar a la banca | El usuario debe estar ya en la cárcel |  |  |  |
| 12 | Casilla sailda | 3,4 | Al pasar por la casilla de salida, se le ingresará un importe X al usuario | Cuando se tiren dados y la ficha pase por dicha casilla |  |  |  |  |
| 13 | Pagar a jugador | 3,4 | Si el usuario cae en una propiedad de otro jugador. Este le deberá de pagar lo que corresponda | Cuando la ficha caiga en la casilla con una propiedad de otro jugador | La casilla debe tener la propiedad de otro jugador. | El jugador podrá hacer una oferta cambiándole una casa o tendrá que pagar.  La consecuencia mayor es que el jugador entrará en bancarrota y perderá la partida. |  |  |
| 14 | Guardar partida | 1,2 | La partida tendrá que estar empezada y el usuario podrá en cualquier momento elegir cuando guardar la partida. Todos los valores se guardarán en ficheros de texto externos que podrán ser cargados posteriormente. | Usuario selecciona la opción guardar partida | La partida debe estar empezada |  |  |  |
| 15 | Cargar partida | 13 | El usuario podrá hacer uso de la función cargar partida que obtendrá los valores guardados previamente | El usuario selecciona la opción cargar partida | Debe existir al menos una partida guardada | La partida actual será reemplazada por la nueva |  |  |
| 16 | Salir del juego | N/A | El usuario saldrá de la aplicación | El usuario selecciona la opción Salir | Debe estar el juego iniciado | Saldrá del juego completamente |  |  |
| 16 | Jugador pierde la partida | 7, 13 | Cuando entra en bancarrota o saldo negativo | Al entrar en bancarota | Tener un saldo superior a 0 y que pase a ser inferior a 0 |  |  |  |

# Planificación en fase de ejecución e hitos

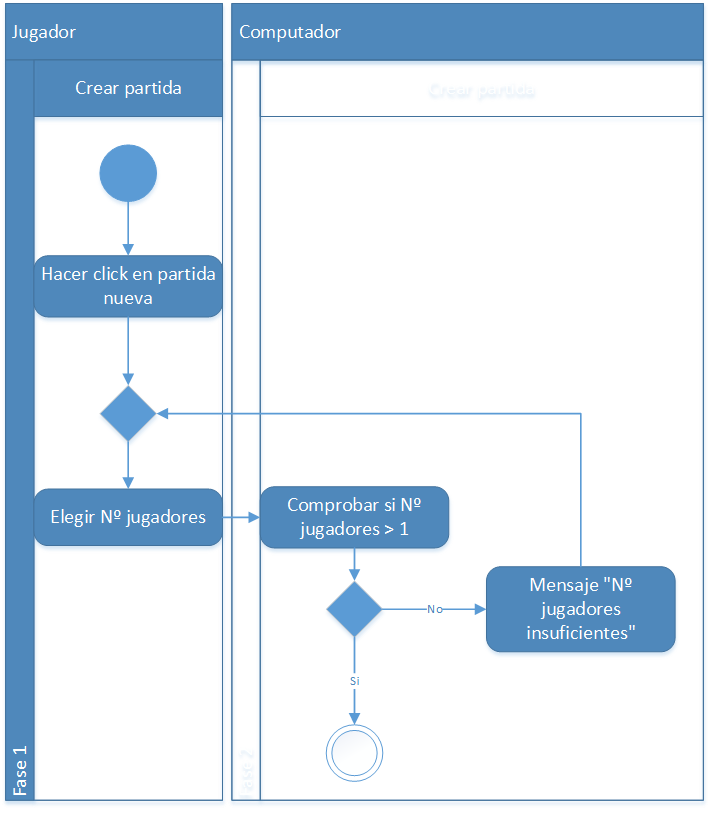
A continuación se detalla la planificación en detalle del proyecto que debe ser actualizada constantemente. Se divide por fases del desarrollo y casos de uso y las fases relacionadas.

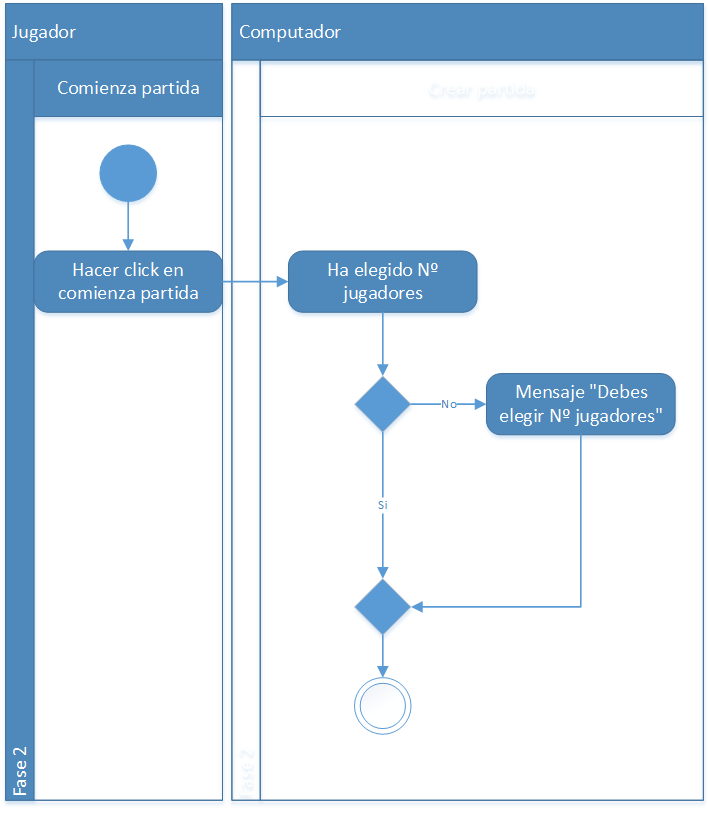
**Planificación y ejecución a fecha 28.03.2020**

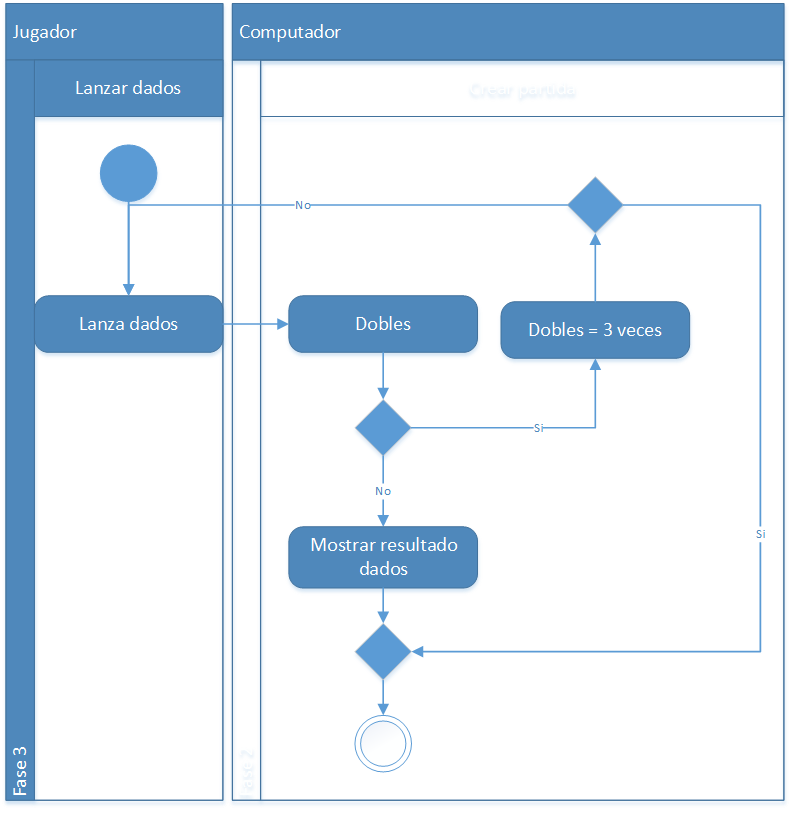


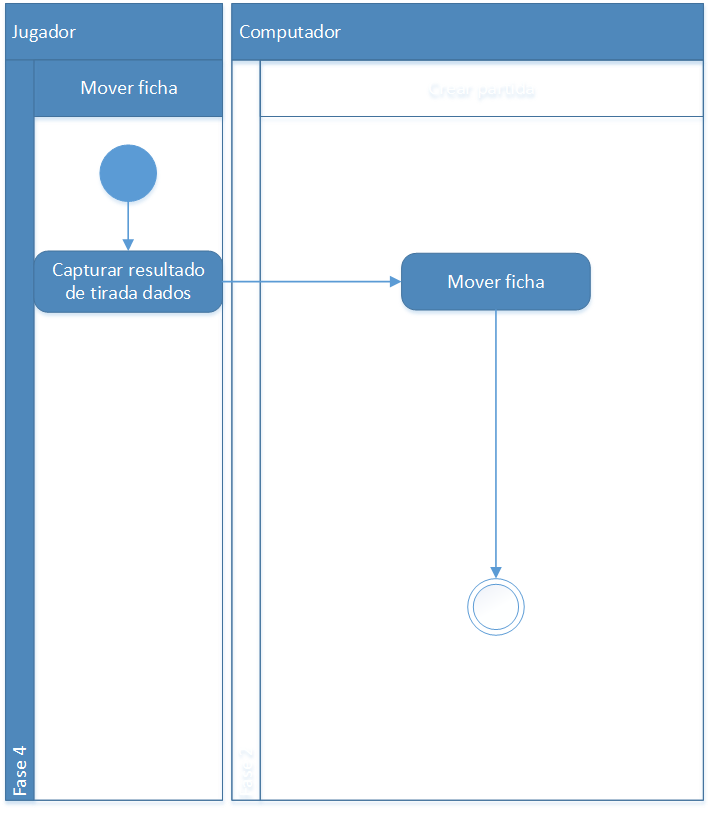
# Diagramas de actividad

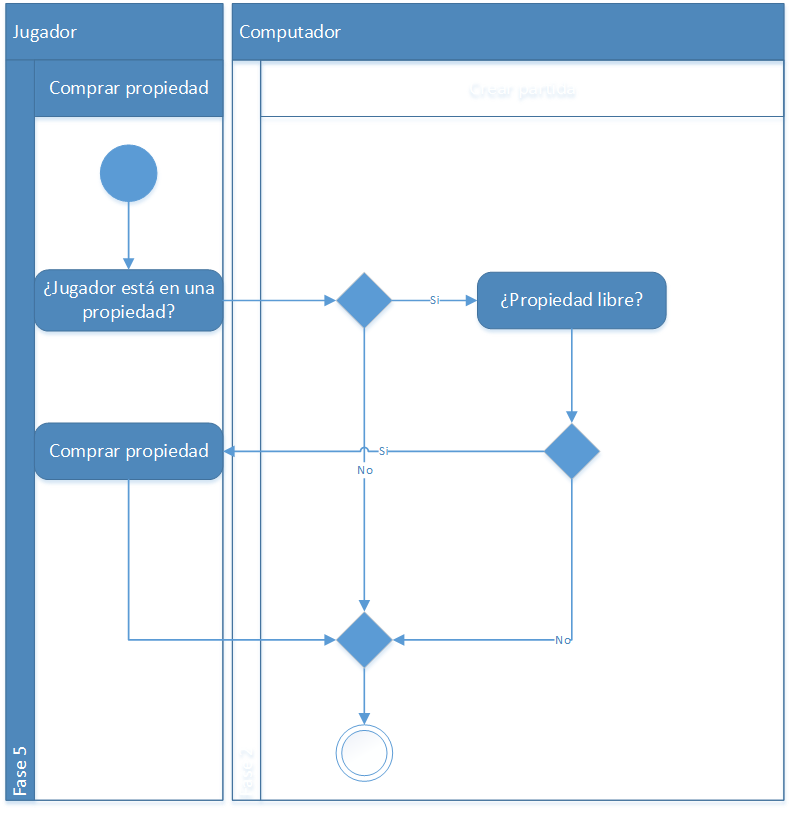
A modo informativo, se incluyen los diagramas de actividad por cada caso de uso que hemos capturado en los requisitos.

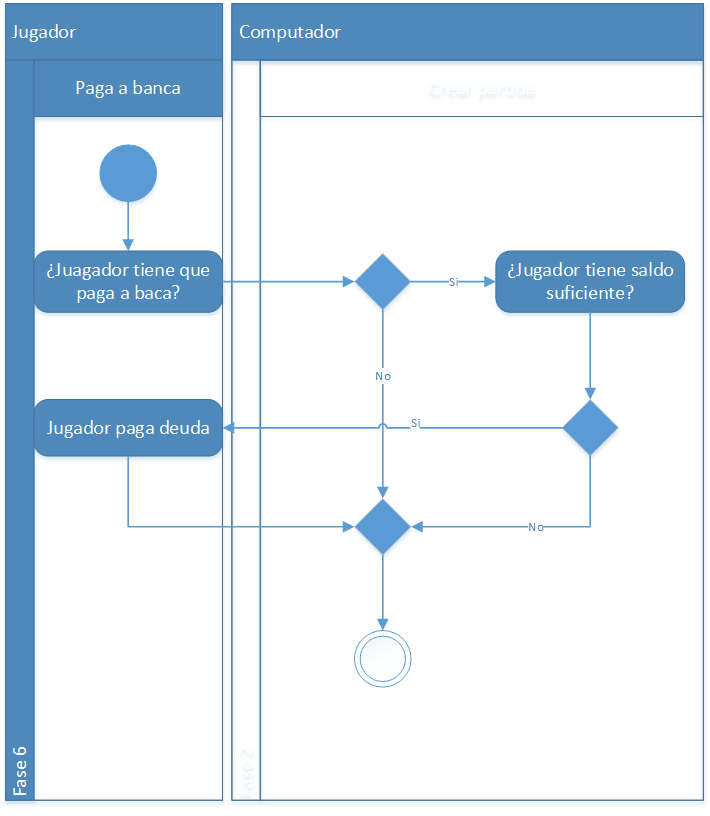


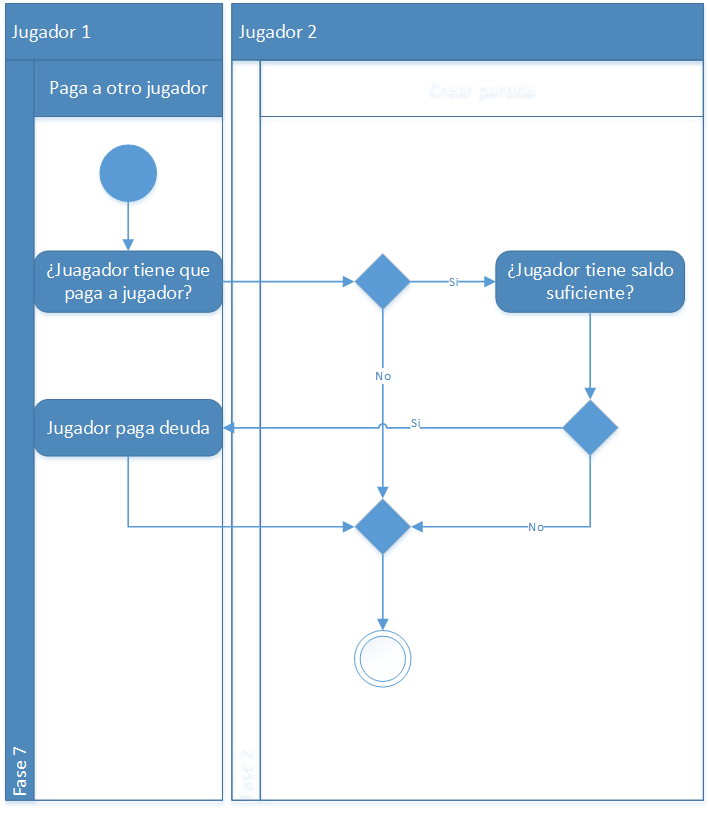


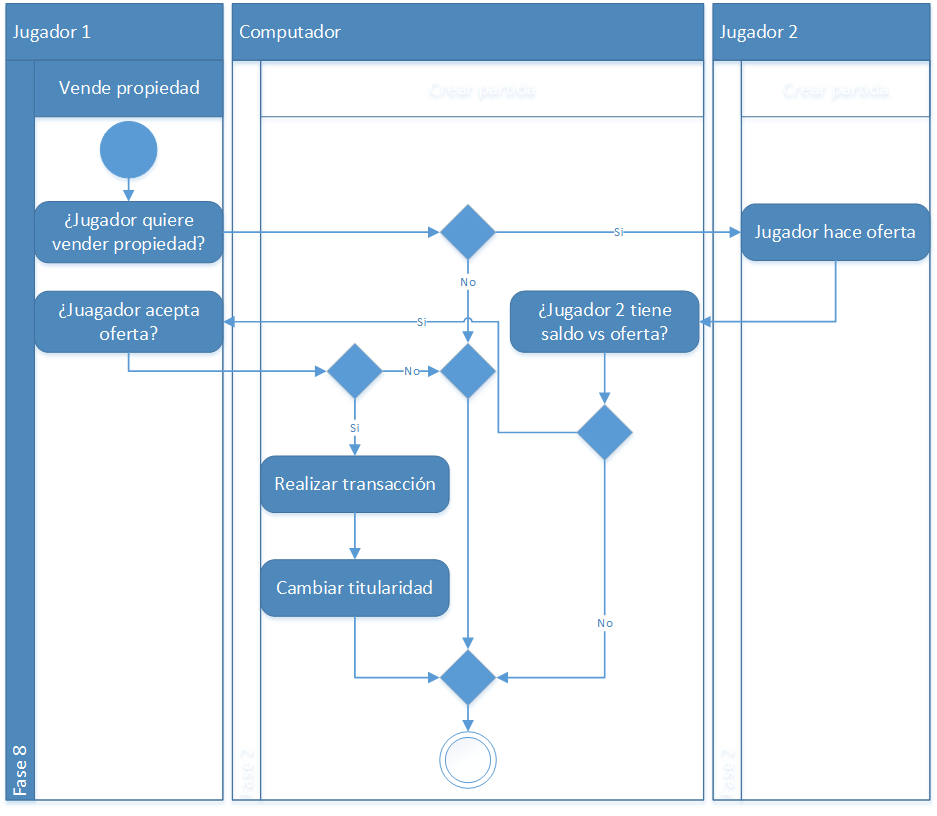


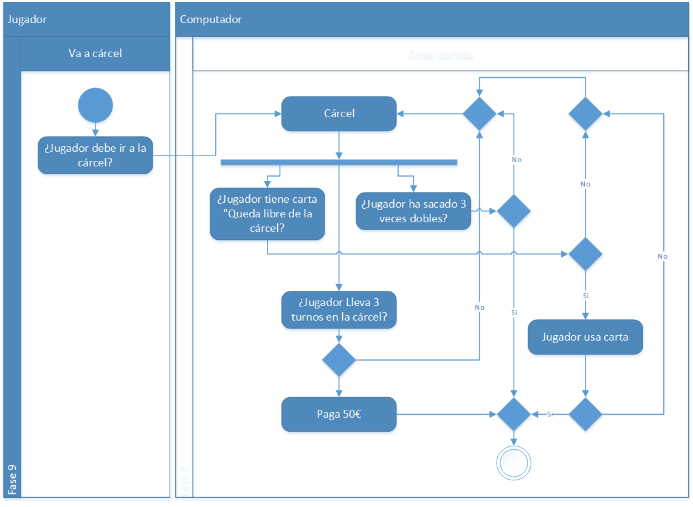


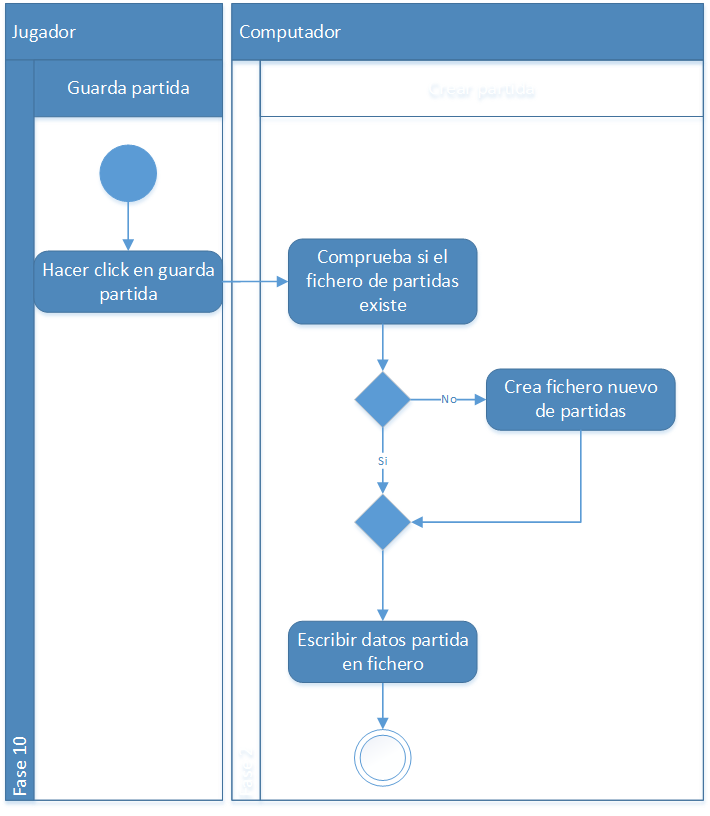


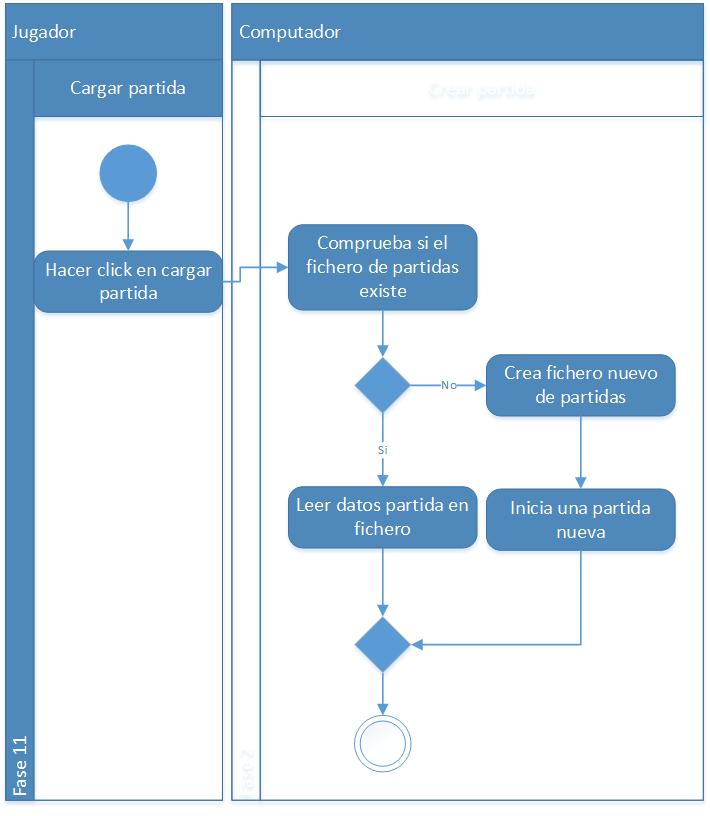


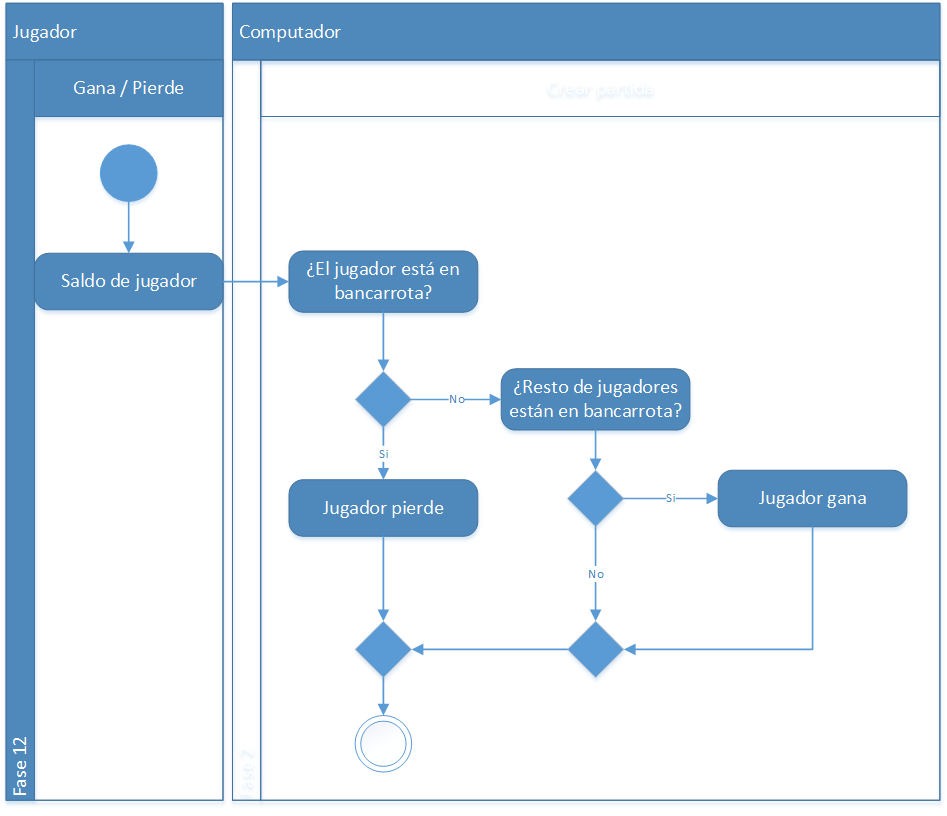


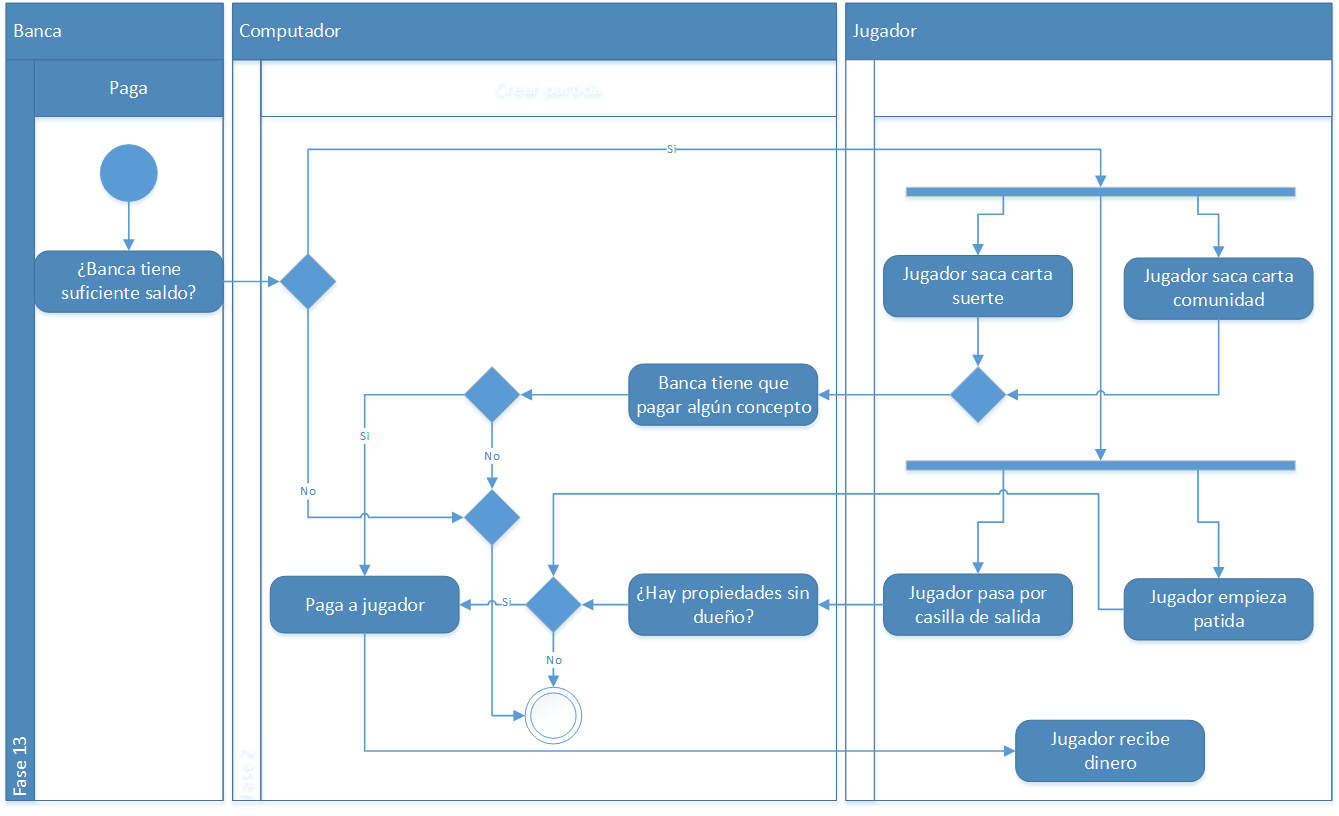


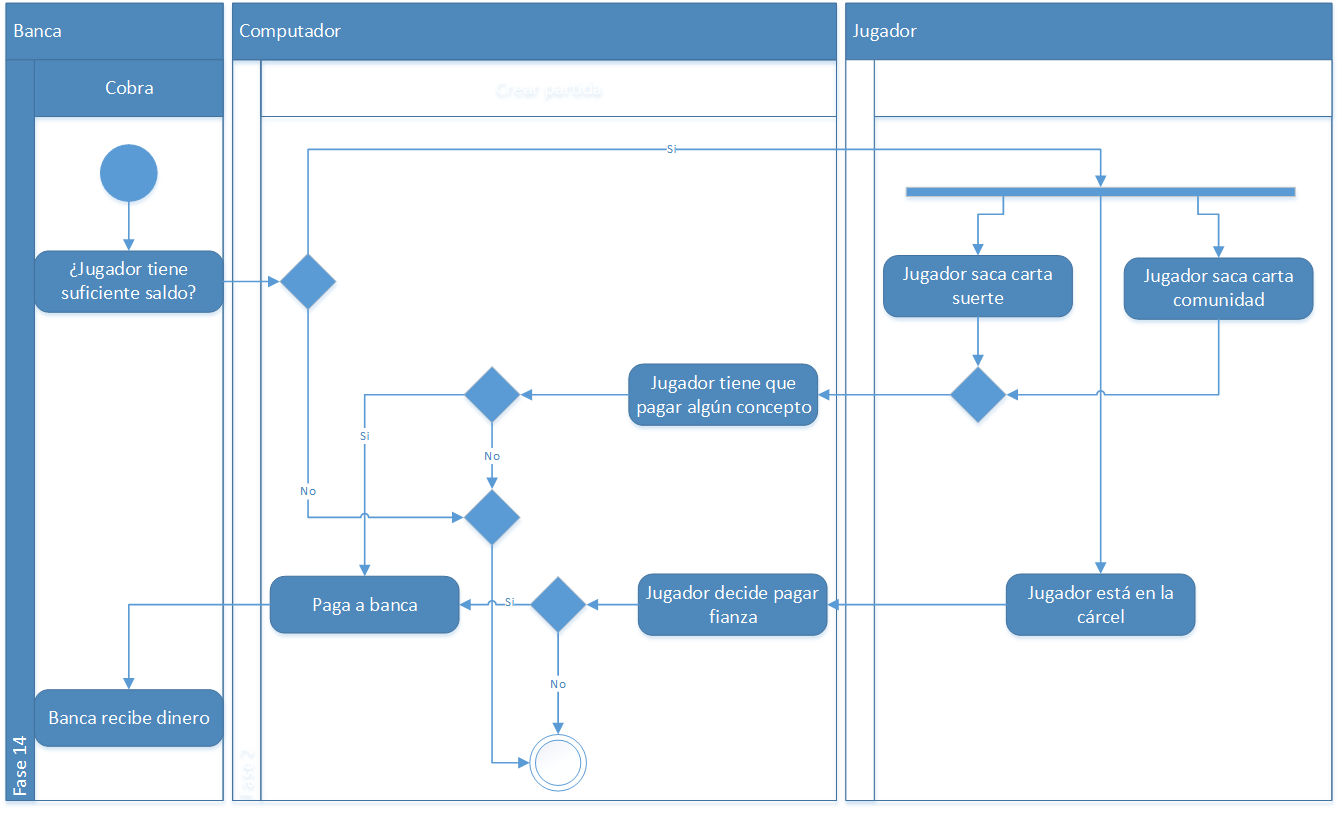






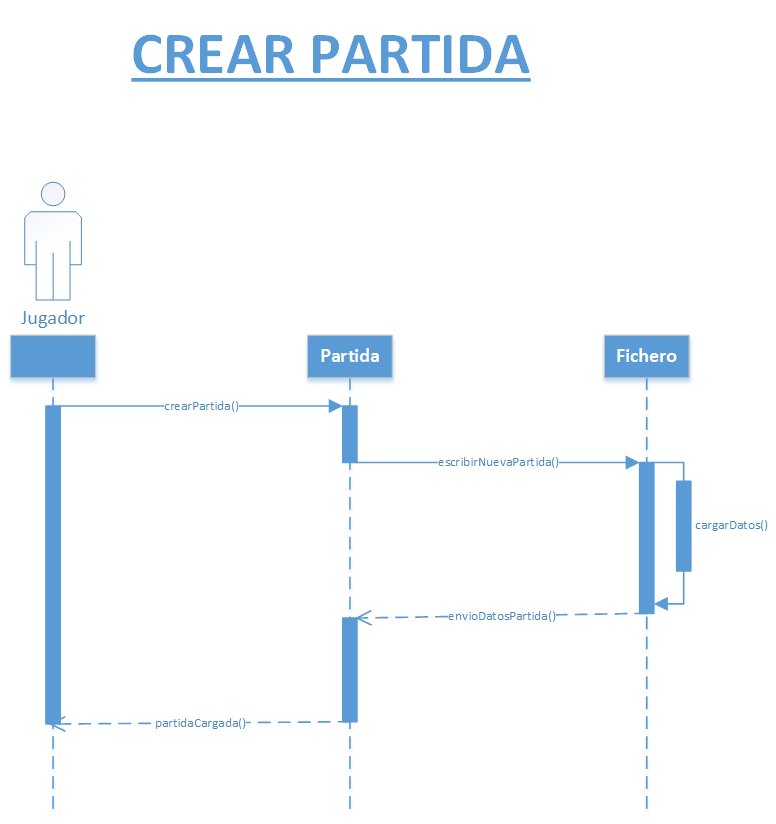


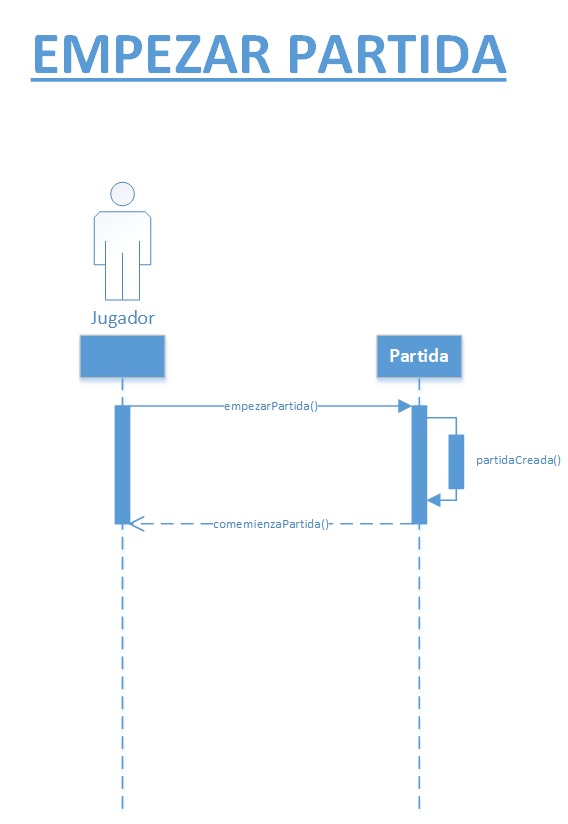


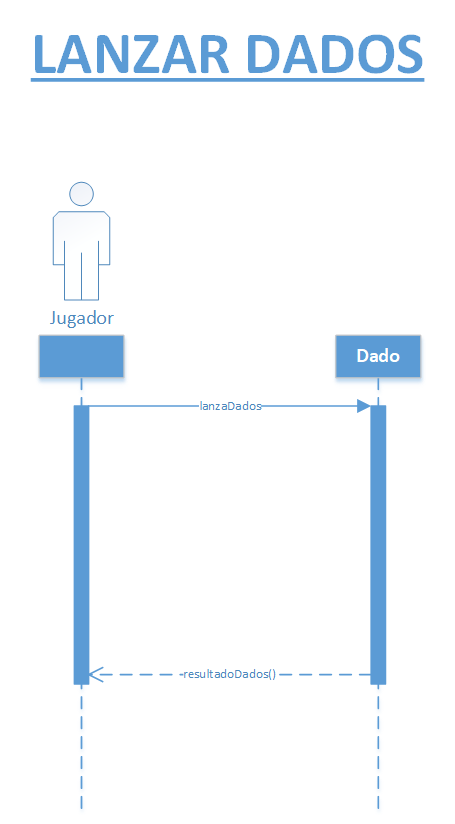


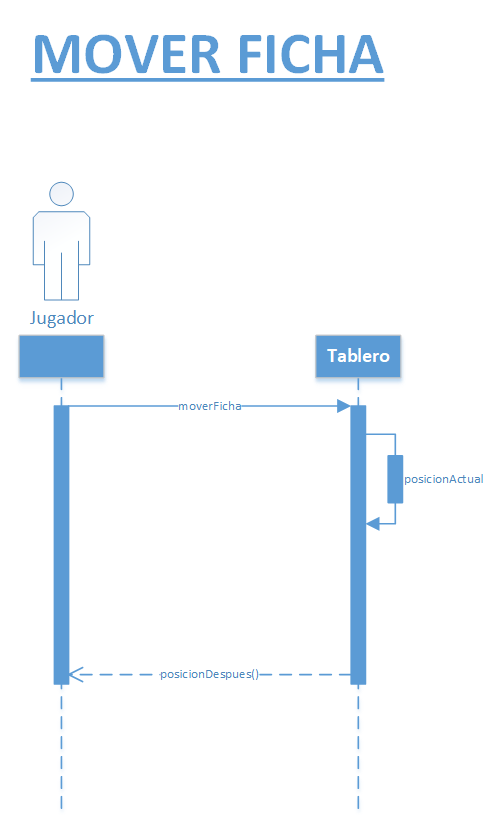
# Diagramas de secuencia

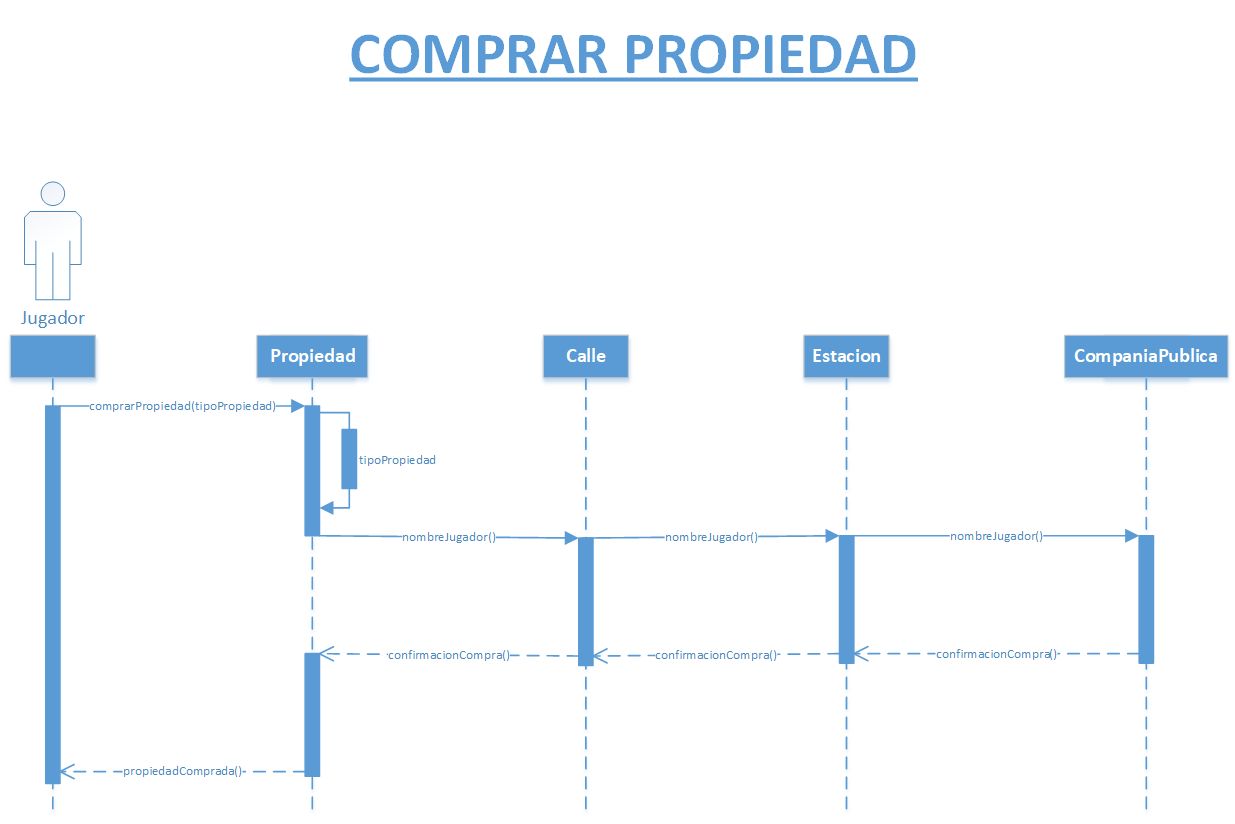
Se incluyen los diagramas de secuencia por cada caso de uso que hemos capturado en los requisitos. Como hemos hablado en otras ocasiones, estos diagramas pueden ser modificados a lo largo del desarrollo del proyecto por diferentes razones (captura de nuevos requisitos, errores no previstos, etc)

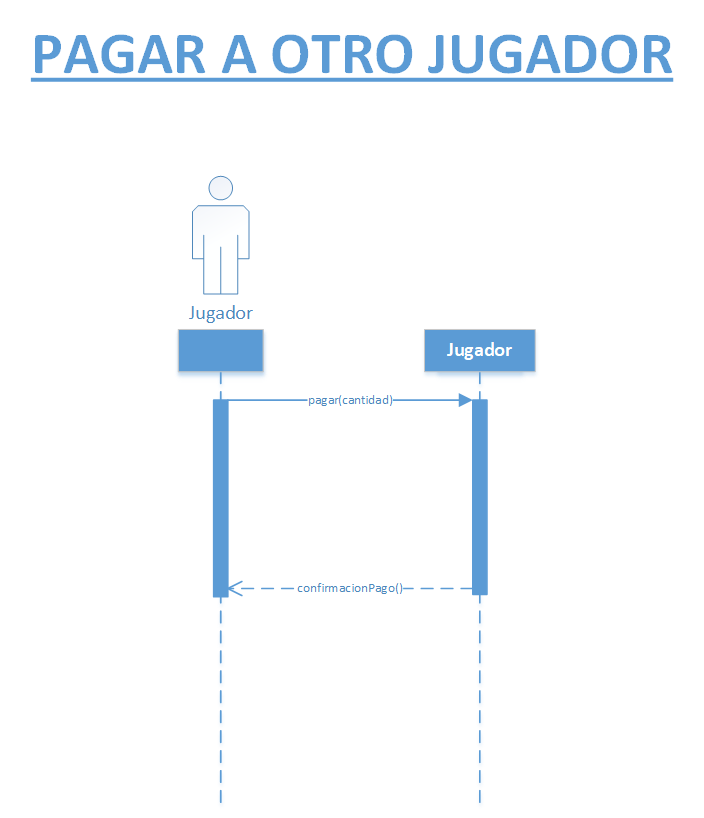


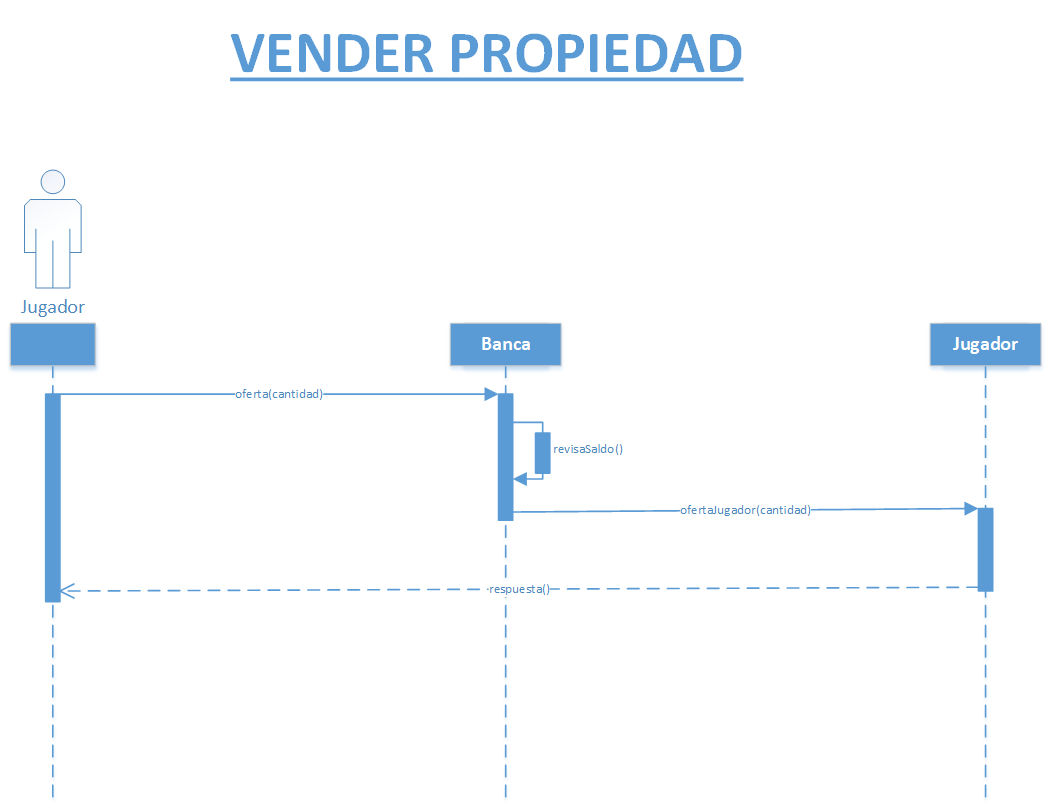


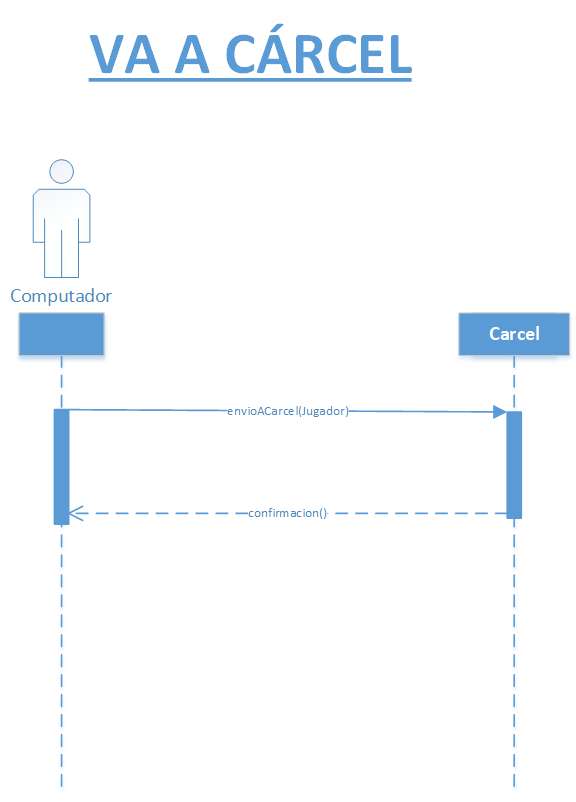


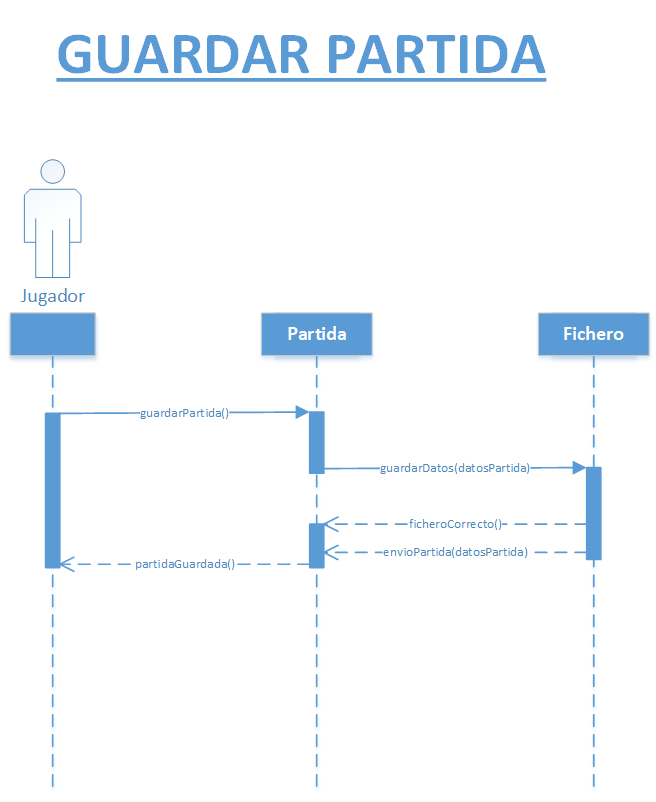


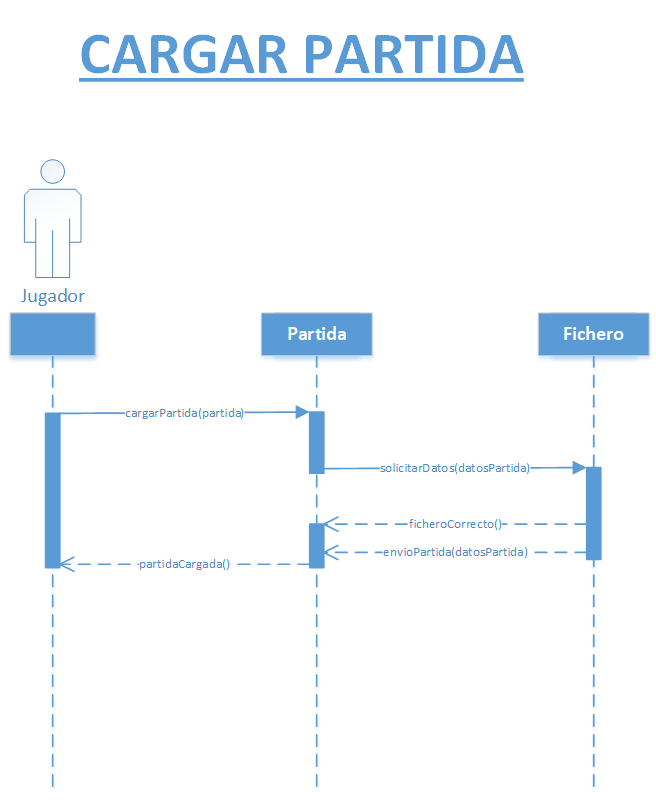


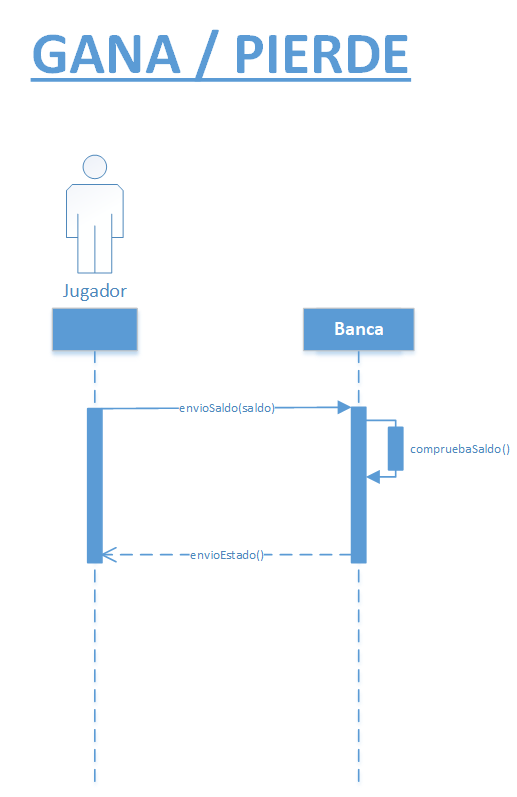


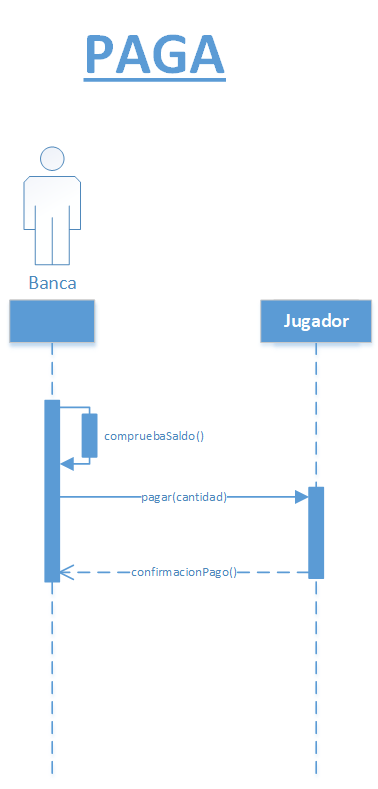


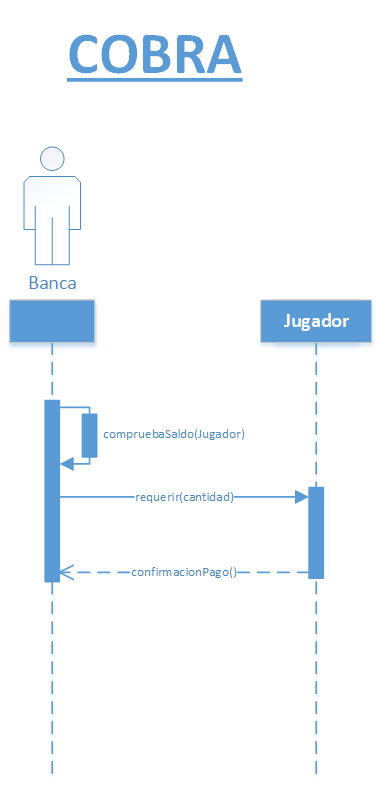












# Diagrama de componentes

A continuación, adjuntamos el diagrama de componentes. Que aunque es muy básico nos ayudará por si en el futuro queremos ampliar funcionalidades de nuestra aplicación.

