

UNIVERSIDAD PANAMERICANA

Documento completo del proyecto “Fluxx”

Aguascalientes, Aguascalientes 2018

Universidad Panamericana Campus Bonaterra

Eduardo Efrén González González

Marco Antonio Montoya Martín

Contenido

Manual de instalación y configuración del proyecto "Fluxx"	3
Introducción	3
Requerimientos de hardware y software	4
Instalación.....	4
Especificación de requisitos de software	5
Introducción	5
Propósito	5
Alcance.....	5
Personal involucrado	5
Definiciones, acrónimos y abreviaturas	5
Referencias	6
Resumen	6
Descripción general	6
Perspectiva del producto.....	6
Funcionalidad del producto.....	6
Características de los usuarios	7
Restricciones.....	7
Suposiciones y dependencias	7
Evolución previsible del sistema.....	7
Requisitos específicos.....	7
Requisitos funcionales.....	7
Requisitos no funcionales.....	8
Requisitos de rendimiento	8
Fiabilidad.....	8
Disponibilidad	8
Mantenibilidad	8
Portabilidad	9
Otros requisitos	9
Estudio de factibilidad	9
Planteamiento del problema.....	9

Arquitectura general	10
Consideraciones del software	10
Hosting.....	11
Manejador de bases de datos	11
Lenguaje de programación	11
Vista del flujo del sistema.....	11
Organigrama de actividades	12
Organización de las actividades.....	12
Plan de tareas	13
Red de actividades.....	14
Calendarización de tareas	15
Factibilidad	16
Factibilidad técnica	16
Factibilidad económica	16
Costos generales.....	16
Costos de hosting	16
Costos de personal	16
Costos operativos durante el desarrollo	17
Costos totales del desarrollo del sistema.....	17
Factibilidad operativa	18
Riesgos	18
Ventajas y desventajas	18
Beneficios tangibles.....	18
Beneficios intangibles	18
Conclusiones.....	19
Diagramas varios	20
Diagrama de casos de uso	20
Diagrama de clases	20
Matriz de trazabilidad.....	21

Manual de instalación y configuración del proyecto "Fluxx"

Introducción

En esta sección se describen los pasos a seguir para poder instalar el juego para celular "Fluxx" en un dispositivo móvil

Fluxx es un juego de cartas cuya base es el cambio constante. Las reglas y la meta del juego pueden ir cambiando conforme éste avanza, siendo definidas ambas con los movimientos de los participantes.

El juego termina cuando un jugador cumpla con las condiciones de la meta, siendo este jugador el ganador.

Requerimientos de hardware y software

Para instalar este juego se requiere de un teléfono con sistema operativo Android 6 o superior, o con sistema iOS 9 o superior.

Instalación

Android:

Para instalar la aplicación, el usuario tiene que entrar a la PlayStore, buscar la aplicación bajo el nombre de “Fluxx” y hacer clic en el botón de descarga. Después de eso, el usuario tiene que aceptar los términos y condiciones de la aplicación.

iOS:

Para instalar la aplicación, el usuario tiene que entrar a la AppStore, buscar la aplicación bajo el nombre de “Fluxx” y hacer clic en el botón de descarga. Después de eso, el usuario tiene que aceptar los términos y condiciones de la aplicación.

Especificación de requisitos de software

Introducción

A continuación, se presenta información relevante sobre el presente documento

Propósito

Este documento se redacta con el objetivo de enlistar todas las especificaciones necesarias para el correcto desarrollo de la aplicación “Fluxx para móviles”, así como todas sus características. Dicho juego es un juego para dispositivos móviles basado en el juego de cartas llamado Fluxx, de tal forma que no sea necesario lidiar con las cartas para poder disfrutar el juego.

Alcance

Se tendrá un prototipo de juego para móviles con las mismas características generales del juego de cartas Fluxx, de tal forma que varias personas puedan organizarse para jugar como si tuvieran las cartas.

Personal involucrado

Marco Antonio Montoya Martín
Eduardo Efrén González González

Definiciones, acrónimos y abreviaturas

SRS: Software Requirements Specification.

Android: Sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas.

iOS: Sistema operativo para dispositivos móviles manufacturado por Apple Inc.

Xamarin: Plataforma para desarrollar aplicaciones para plataformas iOS, Android, y Windows Phone.

Fluxx: Juego de cartas publicado por Looney Labs.

Modelo: Representación abstracta, conceptual, gráfica o visual, física, de fenómenos, sistemas o procesos a fin de analizar, describir, explicar, simular, explorar, controlar y predecir esos fenómenos o procesos.

MySQL: Lenguaje de código abierto para la gestión de bases de datos relacionales.

MVVM: Por sus siglas en inglés Modelo – Vista – Modelo de vista. Patrón de arquitectura que separa la interfaz del usuario de la lógica de una aplicación.

Referencias

Referencia	Título	Ruta
1	How to play Fluxx	http://www.wunderland.com/LooneyLabs/Fluxx/Rules.html
2	Xamarin Documentation	https://docs.microsoft.com/en-us/xamarin/
3	C# Documentation	https://docs.microsoft.com/en-us/dotnet/csharp/
4	XAML Documentation	https://msdn.microsoft.com/en-us/library/gg134030.aspx
5	Xamarin Forms: Build Native Cross-platform Apps with C#	https://www.udemy.com/xamarin-forms-course/

Resumen

El presente documento tiene la finalidad de presentar los requerimientos y procesos del juego Fluxx.

Fluxx es un juego de cartas cuya base es el cambio constante. Las reglas y la meta del juego pueden ir cambiando conforme éste avanza, siendo definidas ambas con los movimientos de los participantes.

El juego termina cuando un jugador cumpla con las condiciones de la meta, siendo este jugador el ganador.

Descripción general

Perspectiva del producto

El producto está conformado por una aplicación para móviles y un servidor. La aplicación permitirá a los usuarios jugar, mientras que el servidor gestionará la actividad del juego.

Funcionalidad del producto

La funcionalidad principal del producto se define en el documento de casos de uso.

Características de los usuarios

Tipo de usuario	Jugador
Formación	Indiferente
Habilidades	Capacidad de jugar juegos de cartas complejos.
Actividades	En sus ratos de ocio gusta de jugar con amigos

Restricciones

La aplicación funciona únicamente en dispositivos móviles. La aplicación no funcionará sin conexión a Internet.

Suposiciones y dependencias

Ya que el proyecto se desarrollará en Xamarin, se está confiando en su estabilidad. Si algo cambia es probable que se le tenga que actualizar la aplicación.

Evolución previsible del sistema

El juego de cartas original viene en varias versiones, cada una con una temática diferente. Se podrían implementar estas temáticas en el futuro para que la gente tenga más diversidad de juegos.

Requisitos específicos

Requisitos funcionales

Identificación	Sub identificación	Descripción del requisito
1		Se podrá iniciar un juego por completo nuevo, generando un código para ese juego
2		Se podrá unir a un juego que ya haya comenzado, utilizando el código de ese juego.
3		Se podrá unir a un juego aleatorio, para lo que no se pedirá ningún código
4	4.1	Habrán cuatro tipos de cartas: keepers, acción, reglas y metas
	4.2	Habrán cuatro tipos de reglas: límite de cartas en la mano, límite de keepers en juego, cartas para tomar y cartas para jugar
5		Todas las cartas jugadas tomarán efecto instantáneo
6	6.1	El jugador podrá elegir qué cartas ver, las cuales se mostrarán en pantallas diferentes.

6	6.2	Un conjunto de cartas que el usuario puede ver son las reglas en juego.
6	6.3	Un conjunto de cartas que el usuario puede ver es su mano.
6	6.4	Un conjunto de cartas que el usuario puede ver son los keepers en juego
6	6.5	El usuario puede ver en todo momento cuál es la meta actual
7	7.1	Todas las metas consistirán en tener un conjunto específico de dos keepers en juego
7	7.2	El juego terminará en cuanto un jugador tenga los dos keepers de la meta en juego
8		En un juego podrá haber un mínimo de dos y un máximo de seis jugadores
9		Cada jugador tendrá tres cartas al comenzar el juego
10		Un jugador podrá unirse en cualquier momento al juego, momento en que se le asignarán tres cartas
11		El primer turno de un juego corresponderá a la primera persona que se unió a éste
12		Si se juega una carta de regla que contradice a una regla anterior, la nueva será la que quede vigente.

Requisitos no funcionales

Requisitos de rendimiento

El turno de cada jugador debe de actualizarse en a lo más dos segundos.

Fiabilidad

Se requiere de conexión a Internet en todo momento para que la aplicación funcione.

Disponibilidad

Se requiere de conexión a Internet en todo momento para que la aplicación funcione

Mantenibilidad

El código estará documentado para que el mantenimiento sea más sencillo.

Portabilidad

- Programación de backend con C#.
- Programación de frontend con XAML.
- Uso de Xamarin para que la aplicación funcione en diferentes plataformas.

Otros requisitos

Se necesita el permiso de la empresa creadora del juego para poder distribuir la aplicación.

Estudio de factibilidad

A continuación, se describirán los problemas que se deberán de resolver en el desarrollo del juego para dispositivos móviles “Fluxx”, así como las herramientas técnicas que se planea utilizar para resolver dichos problemas.

Planteamiento del problema

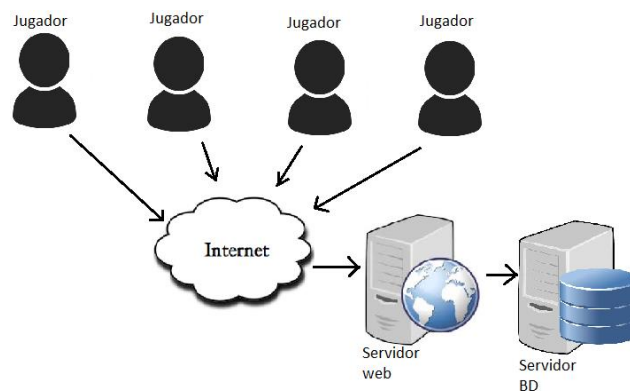
Se desea hacer un sistema en el que el usuario pueda acceder a una partida, y poder jugar con la gente que desee en esa partida hasta que uno de los jugadores gane según las reglas del juego.

Se deberá de contar con las siguientes funcionalidades principales:

- Crear una nueva partida.
- Unirse a una partida ya existente.
- Unirse a una partida aleatoria.
- En cualquiera de los casos anteriores, asignarse un nombre de usuario antes de comenzar la partida.
- Ver las cartas con las que cuento ahora.
- Poder ver la meta actual en todo momento.
- Deben de existir los siguientes tipos de carta: regla, keeper, acción, meta.
- Debe de haber gestión de turnos dentro de una partida.
- El juego debe terminar cuando una persona tenga en juego los keepers indicados por la carta de meta en juego.
- Al jugarse una carta de meta, la anterior debe de ser descartada.
- El juego debe de seguir las reglas indicadas en las tarjetas de reglas.
- Una nueva carta de regla en juego debe sumarse a las anteriores, a menos que contradiga a una de ellas, la cual queda descartada para ser sustituida por la nueva.
- Todas las cartas tienen un efecto inmediato.
- Las acciones se cumplen y después la carta se descarta.

Arquitectura general

En el siguiente diagrama se muestra una visión general de cómo funcionaría la arquitectura del juego para móviles.



Consideraciones del software

Requerimientos técnicos			
Descripción	Software	Costos	Operatividad
Hosting	Go Daddy	\$120.00 al mes	Ofrece 100 GB de almacenamiento, ancho de banda ilimitado y dominio gratis.
	Hostinger	Gratuito	Ofrece dos bases de datos, 100 GB de ancho de banda y 2000 MB de espacio de almacenamiento.
Manejador de bases de datos	MySQL	Gratuito	Es un sistema de bases de datos que funciona muy bien en diferentes plataformas.
Lenguaje de programación	C#	Gratuito	Utilizando la biblioteca de Xamarin se puede hacer un mismo código para iOS y Android.
	Kotlin / Swift	Gratuito	Estos lenguajes permiten entrar

			más a detalle en las características técnicas de cada sistema operativo.
--	--	--	--

Hosting

Se decidió optar por la opción de Go Daddy, ya que ofrece soporte técnico y porque, al tener un costo, garantiza que los servidores estarán funcionando.

Manejador de bases de datos

Con los servidores considerados sólo se puede usar MySQL, así que no es necesario comparar entre varias opciones.

Lenguaje de programación

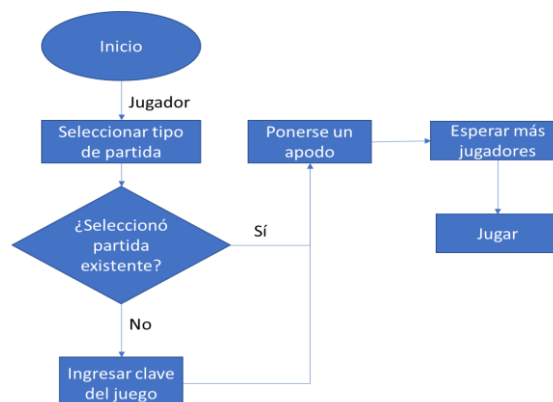
Se decidió usar C#, ya que no es un proyecto con muchos requerimientos técnicos específicos de los sistemas operativos, así que cómodamente se puede aprovechar la ventaja de usar un mismo lenguaje para dos sistemas operativos.

Vista del flujo del sistema

Conforme a lo mencionado anteriormente, el flujo del sistema consistirá en lo siguiente:

- El usuario selecciona un tipo de juego (partida nueva, aleatoria, unirse a existente).
- En caso de haber seleccionado unirse a una partida existente, indica a qué partida desea unirse.
- El usuario se pone un apodo en cualquiera de los tres casos.
- El usuario espera a que seis jugadores, incluyéndolo, estén en la partida.
- Una vez que están todos los jugadores, se le dirige a la partida seleccionada.
- En la partida, el jugador juega siguiendo las reglas en caso de ser su turno, y espera a que lo sea en caso contrario.

En el siguiente diagrama se muestra el flujo de la aplicación



Organigrama de actividades

El desarrollo de la aplicación se puede dividir en las siguientes etapas:

1. **Análisis de requerimientos:** Se definirán las características que tendrá la aplicación, así como lo que es necesario para su funcionamiento.
2. **Diseño del sistema:** Se buscarán soluciones para resolver desarrollar la aplicación, y se definirá las partes en que se dividirá y su comportamiento.
3. **Desarrollo del sistema:** Se crean las pantallas de la aplicación móvil, se programa el back-end y se crea la base de datos con la información necesaria.
4. **Pruebas y corrección de errores:** Se buscan errores cometidos durante el proceso del desarrollo de la aplicación, tanto de diseño como de implementación, y se corrigen.

Organización de las actividades

Etapas	Actividades generales	Tiempo asignado (horas)
Análisis de requerimientos	Definir requerimientos funcionales	3
	Definir requerimientos no funcionales	3
	Definir pantallas del sistema	2
	Definir flujo del sistema	2
	Definir cartas	1
Diseño del sistema	Diseño de la base de datos	3
	Diseño de la arquitectura	3
	Diseño de los componentes	2
	Definición de funcionalidades	5
Desarrollo del sistema	Construcción de vistas	8
	Construcción de bases de datos	2
	Construcción de funcionalidades	12
	Construcción de pruebas unitarias	4
	Unión de módulos	3
Pruebas del sistema y corrección de errores	Pruebas de las vistas	1
	Prueba del ciclo del juego	8

	Prueba del acceso de usuario	1
	Corrección de errores	4

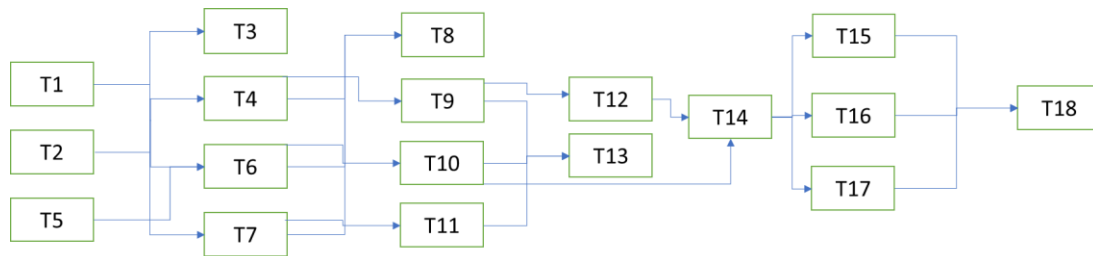
Plan de tareas

Tareas	Actividades generales	Duración (horas)	Dependencia entre tareas
T1	Definir requerimientos funcionales	3	
T2	Definir requerimientos no funcionales	3	
T3	Definir pantallas del sistema	2	T1, T2
T4	Definir flujo del sistema	2	T1, T2
T5	Definir cartas	1	
T6	Diseño de la base de datos	3	T1, T2, T5
T7	Diseño de la arquitectura	3	T1, T2
T8	Diseño de los componentes	2	T4, T6, T7
T9	Definición de funcionalidades	5	T4
T10	Construcción de vistas	8	T4
T11	Construcción de bases de datos	2	T6
T12	Construcción de funcionalidades	12	T9
T13	Construcción de pruebas unitarias	4	T9, T10, T11
T14	Unión de módulos	3	T10, T12
T15	Pruebas de las vistas	1	T14
T16	Prueba del ciclo del juego	8	T14
T17	Prueba del acceso de usuario	1	T14









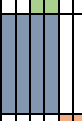


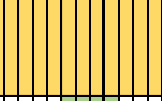






T18	Corrección de errores	4	T15, T16, T17
-----	-----------------------	---	---------------

Red de actividades

La red de actividades muestra la dependencia entre las actividades a realizar, la cual es la siguiente:



Calendarización de tareas

Tareas	Actividades generales	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6
T1	Definir requerimientos funcionales						
T2	Definir requerimientos no funcionales						
T3	Definir pantallas del sistema						
T4	Definir flujo del sistema						
T5	Definir cartas						
T6	Diseño de la base de datos						
T7	Diseño de la arquitectura						
T8	Diseño de los componentes						
T9	Definición de funcionalidades						
T10	Construcción de vistas						
T11	Construcción de bases de datos						
T12	Construcción de funcionalidades						
T13	Construcción de pruebas unitarias						
T14	Unión de módulos						
T15	Pruebas de las vistas						
T16	Prueba del ciclo del juego						
T17	Prueba del acceso de usuario						
T18	Corrección de errores						

Factibilidad

En este apartado se evaluará la factibilidad del proyecto descrito en las secciones anteriores, se mostrará el análisis técnico, económico y operativo que implicaría el juego para móviles “Fluxx”.

Factibilidad técnica

Como se usarán servidores y el usuario utilizará su dispositivo, no es necesario gastar en equipo, aunque se pagará un gasto anual de hospedaje.

Factibilidad económica

Gracias a que el juego se puede jugar en cualquier ocasión social y a que todos los usuarios cuentan con un dispositivo móvil, se considera que el juego se puede monetizar de forma sencilla.

Se pondrán anuncios dentro de la aplicación para poder mantener los servidores en línea y poder pagar los sueldos de los programadores.

Costos generales

No es necesario tomar este tipo de gastos en cuenta, ya que representan materiales de oficina y papelería, y el trabajo no tendrá un lugar fijo.

Costos de hosting

Aquí se toman en cuenta los costos anuales generados por hosting, los cuales serán los siguientes:

Costos de ambiente	Costo mensual
Servicio de hosting (goDaddy)	\$120.00 (\$1440.00 anuales)
TOTAL	\$120.00

Costos de personal

Aquí se desglosan los costos generados por desarrollar la aplicación, que consistirá en un líder de proyecto y dos programadores. Se considera un tiempo de entrega de seis meses.

Puesto	Descripción de actividades	Salario por hora	Salario mensual	Cantidad de personal	Total por mes
Líder de proyecto	Coordinador general del proyecto, así como del desarrollo	\$100	\$16000	1	\$16000
Programador	Programación del juego y cada uno de sus módulos	\$80	\$12800	2	\$25600
TOTAL			\$28800		\$21600

Costos operativos durante el desarrollo

Estos costos se refieren a la renta de oficinas. Como no se usarán en este caso, no hay ningún costo.

Costos totales del desarrollo del sistema

A continuación, se muestra la tabla con los costos totales del sistema.

*El costo de hosting se incluirá por un año.

Concepto de pago	Costo mensual	Costo para 6 meses
Costos de ambiente	\$120.00	\$1,440.00*
Costos de personal	\$21,600.00	\$129,600.00
TOTAL	\$21,720.00	\$131,040.00
IVA	\$3,475.20	\$20,966.40
TOTAL + IVA	\$25,195.20	\$152,006.40

En la tabla se muestra un total que incluye los seis meses de operaciones, además de un año de hosting, dado como total un costo final de \$152,006.40. Dicho pago deberá cubrirse en dos exhibiciones:

Primer pago: 50% del costo total al iniciar el proyecto.

Segundo pago: 50% del costo total al terminar el proyecto.

Factibilidad operativa

La interfaz es amigable y está orientada a usuarios con un dispositivo móvil o una computadora. El usuario tendrá una cuenta donde se guardará su progreso y las estadísticas del juego. La velocidad del juego dependerá de la velocidad del dispositivo.

Riesgos

En esta sección se presentará el análisis de los riesgos que se deberán considerar durante el desarrollo del sistema y que pudieran ocurrir durante el uso de este que pudieran ser:

- En caso de no tener internet, el usuario no podrá jugar.
- Si el usuario se queda sin internet durante el juego, se considerará que se ha desconectado y no podrá volver a entrar al juego.

Ventajas y desventajas

Los beneficios del juego que proponemos son que se puede jugar de una forma mucho más barata que comprando las cartas, las cuales pueden ser costosas para algunas personas. También se podría jugar con gente sin importar dónde esté, y no se tendría preocupación por que alguna carta del juego se pierda.

Como desventaja se puede mencionar la dependencia de internet para poder jugar.

Beneficios tangibles

El juego aportaría los siguientes beneficios tangibles:

- No es necesario adquirir el juego en físico para jugar.
- Se pueden recibir actualizaciones del juego para que éste no aburra.

Beneficios intangibles

- Se facilitará la forma de juego.
- Se evita el riesgo de que se dañen o pierdan las cartas.
- Se podrá tener un control de las estadísticas del juego.

Conclusiones

Se pudo observar que el proyecto es viable porque representaría un beneficio para gente que disfruta de juegos de cartas, pero no tiene el dinero para gastar en uno, o no tiene la intención de hacerlo por diversos motivos.

Podría hacer que se reúna más en sus reuniones sociales y así fomentar las relaciones interpersonales con sus compañeros de trabajo, familiares, amistades, y otra gente que la rodee.

El costo final del juego será de \$152,006.40, que se recuperará en dos pagos del 50% cada uno.

Diagramas varios

En esta sección se muestran diversos diagramas que describen la aplicación.

Diagrama de casos de uso

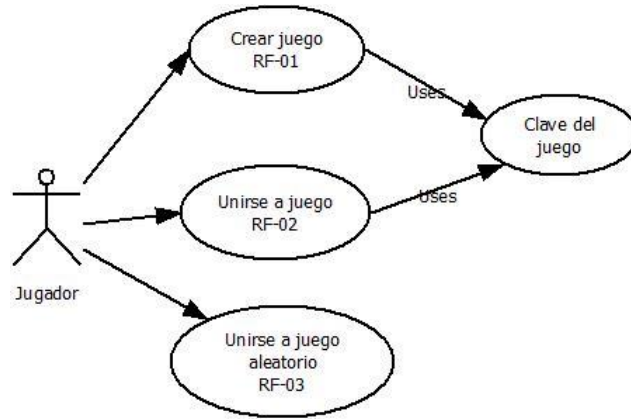
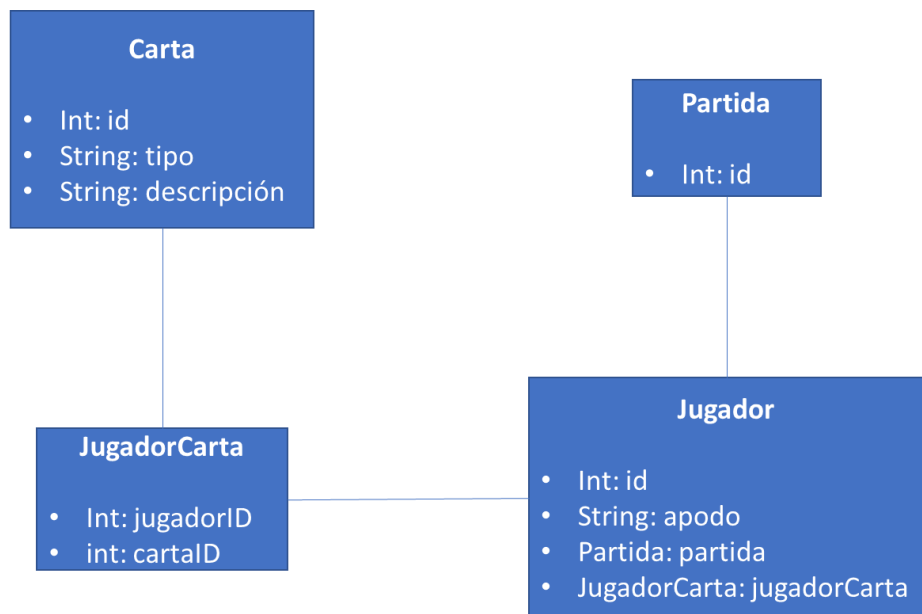


Diagrama de clases



Matriz de trazabilidad

Identificación	Sub identificación	Descripción del requisito	Versión	Estado actual	Última fecha estado registrado	Criterios de aceptación	Nivel de complejidad	Entregables (EDT)
1		Se podrá iniciar un juego por completo nuevo, generando un código para ese juego	1.1	Aprobado	21/03/2018	1. Se generará un código al iniciar un juego nuevo. 2. Se guardará ese código para que más personas puedan usarlo.	Media	Pantalla de inicio con funcionalidad de crear nuevo juego
2		Se podrá unir a un juego que ya haya comenzado, utilizando el código de ese juego.	1.1	Aprobado	21/03/2018	1. Se podrá usar un código generado para unirse a un juego. 2. Un código inválido no redirigirá a ningún juego.	Media	Pantalla de inicio con funcionalidad de unirse a un juego
3		Se podrá unir a un juego aleatorio, para lo que no se pedirá ningún código	1.1	Aprobado	21/03/2018	1. Se generará un código privado para juegos nuevos públicos, y éste se usará para que la gente se una. 2. Si no hay juegos de jugadores aleatorios disponibles, se generará uno.	Alta	Pantalla de inicio con funcionalidad de unirse a un juego aleatorio
4	4.1	Habrán cuatro tipos de cartas: keepers, acción, reglas y metas	1.1	Aprobado	21/03/2018	Existirá una forma de representar todos los tipos de cartas	Media	Clases en que se especifiquen las características de cada tipo de carta
	4.2	Habrán cuatro tipos de reglas: límite de cartas en la mano, límite de keepers en juego, cartas para tomar y cartas para jugar	1.1	Aprobado	21/03/2018	Existirá una forma de representar todos los tipos de reglas	Media	Clases en que se especifiquen las características de cada tipo de regla
5		Todas las cartas jugadas tomarán efecto instantáneo	1.1	Aprobado	21/03/2018	Las cartas tendrán un efecto en el juego al ser jugadas.	Media	El juego reacciona al momento en que una carta es jugada
6	6.1	El jugador podrá elegir qué cartas ver, las cuales se mostrarán en pantallas diferentes.	1.1	Aprobado	21/03/2018	1. Habrá una sección para cada conjunto de cartas que el usuario puede ver. 2. No podrá ver las cartas en la mano de otros	Media	Pantallas de cada sección
6	6.2	Un conjunto de cartas que el usuario puede ver son las reglas en juego.	1.1	Aprobado	21/03/2018	Se pueden ver las reglas que están actualmente en juego	Baja	Pantalla que muestre las reglas en juego
6	6.3	Un conjunto de cartas que el usuario puede ver es su mano.	1.1	Aprobado	21/03/2018	Se pueden ver las cartas que el usuario tiene en su mano.	Baja	Pantalla que muestre la mano del jugador
6	6.4	Un conjunto de cartas que el usuario puede ver son los keepers en juego	1.1	Aprobado	21/03/2018	Se pueden ver los keepers que tanto el usuario como sus contrincantes han jugado	Baja	Pantalla que muestre los keepers
6	6.5	El usuario puede ver en todo momento cuál es la meta	1.1	Aprobado	21/03/2018	Se puede ver la meta actual del juego	Baja	La meta visible en todas las pantallas
7	7.1	Todas las metas consistirán en tener un conjunto específico de dos keepers en juego	1.1	Aprobado	22/03/2018	Se tienen metas que cumplen las características mencionadas.	Baja	Vistas de las tarjetas de meta funcionales
7	7.2	El juego terminará en cuanto un jugador tenga los dos keepers de la meta en juego.	1.1	Aprobado	23/03/2018	El sistema identificará el momento en que los dos keepers en juego de un usuario coincidan con los keepers de la meta	Alta	Juego principal capaz de terminar
8		En un juego podrá haber un mínimo de dos y un máximo de seis jugadores	1.1	Aprobado	24/03/2018	El juego no comenzará si no hay al menos dos personas, y no dejará entrar a más de seis.	Media	Juego que no permite movimientos cuando sólo hay un jugador y no permite que entren más de seis
9		Cada jugador tendrá tres cartas al comenzar el juego	1.1	Aprobado	25/03/2018	El sistema le asignará tres cartas a cada jugador que se una a un juego.	Baja	Funcionalidad de nuevo jugador que incluya el reparto de las cartas
10		Un jugador podrá unirse en cualquier momento al juego, momento en que se le asignarán tres cartas	1.1	Aprobado	26/03/2018	El sistema permitirá a un usuario unirse a una partida, siempre que haya espacio.	Baja	Pantalla de inicio, ligada a pantalla de juego por medio del código de un juego
11		El primer turno de un juego corresponderá a la primera persona que se unió a éste	1.1	Aprobado	27/03/2018	Los usuarios irán jugando en el orden en que se unieron al juego	Baja	Pantalla de juego con la funcionalidad de cambio de turno dependiendo del orden de ingreso
12		Si se juega una carta de regla que contradice a una regla anterior, la nueva será la que quede vigente.	1.1	Aprobado	27/03/2018	Se reemplazarán reglas viejas cuando las nuevas las contradigan	Media	Pantalla de juego con funcionalidad de cambio de reglas
13		Al inicio del juego, las reglas consisten en tomar una carta y dejar una carta	1.1	Aprobado	27/03/2018	El juego comenzará con las reglas mencionadas	Baja	Pantalla de juego con la reglas iniciales siendo las mencionadas
14		Se requiere de conexión a Internet en todo momento para que la aplicación funcione	1.1	Aprobado	27/03/2018		Media	
15		Programación de backend con C#	1.1	Aprobado	27/03/2018		Baja	
16		Programación de frontend con XAML	1.1	Aprobado	27/03/2018		Baja	
17		Uso de Xamarin para que la aplicación funcione en diferentes plataformas.	1.1	Aprobado	27/03/2018		Baja	
18		El código estará documentado para que el mantenimiento sea más sencillo	1.1	Aprobado	27/03/2018		Alta	
19		Se necesita el permiso de la empresa creadora del juego	1.1	Aprobado	27/03/2018		Alta	
20		El turno de cada jugador debe de actualizarse en a lo más dos segundos	1.1	Aprobado	27/03/2018		Media	