

PARCHÍS

Presentado por:

Juan Manuel Jaraba
Jose Camilo Aguirre Silva

Docente:

Darmabel Orozco

Bogotá, 01/06/2023

**FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
0553 ingeniería de Software**

PARCHIS EN C#

Tabla de Contenido

1. Especificación de Requerimientos de Usuario	1
1.1. Nombre del Proyecto	1
1.2. Descripción del Proyecto.....	1
1.3. Alcance del Proyecto.....	1
1.4. Requerimientos de Usuario	2
1.4.1. Tablero	2
1.4.2. DetenerJuego.....	2
1.4.3. IniciarJuego.....	2
1.4.4. NextTurn.....	2
1.4.5. HayGanador.....	3
1.4.6. LanzarDados.....	3
1.4.7. PuedeMoverFicha.....	3
1.4.8. PuedeMoverCasa	4
1.4.9. PuedeMover.....	4
1.4.10. EstablecerEquipoFichas	4
1.4.11. ObtenerEquipoFichas	5
1.4.12. EstablecerValorDado.....	5
1.4.13. Mover.....	5
1.4.14. Preview.....	6
1.4.15. TestPosition	6
1.4.16. MoverACasa	6
1.4.17. NuevaFicha	6
1.4.18. EnviarCasa (Es Cárcel)	7
1.4.19. PuestoLibreCasa	7
1.4.20. FixPos	7
1.4.21. EsSeguro.....	8
1.4.22. FichasEn	8
1.4.23. EsPuente.....	8
1.4.24. EsPuenteCasa	9
1.4.25. Log.....	9
1.4.26. Resaltador.....	9
1.4.27. CasillaEtapaFinal	10

1.4.28. WinColores.....	10
1.4.29. Casillas.....	10
1.4.30. SeleccionarJugadores	11
1.4.31. HabilitarColoresFicha	11
1.4.32. SaltarTurno	11
1.4.33. BotonDado	11
1.4.34. DiceGrafth	12
1.4.35. GetCoin	12
1.4.36. PreviewCoin	12
1.4.37. HidePreviewCoin	13
1.4.38. CoinClick	13
1.4.39. PasaTurno_Tick.....	13
1.4.40. Ficha.....	13
1.4.41. Main.....	14
2. Diagrama de Casos de Uso.....	15
2.1.1. Diagrama.....	15
3. Especificación de Casos de Uso.....	16
3.1. Formato Especificación Caso de Uso 1 - Tablero	16
3.2. Formato Especificación Caso de Uso 2 - DetenerJuego	17
3.3. Formato Especificación Caso de Uso 3 - IniciarJuego.....	18
3.4. Formato Especificación Caso de Uso 4 - NextTurn	20
3.5. Formato Especificación Caso de Uso 5 - HayGanador.....	21
3.6. Formato Especificación Caso de Uso 6 - LanzarDados	22
3.7. Formato Especificación Caso de Uso 7 - PuedeMoverFicha	24
3.8. Formato Especificación Caso de Uso 8 - PuedeMoverCasa.....	26
3.9. Formato Especificación Caso de Uso 9 – PuedeMover.....	27
3.10. Formato Especificación Caso de Uso 10 – EstablecerEquipoFichas	28
3.11. Formato Especificación Caso de Uso 11 - ObtFichasEquipo	30
3.12. Formato Especificación Caso de Uso 12 - EstablecerValorDado.....	31
3.13. Formato Especificación Caso de Uso 13 - Mover	32
3.14. Formato Especificación Caso de Uso 14 - Fantasma.....	34
3.15. Formato Especificación Caso de Uso 15 - TestPos.....	35
3.16. Formato Especificación Caso de Uso 16 – MoverACasa.....	36
3.17. Formato Especificación Caso de Uso 17 - NuevaFicha	38

3.18. Formato Especificación Caso de Uso 18 - EnviarACasa	39
3.19. Formato Especificación Caso de Uso 19 - PuestoLibreCasa	40
3.20. Formato Especificación Caso de Uso 20- FixPos	42
3.21. Formato Especificación Caso de Uso 21- EsSeguro	44
3.22. Formato Especificación Caso de Uso 22- FichasEn	45
3.23. Formato Especificación Caso de Uso 23- EsPuente	46
3.24. Formato Especificación Caso de Uso 24- EsPuenteCasa	48
3.25. Formato Especificación Caso de Uso 25 - Log	49
3.26. Formato Especificación Caso de Uso 26 - Resaltador	51
3.27. Formato Especificación Caso de Uso 27 - CasillaEtapaFinal	52
3.28. Formato Especificación Caso de Uso 28 - WinColores	53
3.29. Formato Especificación Caso de Uso 29 - Casillas	54
3.30. Formato Especificación Caso de Uso 30 - SeleccionarJugadores	56
3.31. Formato Especificación Caso de Uso 31 - HabilitarColoresFicha	57
3.32. Formato Especificación Caso de Uso 32 - SaltarTurno	58
3.33. Formato Especificación Caso de Uso 33 - BotonDado	59
3.34. Formato Especificación Caso de Uso 34 – DiceGrafth	61
3.35. Formato Especificación Caso de Uso 35 – GetCoin	62
3.36. Formato Especificación Caso de Uso 35 – PreviewCoin	63
3.37. Formato Especificación Caso de Uso 37 – HidePreviewCoin	64
3.38. Formato Especificación Caso de Uso 38 – CoinClick	65
3.39. Formato Especificación Caso de Uso 39 – PasaTurno_Click	67
3.40. Formato Especificación Caso de Uso 40 – Ficha	68
3.41. Formato Especificación Caso de Uso 41 – Main	70
4. Diagrama de Clases	72
4.1. Diagrama de Clases Parchidos	72
5. Diagrama de Secuencia	74
5.1. Diagrama de secuencia Parchidos	74
6. Diagrama de Colaboración	77
7. Contratos de Comportamiento del Sistema	78
7.1. Contratos de Comportamiento Clase 1	78
7.2. Contratos de Comportamiento Clase 2	79
7.3. Contratos de Comportamiento Clase 3	83

1. Especificación de Requerimientos de Usuario

1.1. Nombre del Proyecto

Parchís en C#

1.2. Descripción del Proyecto

El proyecto consta de varias etapas, dentro de las cuales se encuentra una instrucción inicial con las generalidades de su funcionamiento y desarrollo bajo la herramienta de Visual Studio 2019. El objetivo de desarrollo es la presentación del juego clásico de parques (Parchís), con sus reglas básicas de juego, 4 Fichas del color de su casa (cárcel), primera salida únicamente con un numero par en los dados, mover a la cárcel la ficha del cárcel), primera salida únicamente con un numero par en los dados, mover a la cárcel la ficha del jugador oponente si este es alcanzado en la misma casilla por una ficha de otro color, manejo de las casillas “seguras”; Ganando el jugador que primero lleve todas sus fichas a la meta (Cielo) ya sea porque recorrió todas las casillas del tablero o porque saco 3 pares seguidos para retirar cada una de las 4 fichas.

A continuación, se expondrá el detalle del proyecto como lo son requerimientos, casos de uso y diagramas.

1.3. Alcance del Proyecto

El alcance se limita a la exposición de los requerimientos actuales en el desarrollo del juego, donde este podrá ser modificado a medida que el mismo sufra actualizaciones en su desarrollo y ejecución.

Por ende, este documento es meramente explicativo y no reflejara en su totalidad la muestra final del juego.

1.4. Requerimientos de Usuario

1.4.1. Tablero

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Tablero
Características:	Asignación del tablero de juego
Descripción del requerimiento:	Establecer el tablero de juego con sus casillas, casas, cielo y casillas especiales.
Prioridad del requerimiento:	Media

1.4.2. DetenerJuego

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	DetenerJuego
Características:	Funcionalidad que establece juego terminado
Descripción del requerimiento:	Si un jugador no cuenta con más fichas en juego, automáticamente el juego terminará
Prioridad del requerimiento:	Alta

1.4.3. IniciarJuego

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	IniciarJuego
Características:	Establecer las condiciones para empezar la partida
Descripción del requerimiento:	Una vez seleccionados los criterios de la partida, como colores y cantidad de jugadores, se dará inicio a la partida por medio de un botón.
Prioridad del requerimiento:	Alta

1.4.4. NextTurn

Identificación del requerimiento:	RF04
--	------

Nombre del Requerimiento:	NextTurn
Características:	Establecer cuando termina el turno de un jugador para que el siguiente realice su tiro.
Descripción del requerimiento:	Una vez establecido el turno a un jugador, se esperará a que el jugador tome la decisión de saltar turno o mover sus fichas, luego de esto será el siguiente turno.
Prioridad del requerimiento:	Alta

1.4.5. HayGanador

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	HayGanador
Características:	Establecer la validación para declarar a un jugador como ganador de la partida
Descripción del requerimiento:	Si un jugador ha llevado sus cuatro fichas al cielo, ya sea recorriendo todo el tablero o sacando las fichas por medio de 3 pares consecutivos
Prioridad del requerimiento:	Alta

1.4.6. LanzarDados

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	LanzarDados
Características:	Permitir el tiro de dados para hacer su respectivo movimiento.
Descripción del requerimiento:	Luego de que el jugador pulse los dados, el sistema generará dos números aleatorios. Si el resultado es par permite un nuevo lanzamiento.
Prioridad del requerimiento:	Alta

1.4.7. PuedeMoverFicha

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	PuedeMoverFicha
Características:	Validar si la ficha debe ingresar a las últimas 8 casillas
Descripción del	Si la ficha a la cual se le asigna un valor corresponde con el color,

requerimiento:	entrará a las últimas 8 casillas del juego, de lo contrario se tomará como un puente para continuar su camino dentro del tablero.
Prioridad del requerimiento:	Alta

1.4.8. PuedeMoverCasa

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	PuedeMoverCasa
Características:	
Descripción del requerimiento:	Siempre y cuando las fichas estén en las últimas 8 casillas se verificará si puede mover y llega al cielo y sale del juego, o si, no puede mover, no llega al cielo y sigue en juego
Prioridad del requerimiento:	Alta

1.4.9. PuedeMover

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	PuedeMover
Características:	Establecer si el jugador puede mover ficha
Descripción del requerimiento:	Denegar el movimiento de las fichas si no existe posibilidad del mismo para el jugador en el turno en curso, ejemplo: tiene 3 Fichas en la Cárcel y la 4ta a un número exacto de la última casilla (Cielo).
Prioridad del requerimiento:	Alta

1.4.10. EstablecerEquipoFichas

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	EstablecerEquipoFichas
Características:	Establecer si el jugador puede mover ficha
Descripción del requerimiento:	Denegar el movimiento de las fichas si no existe posibilidad del mismo, el movimiento para el jugador en el turno en curso,

	ejemplo: tiene 3 Fichas en la Cárcel y la 4ta a un número exacto de la última casilla (Cielo).
Prioridad del requerimiento:	Alta

1.4.11. ObtenerEquipoFichas

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	ObtenerEquipoFichas
Características:	Establecer las cuatro fichas correspondientes a cada jugador
Descripción del requerimiento:	Establecer las cuatro fichas correspondientes a cada jugador en cada una de sus casas
Prioridad del requerimiento:	Alta

1.4.12. EstablecerValorDado

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	EstablecerValorDado
Características:	Validar que la imagen del dado coincida con el valor dentro del juego
Descripción del requerimiento:	Validar que la imagen del dado coincida con el valor dentro del juego
Prioridad del requerimiento:	Alta

1.4.13. Mover

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Mover
Características:	Establecer los criterios de movimiento de la ficha
Descripción del requerimiento:	Asignar las variables derivadas del movimiento de la ficha en juego dentro del tablero según su posición relativa y final.
Prioridad del requerimiento:	Alta

1.4.14. Preview

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Preview
Características:	Muestra el posible movimiento de una ficha.
Descripción del requerimiento:	Mostrar el posible movimiento de una ficha.
Prioridad del requerimiento:	Alta

1.4.15. TestPosition

Identificación del requerimiento:	RF15
Nombre del Requerimiento:	TestPosition
Características:	Establece bajo qué condiciones debe ingresar o no al último tramo
Descripción del requerimiento:	Indicar la condición bajo la cual una ficha según su posición debe ingresar a CasillaEtapFinal o seguir el camino.
Prioridad del requerimiento:	Alta

1.4.16. MoverACasa

Identificación del requerimiento:	RF16
Nombre del Requerimiento:	MoverACasa
Características:	Establecer los criterios de movimiento de la ficha
Descripción del requerimiento:	Asignar las variables derivadas del movimiento de la ficha en juego dentro del tablero según su posición relativa y final.
Prioridad del requerimiento:	Alta

1.4.17. NuevaFicha

Identificación del requerimiento:	RF17
--	------

Nombre del Requerimiento:	NuevaFicha
Características:	Eventos al matar una ficha (Enviarlo a la cárcel)
Descripción del requerimiento:	Si la ficha en juego llega a matar a alguna ficha, asimila y realiza el bonus de movimiento, si no, el bonus de movimiento no se realizará, por último, se guarda la nueva ubicación de la ficha dentro del tablero.
Prioridad del requerimiento:	Alta

1.4.18. EnviarCasa (Es Cárcel)

Identificación del requerimiento:	RF18
Nombre del Requerimiento:	EnviarCasa
Características:	Validar espacios en casa
Descripción del requerimiento:	Validar si hay un espacio dentro de la casa, para enviar la ficha a la cárcel según a qué jugador este asociada la ficha
Prioridad del requerimiento:	Alta

1.4.19. PuestoLibreCasa

Identificación del requerimiento:	RF19
Nombre del Requerimiento:	PuestoLibreCasa
Características:	Ejecutar la acción de enviar la ficha a la cárcel
Descripción del requerimiento:	Al realizar la validación se moverá la ficha a la casa de su equipo correspondiente ajustándola a uno de los puestos libres teniendo en cuenta un orden visual
Prioridad del requerimiento:	Alta

1.4.20. FixPos

Identificación del requerimiento:	RF20
--	------

Nombre del Requerimiento:	FixPos
Características:	Ajusta la posición de las fichas dentro del tablero
Descripción del requerimiento:	Declaración de la condición para que las fichas se mantengan dentro de la escuadra del tablero, realizado para cada color en los ejes X e Y
Prioridad del requerimiento:	Alta

1.4.21. EsSeguro

Identificación del requerimiento:	RF21
Nombre del Requerimiento:	EsSeguro
Características:	La ficha en seguro no podrá ser enviada a la cárcel.
Descripción del requerimiento:	Cuando una ficha se encuentra en una casilla segura, no podrá ser eliminada
Prioridad del requerimiento:	Alta

1.4.22. FichasEn

Identificación del requerimiento:	RF22
Nombre del Requerimiento:	FichasEn
Características:	Permitir más de una ficha del mismo equipo en la misma casilla
Descripción del requerimiento:	Al permitir una ficha del mismo equipo en una casilla, se evitarán errores al momento de mover la ficha
Prioridad del requerimiento:	Alta

1.4.23. EsPuede

Identificación del requerimiento:	RF23
Nombre del Requerimiento:	EsPuede
Características:	Verificar si la ficha puede moverse cuando hay puente
Descripción del requerimiento:	El algoritmo verifica que no haya puente en el recorrido de la ficha, si la ficha quiere mover y hay puente no se moverá

Prioridad del requerimiento:	Alta
-------------------------------------	------

1.4.24. EsPuentesCasa

Identificación del requerimiento:	RF24
Nombre del Requerimiento:	EsPuentesCasa
Características:	Verificar si hay un puente en casa de cualquier jugador
Descripción del requerimiento:	El algoritmo verifica que no haya puente en el recorrido de la ficha, si la ficha quiere mover y hay puente no se moverá
Prioridad del requerimiento:	Alta

1.4.25. Log

Identificación del requerimiento:	RF25
Nombre del Requerimiento:	Log
Características:	Cuadro que mostrara los movimientos realizados en el juego
Descripción del requerimiento:	Mostrará los mensajes creados en la clase GUI, para los movimientos realizados de cada jugador en cada turno.
Prioridad del requerimiento:	Alta

1.4.26. Resaltador

Identificación del requerimiento:	RF26
Nombre del Requerimiento:	Resaltador
Características:	Resaltar el jugador que tiene el turno activo
Descripción del requerimiento:	Una figura resaltará el turno del jugador para que este sea más ameno y cómodo visualmente. Este se posará sobre la casa del jugador por medio de coordenadas.
Prioridad del requerimiento:	Alta

1.4.27. CasillaEtapaFinal

Identificación del requerimiento:	RF27
Nombre del Requerimiento:	CasillaEtapaFinal
Características:	Asignación de las últimas 32 casillas del recorrido del juego
Descripción del requerimiento:	Se establecen las coordenadas de las últimas 32 casillas (carril final de cada jugador)
Prioridad del requerimiento:	Alta

1.4.28. WinColores

Identificación del requerimiento:	RF28
Nombre del Requerimiento:	WinColores
Características:	Establecer las últimas ocho casillas para cada jugador antes de llegar al cielo.
Descripción del requerimiento:	Se clasifican las últimas ocho casillas para cada jugador, estas serán designadas por medio de coordenadas.
Prioridad del requerimiento:	Alta

1.4.29. Casillas

Identificación del requerimiento:	RF29
Nombre del Requerimiento:	Casillas
Características:	Asignación de las casillas generales.
Descripción del requerimiento:	Se asignarán por medio de coordenadas las casillas generales de la partida. Las casillas generales son todas aquellas que no cuentan como seguro, cárcel o las últimas ocho casillas previas al cielo.
Prioridad del requerimiento:	Alta

1.4.30. SeleccionarJugadores

Identificación del requerimiento:	RF30
Nombre del Requerimiento:	SeleccionarJugadores
Características:	Seleccionar jugadores de la partida
Descripción del requerimiento:	Se podrá escoger cuántos jugadores se desean en la partida. El mínimo es de 2 y el máximo es de jugadores es de 4.
Prioridad del requerimiento:	Alta

1.4.31. HabilitarColoresFicha

Identificación del requerimiento:	RF31
Nombre del Requerimiento:	HabilitarColoresFicha
Características:	Seleccionar los colores de los jugadores
Descripción del requerimiento:	Ayuda para ver los colores y nombres de las fichas a la hora de seleccionar los jugadores deseados para jugar
Prioridad del requerimiento:	Alta

1.4.32. SaltarTurno

Identificación del requerimiento:	RF32
Nombre del Requerimiento:	SaltarTurno
Características:	Saltar el turno del jugador
Descripción del requerimiento:	El jugador podrá saltar el turno si así lo desea
Prioridad del requerimiento:	Alta

1.4.33. BotonDado

Identificación del requerimiento:	RF33
Nombre del	BotonDado

Requerimiento:	
Características:	Permitir que el usuario pueda escoger que dado desea tomar
Descripción del requerimiento:	Cuando el usuario lance los dados, podrá seleccionar el dado con su respectivo valor para mover la ficha a elegir.
Prioridad del requerimiento:	Alta

1.4.34. DiceGrafth

Identificación del requerimiento:	RF34
Nombre del Requerimiento:	DiceGrafth
Características:	Mostrar los dados en pantalla
Descripción del requerimiento:	Permitir observar los dados y sus respectivos valores
Prioridad del requerimiento:	Alta

1.4.35. GetCoin

Identificación del requerimiento:	RF35
Nombre del Requerimiento:	GetCoin
Características:	Permitir seleccionar la ficha a jugar
Descripción del requerimiento:	Permite obtener la posición de la ficha que se desea jugar
Prioridad del requerimiento:	Alta

1.4.36. PreviewCoin

Identificación del requerimiento:	RF36
Nombre del Requerimiento:	PreviewCoin
Características:	Crear copia de la ficha a mover
Descripción del requerimiento:	Se mostrará un “Fantasma” de la ficha a mover a su posible posición final, según el valor generado en los dados

Prioridad del requerimiento:	Alta
-------------------------------------	------

1.4.37. HidePreviewCoin

Identificación del requerimiento:	RF37
Nombre del Requerimiento:	HidePreviewCoin
Características:	Esconder el fantasma
Descripción del requerimiento:	Esconder el fantasma luego de seleccionar
Prioridad del requerimiento:	Alta

1.4.38. CoinClick

Identificación del requerimiento:	RF38
Nombre del Requerimiento:	CoinClick
Características:	Accionar la ficha cuando se da click sobre ella
Descripción del requerimiento:	Al dar click sobre la ficha, esta se moverá o no por el tablero
Prioridad del requerimiento:	Alta

1.4.39. PasaTurno_Tick

Identificación del requerimiento:	RF39
Nombre del Requerimiento:	PasaTurno_Tick
Características:	Establecer cuando termina el juego
Descripción del requerimiento:	Este método se encargará de establecer cuando termina el juego mostrando al ganador en consola.
Prioridad del requerimiento:	Alta

1.4.40. Ficha

Identificación del requerimiento:	RF40
Nombre del	Ficha

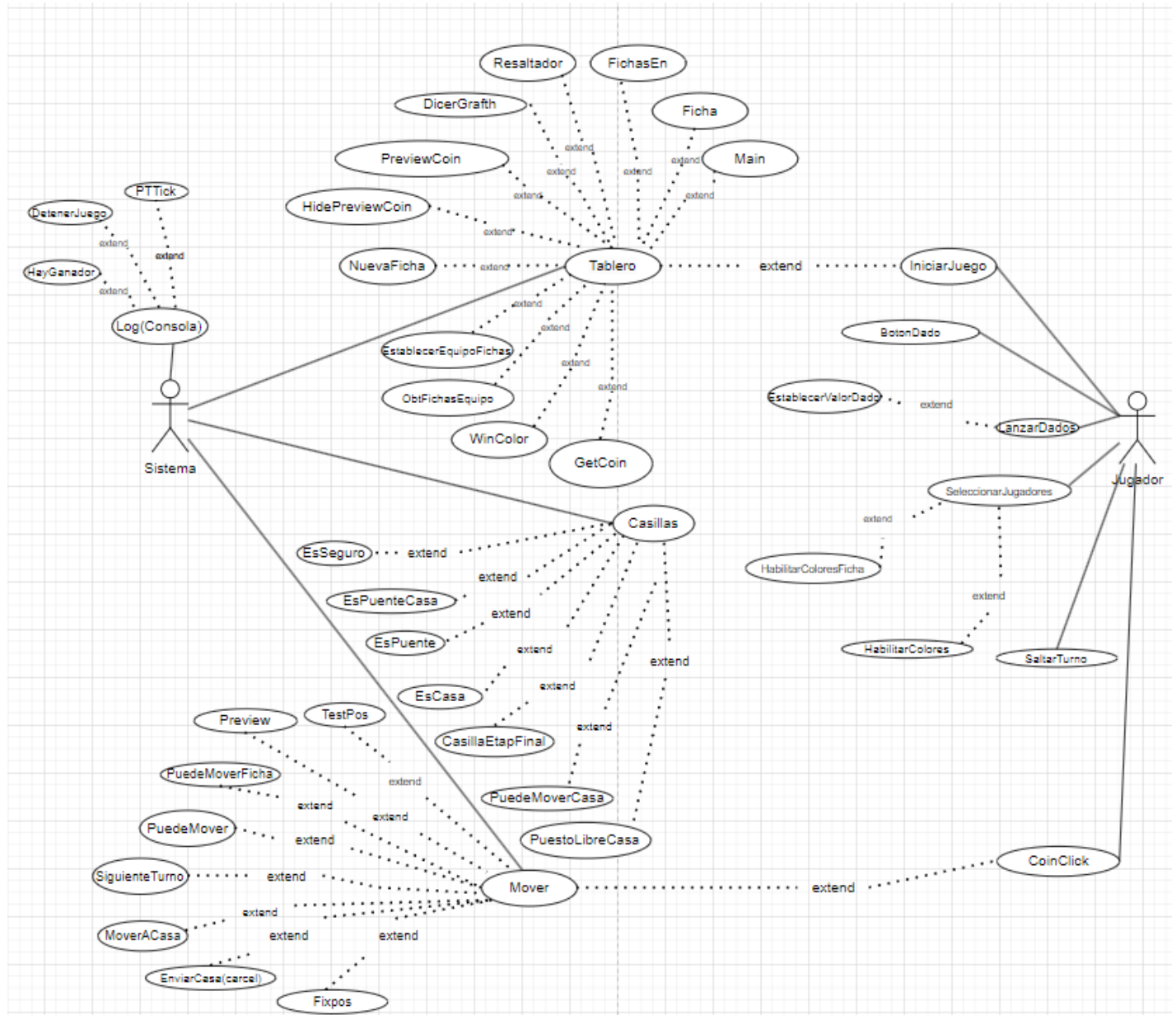
Requerimiento:	
Características:	Establece características básicas de las fichas
Descripción del requerimiento:	Establece las imágenes de las fichas., posicionamiento inicial y asignación a casa según su color
Prioridad del requerimiento:	Alta

1.4.41. Main

Identificación del requerimiento:	RF41
Nombre del Requerimiento:	Main
Características:	Ejecutar correctamente el FORMS1 (MAIN)
Descripción del requerimiento:	Mostrar correctamente toda la interfaz gráfica creada en el Forms1, el llamado al método Main
Prioridad del requerimiento:	Alta

2. Diagrama de Casos de Uso

2.1.1. Diagrama



3. Especificación de Casos de Uso

3.1. Formato Especificación Caso de Uso 1 - Tablero

Proyecto	Parchís en C#			Fecha	8-06-2023	
Código Caso de Uso	RF01	Nombre Caso de Uso	Tablero			
Propósito: Tener el espacio para mostrar todos los elementos del juego						
Alcance: Aplica a todos los jugadores del juego Parchidos						
ACTORES		SIMPLE		MEDIO		COMPLEJO
Jugador						X
Sistema						X
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA		ALTA X
CASOS DE USO ASOCIADOS						
Incluye a: FichaEn, Resaltador, GrafDado, Fantasmon, EsconderFantasmon, NuevaFicha						
Extiende de: Sistema						
Extendido por: Jugador - IniciarJuego						
Precondiciones: El sistema debe tener establecidos todos los ítems que interactúan directamente con el tablero hacía el jugador, como fichas, dados etc.						
Postcondiciones: El jugador ha podido observar todos los items que interactuan con el tablero.						
DATOS DE ENTRADA						
NOMBRE	TIPO		VALIDACIONES			
Dados	Evento		Mostrar los dados			
Ficha	Evento		Establecer las fichas en cada casa			
TurnoActivo	Evento		Mostrar el turno en juego			
Fantasma	Evento		Mostrar la posición relativa de la ficha seleccionada a mover			
DATOS DE SALIDA						
NOMBRE	TIPO		VALIDACIONES			
Dados	Evento		Se muestran los dados con su respectivo valor al jugador			
Ficha	Evento		Se establecen las fichas en las casas correspondientes según su color			
TurnoActivo	Evento		Se muestra correctamente el turno que puede tirar sus dados			
Fantasma	Evento		Se muestra la ficha fantasma al jugador antes de mover su ficha real			
FLUJO DE TRABAJO NORMAL						
PASO	USUARIO		SISTEMA			
1	El jugador ve correctamente el valor de sus dados		El sistema arroja el valor correctamente de manera aleatoria			
2	El jugador ve sus fichas del color de su propia casa correctamente		El sistema asignará las fichas a sus casas correspondientes			
3	El jugador ve de forma clara como se resalta el jugador activo		El sistema resalta de forma clara el jugador activo			
4	El jugador aprecia correctamente la ficha fantasma		El sistema muestra correctamente la ficha fantasma			
FLUJO DE TRABAJO ALTERNATIVO 1						
PASO	USUARIO		SISTEMA			
1	El jugador no puede ver sus dados		El sistema no muestra de manera correcta los dados			
2	El jugador no ve fichas o las ve en la casa errónea		El sistema asigna mal las fichas			
3	El jugador no ve cuando es su turno al lanzar los dados y/o mover las fichas		El sistema no muestra correctamente el turno del jugador			
4	El jugador no aprecia la ficha fantasma		El sistema no muestra correctamente la ficha fantasma			

Requerimientos Especiales: Deben estar previamente cargadas las imágenes que se deben mostrar en el tablero		
Riesgos: El tablero no funciona o no es posible para el jugador jugar al juego		
Criterios de Aceptación: El jugador puede jugar correctamente al juego		
Pantalla Propuesta:		
FIRMAS		
Jose Aguirre	Politécnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.2. Formato Especificación Caso de Uso 2 - DetenerJuego

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF02		Nombre Caso de Uso		DetenerJuego			
Propósito: Mostrar en pantalla que el juego se ha detenido cuando un jugador ya no tiene fichas en juego								
Alcance: Aplica a todos los jugadores del Parchídos								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema					X			
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA		X	
					ALTA			
CASOS DE USO ASOCIADOS								
Incluye a: Log (Consola) Extiende de: Sistema Extendido por: Sistema								
Precondiciones: El sistema debe estar configurado para saber cuándo emplear el DetenerJuego y mostrarlo en el Log								
Postcondiciones: El jugador ha podido observar un mensaje en el Log anunciando que el juego ha sido detenido								
DATOS DE ENTRADA								
NOMBRE		TIPO		VALIDACIONES				
Ficha		Evento		Todas las Fichas del mismo color llegan al Cielo				
DATOS DE SALIDA								
NOMBRE		TIPO		VALIDACIONES				
Ficha		Evento		El jugador observa las fichas en el cielo, por lo tanto, se mostrará el DetenerJuego				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador aprecia como se detiene el juego cuando un equipo de fichas ha llegado al cielo	El sistema detiene el juego al detectar que un equipo de fichas ha ganado
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El jugador observa que el juego no se detiene cuando ya ha ganado un grupo de fichas	El sistema no detiene el juego cuando ya ha ganado un grupo de fichas
Requerimientos Especiales: Debe estar correctamente codificado DetenerJuego para que funcione como se espera		
Riesgos: El DetenerJuego no funciona por lo tanto no habrá un ganador		
Criterios de Aceptación: El juego de detiene correctamente		
Pantalla Propuesta:		
FIRMAS		
Jose Aguirre	Politécnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.3. Formato Especificación Caso de Uso 3 - IniciarJuego

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF03		Nombre Caso de Uso		IniciarJuego			
Propósito: Permitir iniciar el juego cuando se desee siempre y cuando se llenen parámetros como cantidad de jugadores (mínimo 2)								
Alcance: Aplica a todos los jugadores de Parchídos								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Jugador							X	
Sistema							X	
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA		ALTA	X
CASOS DE USO ASOCIADOS								
Incluye a: Tablero								
Extiende de: Jugador								
Extendido por: Sistema - Tablero								

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Precondiciones: Se muestra correctamente el botón de Iniciar Juego y este efectúa de manera correcta el inicio del Tablero		
Postcondiciones: Permitir que el jugador pueda seleccionar la cantidad de jugadores antes de iniciar la partida		
DATOS DE ENTRADA		
NOMBRE	TIPO	VALIDACIONES
Click Btn Inicio	Evento	Al ejecutar el botón, se dará inicio a la partida
Cant. Jugadores	Evento	Se establece la Cantidad de jugadores en partida
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
Mensaje en pantalla	Evento	Se mostrará un mensaje para confirmar el inicio del juego
Cant. Jugadores	Evento	Se activarán únicamente turnos de juego a la cantidad de jugadores indicada previamente
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador ha iniciado el juego	Ejecuta de forma normal el juego
2	El jugador establece la cantidad de jugadores en partida	Se muestra únicamente los jugadores previamente establecidos para la partida en curso
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El jugador no puede iniciar el juego	No se ejecuta el inicio del juego al dar Click en el btn Iniciar juego
2	No se puede establecer la cantidad de jugadores para la partida	Muestra más de los jugadores solicitados o de plano no activa a ningún jugador para la partida en curso
Requerimientos Especiales: Deben mostrarse correctamente los botones de iniciar juego y selección de jugadores		
Riesgos: Aun así seleccionando y aplicando los botones, estos no funcionan de la manera esperada		
Criterios de Aceptación: Los botones funcionan correctamente		
Pantalla Propuesta:		
FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

3.4. Formato Especificación Caso de Uso 4 - NextTurn

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF04	Nombre Caso de Uso	NextTurn				
Propósito: Permitir que el sistema realice el cambio de jugador cuando este finalice su turno							
Alcance: Para cada uno de los jugadores activos en partida							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Mover Extiende de: Mover Extendido por: Sistema							
Precondiciones: Establecer el momento en que un jugador finaliza con su turno, para pasar al siguiente, según las manecillas del reloj, teniendo al menos 2 jugadores activos							
Postcondiciones: se pasa de turno correctamente, en orden de las manecillas del reloj							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
Siguiente Turno	Evento		Se entenderá como finalizado un turno, cuando un jugador no pueda mover ninguna de sus fichas o este haya decidido saltar turno o en su defecto haya corrido con sus fichas los valores dados por su lanzamiento de dados.				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
Siguiente Turno	Evento		Se dispondrá a asignar al siguiente jugador, el turno activo y se dejará a espera del cumplimiento de la regla para ser ejecutada nuevamente.				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	Un jugador finaliza de mover sus fichas o salta su turno			Se establece el turno al siguiente jugador siempre que se cumpla: Saltar turno, se han movido las fichas del jugador activo, no se puede mover ninguna ficha.			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El jugador finaliza de mover sus fichas, pero el turno no cambia al siguiente jugador			El sistema no es capaz de reconocer la finalización de turno, por lo cual queda en el bucle de no finalizar el turno			

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



		activo
Requerimientos Especiales: Establecer los parámetros de finalización de turno: Todas las fichas movidas según la cantidad arrojada por los dados, No se puede mover ninguna ficha por el jugador, el jugador ha decidido saltar su turno.		
Riesgos: El sistema no es capaz de ejecutar las condiciones de cambio de turno		
Criterios de Aceptación: Se realiza el siguiente turno correctamente		
Pantalla Propuesta:		
FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

3.5. Formato Especificación Caso de Uso 5 - HayGanador

Proyecto	Parchís en C#	Fecha	8-06-2023
Código Caso de Uso	RF05	Nombre Caso de Uso	HayGanador
Propósito: Definir al ganador de la partida			
Alcance: Para cada uno de los jugadores de la partida			
ACTORES	SIMPLE	MEDIO	COMPLEJO
Jugador		X	
Sistema		X	
COMPLEJIDAD DEL CASO DE USO	BAJA	MEDIA	ALTA
		X	
CASOS DE USO ASOCIADOS			
Incluye a: Consola, Jugador Extiende de: Log Extendido por: Sistema			
Precondiciones: Tener las 4 fichas del mismo color en la casilla final (Cielo)			
Postcondiciones: Arrojar el mensaje con el color del jugador ganador			
DATOS DE ENTRADA			
NOMBRE	TIPO	VALIDACIONES	
HayGanador	Evento	Se entenderá como ganador al jugador que lleve sus 4 fichas hasta la última casilla del tablero (Cielo)	

DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
HayGanador	Evento	Se mostrará un mensaje en el Log, con el color del jugador ganador
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El usuario lleva sus 4 fichas por todas las casillas del tablero	Establece al ganador según el criterio previamente establecido para declarar un ganador
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	No puede llevar sus fichas hasta la casilla del Cielo, no se le permite el movimiento o desplazamiento de 1 o más fichas.	Aun teniendo las condiciones cumplidas por parte del jugador, el sistema no lo declara como ganador y no finaliza la partida.
Requerimientos Especiales: N/A		
Riesgos: No poder declarar un ganador		
Criterios de Aceptación: Se declara el ganador de la partida de forma correcta		
Pantalla Propuesta:		
FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

3.6. Formato Especificación Caso de Uso 6 - LanzarDados

Proyecto	Parchís en C#			Fecha	8-06-2023
Código Caso de Uso	RF06	Nombre Caso de Uso	LanzarDados		
Propósito: Permitir que los jugadores licen el lanzamiento de dados para que se les asigne cuantas casillas pueden mover sus fichas					
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4					
ACTORES		SIMPLE	MEDIO	COMPLEJO	
Jugador			X		
Sistema			X		

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Establecer valor dado Extiende de: Tirara Dados Extendido por: Usuario - Sistema							
Precondiciones: Tener asignado un botón para realizar la ejecución del lanzamiento, disponer de los gráficos de los dados dentro del tablero para que el usuario vea el resultado en pantalla.							
Postcondiciones: Que el usuario vea el numero random asignado visualmente en los dados, y pueda proceder a mover sus fichas							
DATOS DE ENTRADA							
NOMBRE		TIPO		VALIDACIONES			
Tirar Dados		Evento		Dar click al botón de tirar dados y que este arroje un valor random para que el jugador proceda a mover sus fichas según el valor asignado.			
DATOS DE SALIDA							
NOMBRE		TIPO		VALIDACIONES			
Tirar Dados		Evento		Se muestra correctamente mediante los gráficos de los dados en el tablero, el valor random generado por el sistema			
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	Puede lanzar los dados por medio del botón designado para tal fin			Es capaz de asignar un valor random para cada uno de los dados y asociarlo a las imágenes de los dados que serán visibles por el usuario.			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	No es posible lanzar los dados por medio del botón.			No es capaz de generar un valor aleatorio y no muestra el resultado en los gráficos de los dados, o genera un valor aleatorio, pero aun así no lo asocia a las imágenes de los dados, para ser leído el valor por el usuario			
Requerimientos Especiales: Diseño gráfico de los dados y creación del botón lanzar dados							
Riesgos: Que no se ejecute la acción del lanzamiento de dados							
Criterios de Aceptación: se ejecuta la acción solicitada, mostrando los valores visualmente de forma correcta							
Pantalla Propuesta:							
FIRMAS							

Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

3.7. Formato Especificación Caso de Uso 7 - PuedeMoverFicha

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF07	Nombre Caso de Uso	PuedeMoverFicha				
Propósito: Permitir que el jugador pueda mover sus fichas, siempre que estas se encuentren fuera de casa o sea en juego.							
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Jugador				X			
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Mover Extiende de: Mover Extendido por: Sistema							
Precondiciones: Disponer de al menos una ficha en juego que pueda moverse dentro del tablero, por el valor dado por los dados							
Postcondiciones: Se realiza el movimiento, se evidencia gráficamente y se realiza la finalización del turno activo							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
PuedeMoverFicha	Evento		Se dispone de fichas dentro del tablero para realizar un movimiento				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
PuedeMoverFicha	Evento		Se muestra correctamente la posición final de la o las fichas movidas.				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO		SISTEMA				
1	Verifica que ficha dentro del tablero de juego desea mover		Define si la ficha puede moverse las casillas según el valor asignado por los dados, teniendo como criterio adicional, que no esté en casa (carcel), disponga de los espacios				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



		necesarios para su movimiento y esta no este previamente en el Cielo
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario no ve que ficha puede mover	No es capaz de distinguir las características especiales de las casillas dentro del juego, por lo cual permite el movimiento de cualquier ficha dentro del tablero, indistinto del color o si está en el Cielo
Requerimientos Especiales: Disponer de al menos una ficha para realizar el movimiento		
Riesgos: Que no se pueda mover ninguna ficha o por el contrario se pueda mover cualquier ficha indistinta del turno activo		
Criterios de Aceptación: Las fichas se mueven correctamente dentro del tablero y del color del turno activo		
Pantalla Propuesta:		
FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

3.8. Formato Especificación Caso de Uso 8 - PuedeMoverCasa

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF08	Nombre Caso de Uso	PuedeMoverCasa				
Propósito: Si las fichas están en las últimas 8 casillas previas al cielo, se verificara si puede mover para llegar al cielo y salir del juego, si la ficha no puede mover, no llega al cielo y sigue en juego Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Casillas Extiende de: Casillas Extendido por: Sistema							
Precondiciones: Es necesario tener una ficha en las 8 últimas casillas previas al cielo para que dicho algoritmo se ejecute							
Postcondiciones: Se realiza el movimiento para verificar si la ficha llega al cielo o no, depende del resultado antes mencionado para saber si la ficha sigue en juego o si se dirige a su casa							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
PuedeMoverCasa	Evento		El valor del dado coincide o no con las casillas que faltan para llegar al cielo, esto le dará movimiento a la ficha y posteriormente se sabrá si la ficha debe ir a casa				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
PuedeMoverCasa	Evento		Dependiendo del resultado, el algoritmo funciona correctamente				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO		SISTEMA				
1	Rectifica que el movimiento de la ficha independientemente del resultado sea correcto		Realiza el movimiento correcto de la ficha según el resultado				
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO		SISTEMA				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



1	La ficha no se mueve a su casa después de haber llegado al cielo	El sistema no ejecuta bien el algoritmo
Requerimientos Especiales: N/A		
Riesgos: Que la ficha no se mueva y el algoritmo no pueda correr debido a la falta de información		
Criterios de Aceptación: Se observa que la ficha siga en juego o vaya a casa correctamente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.9. Formato Especificación Caso de Uso 9 – PuedeMover

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF09		Nombre Caso de Uso		PuedeMover		
Propósito: Validar las condiciones de movimiento de las fichas dentro del tablero							
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4							
ACTORES			SIMPLE		MEDIO		COMPLEJO
Sistema					X		
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	X	ALTA
CASOS DE USO ASOCIADOS							
Incluye a: Mover							
Extiende de: Mover							
Extendido por: Sistema							
Precondiciones: PuedeMoverFicha							
Postcondiciones: Se moverá la ficha dentro del tablero, y en las casillas correspondientes							
DATOS DE ENTRADA							
NOMBRE	TIPO			VALIDACIONES			
PuedeMover	Evento			Si la ficha ya salió de la casa o no está en cárcel, se considera que se puede mover			
DATOS DE SALIDA							

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



NOMBRE		TIPO	VALIDACIONES
PuedeMover		Evento	Se realiza el movimiento de la ficha dentro de las casillas del tablero
FLUJO DE TRABAJO NORMAL			
PASO	USUARIO	SISTEMA	
1	Puede mover alguna de sus fichas siempre que no esté en la cárcel o casa	Valida la posición de las fichas dentro del tablero, para permitir o no su movimiento en las casillas del tablero	
FLUJO DE TRABAJO ALTERNATIVO 1			
PASO	USUARIO	SISTEMA	
1	No se puede mover ninguna ficha	No es capaz de discriminar si una ficha puede o no ser movida dentro del tablero, o las mueve en posiciones aleatorias dentro del tablero.	
Requerimientos Especiales: Disponer de fichas ya en juego (Fuera de Casa – Carcel)			
Riesgos: Que no se muestre de manera adecuada el movimiento de las fichas dentro ddel tablero de juego			
Criterios de Aceptación: Permitir el movimiento fluido de las fichas dentro de las casillas del tablero			
Pantalla Propuesta:			
FIRMAS			
Jose Aguirre		Politecnico	Darmabel Orozco
ANALISTA		CLIENTE	REVISOR

3.10. Formato Especificación Caso de Uso 10 – EstablecerEquipoFichas

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF10		Nombre Caso de Uso		EstablecerEquipoFichas			
Propósito: Establecer las cuatro fichas correspondientes a cada jugador en su casa								
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema					X			
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	X	ALTA	

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



CASOS DE USO ASOCIADOS		
Incluye a: Tablero		
Extiende de: Tablero		
Extendido por: Sistema		
Precondiciones: Haber asignado correctamente a qué casa o posición pertenecerán ciertas fichas		
Postcondiciones: Se establecerán las fichas correctamente en las casas de cada jugador		
DATOS DE ENTRADA		
NOMBRE	TIPO	VALIDACIONES
Fichas	Evento	Se obtiene la información de las fichas para que el sistema pueda asignarlas a sus casas correspondientes
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
Fichas	Evento	El sistema asigna correctamente las fichas a sus casas
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El usuario ve que tiene todas sus fichas del mismo equipo en su casa correspondiente	El sistema logra de manera exitosa asignar correctamente las fichas en las casas de los jugadores
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario ve que sus fichas o las de los demás jugadores no están completas o incluso no están en las casas a las que corresponden	El sistema recibe de forma incorrecta las posiciones de las fichas para asignarlas en el tablero
Requerimientos Especiales: Establecer correctamente los datos de las fichas para no tener errores		
Riesgos: Las fichas están incompletas o son mostradas incorrectamente en casas ajenas		
Criterios de Aceptación: El juego es cómodo visualmente al ver que las fichas hacen parte de su propio equipo		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.11. Formato Especificación Caso de Uso 11 - ObtFichasEquipo

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF11	Nombre Caso de Uso	ObtFichasEquipo				
Propósito: Posicionar u obtener las fichas del equipo para cada jugador							
Alcance: Se realiza para cada uno de los jugadores.							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							
Extiende de: Tablero							
Extendido por: Sistema							
Precondiciones: Tener cuatro fichas para cada jugador							
Postcondiciones: Establecer cada una de las fichas del jugador en su respectiva casa							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
Fichas	Evento		Se obtienen las fichas de cada jugador para poder asignarlas a cada casa				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
Fichas	Evento		Se asignan correctamente las fichas				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario aprecia que sus fichas están colocadas en su propia casa al iniciar el juego			El sistema al iniciar el juego asigna correctamente las fichas en las casas correspondientes			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El usuario ve que sus fichas están en otra casa			El sistema asigna las fichas de los jugadores en casas no correspondientes			
Requerimientos Especiales: N/A							

Riesgos: No asignar las fichas correctamente		
Criterios de Aceptación: Definir las fichas correctamente para que no haya problemas en el juego		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politenico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.12. Formato Especificación Caso de Uso 12 - EstablecerValorDado

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF12		Nombre Caso de Uso		EstablecerValorDado		
Propósito: Validar que el valor del dado que se muestra en el tablero coincida con el valor random asignado por el juego							
Alcance: Se realiza para todos los jugadores a la vez							
ACTORES			SIMPLE		MEDIO		COMPLEJO
Jugador					X		
Sistema							X
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	X	ALTA
CASOS DE USO ASOCIADOS							
Incluye a: TirarDados Extiende de: TirarDados Extendido por: Tablero, Sistema							
Precondiciones: Establecer correctamente el valor del dado dentro del juego para que se pueda jugar de forma correcta							
Postcondiciones: Se establece el valor del dado correctamente							
DATOS DE ENTRADA							
NOMBRE	TIPO			VALIDACIONES			
EstablecerValorDado	Evento			Se asigna el valor del dado correctamente, este debe coincidir con la imagen del dado que se muestra en pantalla			
DATOS DE SALIDA							
NOMBRE	TIPO			VALIDACIONES			
EstablecerValorDad	Evento			Se aprecia la correcta asignación del valor de los dados			

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador aprecia que el valor de los dados ha sido establecido correctamente al ver que su ficha mueve lo que indica el valor del dado	El sistema asigna correctamente el valor de los dados en el juego
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El jugador ve que no se han asignado los valores del dado correctamente, ya que al mover una ficha se mueve un valor distinto al dado	El sistema es incapaz de asignar correctamente los dados
Requerimientos Especiales:		
Riesgos:		
Criterios de Aceptación:		
Pantalla Propuesta:		
FIRMAS		
<Nombre>	<Nombre>	<Nombre>
ANALISTA	CLIENTE	REVISOR

3.13. Formato Especificación Caso de Uso 13 - Mover

Proyecto	Parchís en C#			Fecha	8-06-2023	
Código Caso de Uso	RF13	Nombre Caso de Uso	Mover			
Propósito: Establecer los criterios de movimiento de la ficha						
Alcance: Se realiza para todos los jugadores del Parchís						
ACTORES		SIMPLE	MEDIO	COMPLEJO		
Jugador				X		
Sistema				X		
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA		ALTA X
CASOS DE USO ASOCIADOS						

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Incluye a: TestPos, Fantasma, PuedeMoverFicha, PuedeMover, SiguienteTurno, MoverACasa, EnviarACasa(cárcel), ArreglarPosicion Extiende de: Sistema Extendido por: FichaClick		
Precondiciones: El sistema debe tener establecidos todos los ítems que interactúan directamente con el movimiento, como: Fichas, Dados, etc.		
Postcondiciones: El jugador ha podido observar que todos los ítems interactúan de forma correcta		
DATOS DE ENTRADA		
NOMBRE	TIPO	VALIDACIONES
Fichas	Evento	Las fichas tienen un movimiento correcto
Dados	Evento	Los dados funcionan correctamente al mover por un valor
Coordenadas	Evento	Las coordenadas realizan el correcto movimiento de las fichas en el tablero
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
Fichas	Eventos	Las fichas se mueven correctamente
Dados	Eventos	El valor que muestran los dados es el valor que se mueve la ficha
Coordenadas	Eventos	Se muestra el correcto movimiento de las fichas en el juego
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	Observa el correcto movimiento de las fichas	El sistema realiza el correcto movimiento de las fichas
2	Los dados arrojan valores lógicos	El sistema arroja valores lógicos
3	Observa el movimiento de forma correcta por el tablero	El sistema mueve las fichas por el tablero de forma lógica
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	Observa que las fichas no se mueven correctamente	El sistema no realiza el correcto movimiento de las fichas
2	Los dados no responden o no arrojan valores	El sistema no arroja correctamente los valores de de los dados
3	El movimiento de las coordenadas no es congruente	El sistema no mueve las fichas correctamente por el tablero
Requerimientos Especiales: Debe estar previamente codificado el método mover para hacer sintonía con los demás ítems.		
Riesgos: El movimiento no funciona como se esperaba		
Criterios de Aceptación: El jugador puede mover correctamente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politécnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.14. Formato Especificación Caso de Uso 14 - Fantasma

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF14	Nombre Caso de Uso	Preview				
Propósito: Permitir que los jugadores puedan observar el posible movimiento de una ficha							
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Jugador				X			
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Mover Extiende de: Mover Extendido por: Sistema, CoinClick, Jugador							
Precondiciones: El jugador debe haber lanzado sus dados para seleccionar la ficha a mover, luego de esto al posar el mouse en la ficha, podrá ver el posible movimiento							
Postcondiciones: El jugador podrá ver el posible movimiento							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
Preview	Evento		Se recibe la ficha para mostrar el posible movimiento				
Ficha	Evento		Se recibe la ficha para poder seleccionarla y mostrar su posible movimiento				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
Preview	Evento		El posible movimiento de la ficha se muestra correctamente				
Ficha	Evento		La ficha seleccionada muestra su posible movimiento correctamente				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO		SISTEMA				
1	El usuario nota el posible movimiento al posar el mouse sobre la ficha		El sistema muestra el correcto movimiento de la ficha al posar el mouse sobre la misma				
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO		SISTEMA				
1	El usuario al posar el mouse sobre la ficha no aprecia o no ve correctamente el posible movimiento		El sistema muestra o no el posible movimiento de la ficha correctamente				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Requerimientos Especiales: Programar correctamente el uso de Preview		
Riesgos: Muestra la ficha previa al movimiento en lugares o zonas incorrectas		
Criterios de Aceptación: Tener todas las fichas para al momento de seleccionar para moverla mostrar correctamente el movimiento		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.15. Formato Especificación Caso de Uso 15 - TestPos

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF15	Nombre Caso de Uso	TestPos				
Propósito: Establecer ciertas condiciones para saber si una ficha sigue su camino o entra a las últimas 8 casillas del juego							
Alcance: Para todos los jugadores del Parchís							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				X			
				X			
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	X	ALTA
CASOS DE USO ASOCIADOS							
Incluye a: Casillas Extiende de: Casillas Extendido por: Sistema							
Precondiciones: Luego de que una ficha haya dado la vuelta y este a punto de llegar a las últimas 8 casillas, se validará si el color de la ficha corresponde con las coordenadas asignadas para poder seguir.							
Postcondiciones: La ficha fue verificada y gracias a las coordenadas puede pasar a las últimas ocho casillas previas al cielo							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
TestPos	Evento		La ficha se valida exitosamente al verificar el color de la ficha con las coordenadas para seguir su camino hacia el cielo				

DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
TestPos	Evento	La ficha ha podido ingresar a las ocho últimas casillas previas al cielo
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador observa que su ficha entra en las 8 últimas ocho casillas correspondientes a su equipo	El sistema funciona correctamente al comparar el color con las coordenadas para hacer un buen procedimiento
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario observa que luego de dar una vuelta su ficha ingresa a las últimas 8 casillas de otro equipo	El sistema tiene mal codificado las coordenadas para el color de las fichas, por lo tanto, no funciona como debería
Requerimientos Especiales: Establecer las coordenadas y asignación de las mismas hacía los colores correspondientes		
Riesgos: Las fichas no se mueven correctamente		
Criterios de Aceptación: La ficha entra en las casillas previas al cielo siempre y cuando sea parte de su color		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politécnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.16. Formato Especificación Caso de Uso 16 – MoverACasa

Proyecto	Parchís en C#			Fecha	8-06-2023
Código Caso de Uso	RF16	Nombre Caso de Uso	MoverACasa		
Propósito: Establecer los criterios de movimiento de la ficha					
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4					
ACTORES		SIMPLE	MEDIO	COMPLEJO	
Sistema			X		

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Mover Extiende de: Mover Extendido por: Sistema							
Precondiciones: Haber asignado correctamente las variables derivadas del movimiento de la ficha en juego dentro del tablero según su posición relativa							
Postcondiciones: La ficha tiene asignado correctamente las variables de movimiento							
DATOS DE ENTRADA							
NOMBRE		TIPO		VALIDACIONES			
MoverACasa		Evento		La ficha ha recibido correctamente las variables derivadas del movimiento de la ficha para la posición			
DATOS DE SALIDA							
NOMBRE		TIPO		VALIDACIONES			
MoverACasa		Evento		La ficha ha recibido correctamente las variables de movimiento para su posición relativa y final, todo esto para poder desplazar una ficha a Casa			
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO		SISTEMA				
1	El usuario observa el correcto movimiento		El sistema se apoya el método para poder ejecutar el movimiento correctamente				
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO		SISTEMA				
1	El usuario detecta el mal movimiento		El sistema no realiza correctamente lo mencionado anteriormente				
Requerimientos Especiales: Asignar correctamente las variables derivadas del movimiento de la ficha							
Riesgos: Que el método no reciba correctamente las variables derivadas del movimiento de la ficha							
Criterios de Aceptación: El método cumple su funcionalidad correctamente en el juego							
Pantalla Propuesta:							
FIRMAS							
Juan Jaraba		Politecnico			Darmabel		
ANALISTA		CLIENTE			REVISOR		

3.17. Formato Especificación Caso de Uso 17 - NuevaFicha

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF17	Nombre Caso de Uso	NuevaFicha				
Propósito: Permitir cierto movimiento al matar una ficha enemiga, este movimiento no se podrá efectuar si hay un puente mas adelante							
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Jugador				X			
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							
Extiende de: Tablero							
Extendido por: Sistema							
Precondiciones: Realizar el movimiento acertado para matar una ficha							
Postcondiciones: Al matar la ficha se otorgarán 20 casillas de movimiento							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
NuevaFicha	Evento		Recibe si una ficha ha sido eliminada o no				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
NuevaFicha	Evento		Otorga 20 casillas de movimiento por haber eliminado a una ficha enemiga				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario verifica que se realice el bonus de movimiento cuando es posible			El sistema realiza o no el movimiento especial si se cumplen las condiciones			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El movimiento especial no pasa			El sistema no realiza el movimiento especial			

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Requerimientos Especiales: Haber asignado correctamente cuando dar el movimiento especial		
Riesgos: El movimiento de 20 casillas no se realiza o no funciona		
Criterios de Aceptación: El movimiento ha sido realizado y guardado correctamente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.18. Formato Especificación Caso de Uso 18 - EnviarACasa

Proyecto	Parchís en C#			Fecha	8-06-2023	
Código Caso de Uso	RF18	Nombre Caso de Uso	EnviarACasa			
Propósito: Validar si en la casa del jugador hay espacio para asignar una ficha						
Alcance: Se realiza para cada uno de los jugadores, hasta un número máximo de 4						
ACTORES		SIMPLE		MEDIO		COMPLEJO
Sistema				X		
		BAJA		MEDIA	X	ALTA
CASOS DE USO ASOCIADOS						
Incluye a:Mover Extiende de:Mover Extendido por:Sistema						
Precondiciones: Tener una ficha a punto de eliminar o eliminada						

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Postcondiciones: Al haber una ficha de un equipo eliminada se confirmará si hay una posición libre en casa para asignarla		
DATOS DE ENTRADA		
NOMBRE	TIPO	VALIDACIONES
EnviarCasa	Evento	Si la ficha muere se ejecuta EnviarCasa para validar la casa del jugador
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
EnviarCasa	Evento	La ficha ha sido asignada a la cárcel correctamente luego de verificar si existe un espacio en la cárcel del jugador
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El usuario observa el correcto funcionamiento de la cárcel	El sistema asigna correctamente la ficha a la cárcel luego de verificar si hay un espacio en la casa del jugador
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario observa que la ficha no ha sido enviada a la cárcel	El sistema no es capaz de asignar correctamente la ficha
Requerimientos Especiales: N/A		
Riesgos: Que la ficha no sea asignada a la cárcel correctamente, poniendo en riesgo la mecánica del juego		
Criterios de Aceptación: La ficha es correctamente asignada en la cárcel al momento de ser eliminada		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politécnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.19. Formato Especificación Caso de Uso 19 - PuestoLibreCasa

Proyecto	Parchís en C#	Fecha	8-06-2023
-----------------	---------------	--------------	-----------

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Código Caso de Uso	RF20	Nombre Caso de Uso	PuestoLibreCasa		
Propósito: Ejecutar la acción de enviar la ficha a la cárcel					
Alcance: Todos los jugadores de Parchidos					
ACTORES		SIMPLE	MEDIO		COMPLEJO
Sistema			X		
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	ALTA
				X	
CASOS DE USO ASOCIADOS					
Incluye a: Casillas Extiende de: Casillas Extendido por: Sistema					
Precondiciones: Se debe eliminar una ficha para que el algoritmo funcione					
Postcondiciones: Luego de que una ficha sea eliminada o comida, el algoritmo asignara la ficha en su casa correspondiente					
DATOS DE ENTRADA					
NOMBRE	TIPO		VALIDACIONES		
Ficha	Evento		Se recibe la información de que la ficha ha sido eliminada		
DATOS DE SALIDA					
NOMBRE	TIPO		VALIDACIONES		
Ficha	Evento		Al realizar la validación se moverá la ficha a la casa de su equipo correspondiente ajustándola a uno de los puestos libres teniendo en cuenta un orden visual		
FLUJO DE TRABAJO NORMAL					
PASO	USUARIO		SISTEMA		
1	El usuario ve que al ser eliminada una ficha se mueve hacia su posición inicial al empezar el juego, o sea la casa.		El sistema hace la ejecución correcta del algoritmo al posicionar la ficha en su casa al ser eliminada		
FLUJO DE TRABAJO ALTERNATIVO 1					
PASO	USUARIO		SISTEMA		
1	El usuario detecta un fallo al ver que la ficha no se mueve hacia su casa al ser eliminada		El sistema no realiza correctamente el movimiento de la ficha hacia su casa luego de ser eliminada		
Requerimientos Especiales: Codificar correctamente el algoritmo para que la ficha pueda ser enviada a la cárcel luego de haber sido eliminada					
Riesgos: La ficha presenta fallos de movimiento, lo que puede arruinar la jugabilidad de los jugadores					

Criterios de Aceptación: Se realiza correctamente el movimiento de la ficha hacia la cárcel cuando la ficha ha sido eliminada		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.20. Formato Especificación Caso de Uso 20- FixPos

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF21	Nombre Caso de Uso	FixPos				
Propósito: Ajustar la posición de las fichas dentro del tablero							
Alcance: Para todos los jugadores de Parchís							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Mover							
Extiende de: Mover							
Extendido por: Sistema							
Precondiciones: Establecer la posición inicial de las fichas de cada jugador en su respectiva casa							
Postcondiciones: Se establecen las posiciones de las fichas correctamente							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
FixPos	Evento		Se establecerá la condición que asigna las fichas dentro de la escuadra del tablero, ya sea en casa, en casillas normales o en las últimas ocho casillas previas al cielo				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
FixPos	Evento		Se muestran las fichas correctamente en el tablero al realizar sus movimientos				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador detecta que sus fichas y las fichas enemigas se muevan de forma correcta por el tablero	El sistema mueve las fichas correctamente por el tablero
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El jugador ve sus fichas fuera de control y sin orden	El sistema no determina correctamente la posición de las fichas en el tablero y presenta errores
Requerimientos Especiales: Establecer correctamente las posiciones de las fichas en el tablero para tener un juego agradable		
Riesgos: El sistema no asigna correctamente el movimiento de las fichas durante el recorrido del tablero		
Criterios de Aceptación: Se realiza la asignación de las casillas de las fichas correctamente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.21. Formato Especificación Caso de Uso 21- EsSeguro

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF21	Nombre Caso de Uso	EsSeguro				
Propósito: Las fichas que se posen sobre una casilla segura no podran ser eliminadas							
Alcance: Para todos los jugadores de Parchís							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema						X	
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Casillas							
Extiende de: Casillas							
Extendido por: Sistema							
Precondiciones: Establecer por medio de coordenadas cuales son las posiciones en donde no se puede matar una ficha							
Postcondiciones: Al establecer correctamente las posiciones, la ficha que se pose en un seguro no es eliminada							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
EsSeguro	Evento		Cuando una ficha se encuentra en una casilla segura, no podrá ser eliminada				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
EsSeguro	Evento		Cuando la ficha se posa sobre una casilla segura no es eliminada				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El jugador aprecia que sus fichas están sanas y salvadas dentro de una casilla segura			El sistema asegura la ficha que se pose en una casilla segura, por lo tanto, la ficha no podrá ser eliminada			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El jugador ve que sus fichas son eliminadas estando en una casilla segura			El sistema no protege correctamente a la ficha estando en una casilla segura			
Requerimientos Especiales: Establecer correctamente las posiciones de las fichas en el tablero para tener un juego agradable							

Riesgos: El sistema no funciona correctamente al ver que las fichas posadas en EsSeguro han sido eliminadas		
Criterios de Aceptación: Se realiza la asignación de las casillas de las fichas correctamente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.22. Formato Especificación Caso de Uso 22- FichasEn

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF22		Nombre Caso de Uso		FichasEn			
Propósito: Permitir más de una ficha del mismo equipo en la misma casilla								
Alcance: Para todos los jugadores de Parchís								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema							X	
Jugador			X					
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA		X	
					ALTA			
CASOS DE USO ASOCIADOS								
Incluye a: Tablero								
Extiende de: Tablero								
Extendido por: Sistema								
Precondiciones: Establecer correctamente cuando las fichas que son del mismo equipo se pueden posar sobre la misma casilla o coordenada.								
Postcondiciones: Establecer correctamente en la clase tablero el algoritmo para que el sistema lo pueda mostrar de manera correcta								
DATOS DE ENTRADA								
NOMBRE		TIPO		VALIDACIONES				
FichasEn		Evento		Varias fichas de un equipo se posan sobre una casilla libremente				
DATOS DE SALIDA								
NOMBRE		TIPO		VALIDACIONES				
FichasEn		Evento		Cuando varias fichas del mismo equipo se posan en una misma casilla, las fichas no son auto eliminadas				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador aprecia que sus fichas están sanas y salvadas al tener varias fichas en una misma casilla	El sistema garantiza que todas las fichas de los equipos no se eliminen entre ellas al estar posadas en una misma casilla
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El jugador ve que sus fichas son eliminadas cuando tiene fichas de su mismo equipo en una casilla	El sistema no protege correctamente las fichas de un mismo jugador cuando se posan sobre una sola casilla
Requerimientos Especiales: Establecer correctamente en el código la excepción de matar a una ficha siempre y cuando sea del mismo equipo		
Riesgos: El sistema no funciona correctamente ya que las fichas de un mismo jugador al posarse sobre una casilla se auto eliminan		
Criterios de Aceptación: Se realiza la asignación de cuando debe ser eliminada una ficha y cuando no		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politécnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.23. Formato Especificación Caso de Uso 23- EsPuentes

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF23		Nombre Caso de Uso		EsPuentes			
Propósito: Verificar si la ficha puede moverse al existir un puente en el juego								
Alcance: Para todos los jugadores de Parchís								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema							X	
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS								
Incluye a: Casillas								
Extiende de: Casillas								

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Extendido por: Sistema		
Precondiciones: Establecer correctamente el algoritmo para verificar si la ficha debe o no moverse al existir puente		
Postcondiciones: Establecer correctamente en el algoritmo que la ficha pueda o no moverse		
DATOS DE ENTRADA		
NOMBRE	TIPO	VALIDACIONES
EsPuente	Evento	Verificar si la ficha puede moverse libremente, si hay puentes no se moverá
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
EsPuente	Evento	Determinar si la ficha pudo o no seguir su camino
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El jugador aprecia que sus fichas no pueden moverse siempre y cuando exista un puente	El sistema garantiza luego de la verificación que la ficha se pueda o no mover
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El jugador al llegar a un puente pasa por encima de el mismo como si nada	El sistema no realiza la verificación correcta y la ficha evade el puente
Requerimientos Especiales: Establecer correctamente en el código la condición para que una ficha no pueda moverse cuando exista un puente		
Riesgos: El sistema no funciona correctamente, ya que las fichas hacen como si el puente no existiera		
Criterios de Aceptación: Se realiza la verificación de cuando una ficha no debe moverse al existir un puente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.24. Formato Especificación Caso de Uso 24- EsPuenteCasa

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF24	Nombre Caso de Uso	EsPuenteCasa				
Propósito: Determinar cuando existe un puente en la casa del jugador							
Alcance: Para todos los jugadores de Parchís							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema						X	
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Casillas							
Extiende de: Casillas							
Extendido por: Sistema							
Precondiciones: Establecer si existe un puente en la casa del jugador							
Postcondiciones: Determina correctamente cuando existe un puente en la casa de un jugador							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
EsPuenteCasa	Evento		Verificar el puente en la casa de un jugador				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
EsPuenteCasa	Evento		Verificar que haya un puente en la casa del jugador				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El jugador aprecia que sus fichas están formando un puente en la casa del jugador y las fichas enemigas no pueden pasar			El sistema garantiza luego de la verificación que existe un puente en la casa del jugador			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El jugador ve que las fichas enemigas evaden el puente en su casa			El sistema no realiza la verificación correcta			
Requerimientos Especiales: Establecer correctamente en el código la condición de verificar si existe un puente en la casa de los jugadores							
Riesgos: El sistema no funciona correctamente al observar que las fichas pasan el bloqueo o puente como si nada							

Criterios de Aceptación: Se realiza la verificación de cuando una ficha debe continuar o no su camino siempre y cuando no exista un puente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.25. Formato Especificación Caso de Uso 25 - Log

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF25	Nombre Caso de Uso	Log				
Propósito: Mostrar las jugadas, movimientos y demás en un cuadro de texto							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema						X	
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA		ALTA	X
CASOS DE USO ASOCIADOS							
Incluye a: Timer, DetenerJuego, HayGanador							
Extiende de:Sistema							
Extendido por:Tablero							
Precondiciones: El sistema debe tener asignada en la clase GUI todo lo que el LOG llegue a necesitar para poder mostrarlo en la partida							
Postcondiciones: El jugador ha podido observar los mensajes en partida correctamente al ejecutar movimientos o jugadas							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
String	Evento		Los datos o movimientos de la partida se adquieren para ser mostrados en el log				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
String	Evento		Los mensajes de la partida se pueden observar claramente en el log al jugar				
FLUJO DE TRABAJO NORMAL							

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



PASO	USUARIO	SISTEMA
1	El usuario aprecia en el log el historial de juego, en él puede ver movimientos, valor de los dados y mensajes informativos	El sistema muestra correctamente los mensajes en el log, haciendo la experiencia de juego más clara e intuitiva para los jugadores
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario no ve los mensajes en el log o nota que hay fallos en los mensajes	El sistema tiene algún error de código en el log para que genere dichos problemas en el LOG
Requerimientos Especiales: Haber programado exitosamente y cuidadosamente la clase GUI para poder mostrar los mensajes cuando son solicitados		
Riesgos: Mostrar mensajes erróneos en pantalla a los jugadores		
Criterios de Aceptación: Se deben mostrar los mensajes de manera correcta y secuenciada con los movimientos de la partida		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.26. Formato Especificación Caso de Uso 26 - Resaltador

Proyecto	Parchís en C#			Fecha	8-06-2023	
Código Caso de Uso	RF26	Nombre Caso de Uso	Log			
Propósito: Posarse sobre la casa del jugador para indicar que ese jugador es el que tiene el turno de tirar sus dados y mover si es el caso						
Alcance: Se realiza para cada uno de los jugadores de Parchidos						
ACTORES		SIMPLE		MEDIO		COMPLEJO
Sistema						X
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA		ALTA X
CASOS DE USO ASOCIADOS						
Incluye a: Tablero Extiende de: Tablero Extendido por: Sistema						
Precondiciones: Se debe de haber asignado coordenadas de posición y demás para que el circulo resaltador funcione correctamente						
Postcondiciones: El jugador aprecia correctamente los turnos de una forma más intuitiva ya que el circulo rotara entre las casas de los jugadores dependiendo del turno						
DATOS DE ENTRADA						
NOMBRE	TIPO		VALIDACIONES			
Resaltador	Evento		El resaltador se aplica a ciertos metodos para poder rotar sobre los jugadores, por ejemplo, cuando hacen un skip turn.			
DATOS DE SALIDA						
NOMBRE	TIPO		VALIDACIONES			
Resaltador	Evento		Los jugadores aprecian correctamente como el resaltador se posa sobre cada jugador solo cuando es su turno			
FLUJO DE TRABAJO NORMAL						
PASO	USUARIO		SISTEMA			
1	El usuario aprecia el resaltador correctamente durante la partida		El sistema muestra el resaltador correctamente durante la partida			
FLUJO DE TRABAJO ALTERNATIVO 1						
PASO	USUARIO		SISTEMA			
1	El usuario no ve el resaltador y si lo ve no se muestra correctamente		El sistema no muestra correctamente el resaltador y si lo muestra, lo muestra de forma erronea			

Requerimientos Especiales: Asignar correctamente cuando y en dónde se debe posicionar el resaltador		
Riesgos: Mostrar incorrectamente el resaltador		
Criterios de Aceptación: Se debe mostrar correctamente el resaltador		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.27. Formato Especificación Caso de Uso 27 - CasillaEtapaFinal

Proyecto	Parchís en C#	Fecha	8-06-2023
Código Caso de Uso	RF27	Nombre Caso de Uso	CasillaEtapaFinal
Propósito: Asignación de las últimas 32 casillas para cada carril de los jugadores			
Alcance: Se realiza para cada uno de los jugadores de Parchidos			
ACTORES	SIMPLE	MEDIO	COMPLEJO
Sistema			X
COMPLEJIDAD DEL CASO DE USO		BAJA	MEDIA
			ALTA
			X
CASOS DE USO ASOCIADOS			
Incluye a: Casillas			
Extiende de: Casillas			
Extendido por: Tablero			
Precondiciones: Se debe de haber asignado coordenadas de las 32 casillas finales para cada carril del jugador			
Postcondiciones: Las fichas recorren correctamente sus casillas asignadas			
DATOS DE ENTRADA			
NOMBRE	TIPO	VALIDACIONES	
CasillaEtapaFinal	Evento	Se asignan correctamente las posiciones para cada jugador	
DATOS DE SALIDA			
NOMBRE	TIPO	VALIDACIONES	
CasillaEtapaFinal	Evento	Una vez obtenidas las posiciones de los carriles finales para cada jugador, las fichas deberían recorrerlo tranquilamente	
FLUJO DE TRABAJO NORMAL			
PASO	USUARIO	SISTEMA	

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



1	El usuario aprecia el movimiento correctamente	El sistema asigna las posiciones correctamente, por lo tanto, la visualización de las posiciones se debería de ver bien
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario ve los movimientos raros o desalineados	El sistema no muestra correctamente el resaltador y si lo muestra, lo muestra de forma errónea
Requerimientos Especiales: N/A		
Riesgos: Mostrar incorrectamente los movimientos		
Criterios de Aceptación: Se debe mostrar correctamente el movimiento		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel
ANALISTA	CLIENTE	REVISOR

3.28. Formato Especificación Caso de Uso 28 - WinColores

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF28		Nombre Caso de Uso		WinColores			
Propósito: Asignación de las últimas ocho casillas para cada jugador antes de llegar al cielo								
Alcance: Se realiza para cada uno de los jugadores de Parchidos								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema							X	
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA		ALTA	X
CASOS DE USO ASOCIADOS								
Incluye a: Casillas Extiende de:Casillas Extendido por:Tablero								
Precondiciones: Se debe de haber asignado las coordenadas								
Postcondiciones: Las fichas recorren correctamente sus casillas asignadas								
DATOS DE ENTRADA								
NOMBRE	TIPO			VALIDACIONES				
WinColores	Evento			Se asignan correctamente las posiciones de las ultimas ocho casillas para cada jugador				

DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
WinColores	Evento	Las fichas entran a sus últimas ocho casillas siempre y cuando no haya puentes
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El usuario aprecia el movimiento correctamente	El sistema asigna las posiciones correctamente, por lo tanto, la visualización de las posiciones se debería de ver bien
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario ve que su ficha no ingresa en sus casillas previas al cielo correctamente	El sistema no muestra correctamente las fichas entrando a sus debidas últimas ocho casillas debido a una mala asignación
Requerimientos Especiales: Asignar correctamente las coordenadas a cada color		
Riesgos: Mostrar incorrectamente los movimientos		
Criterios de Aceptación: Se debe mostrar correctamente el movimiento para cada una de las fichas		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.29. Formato Especificación Caso de Uso 29 - Casillas

Proyecto	Parchís en C#			Fecha	8-06-2023
Código Caso de Uso	RF29	Nombre Caso de Uso	Casillas		
Propósito: Asignación de las casillas					
Alcance: Se realiza para cada uno de los jugadores de Parchidos					
ACTORES		SIMPLE	MEDIO	COMPLEJO	
Sistema				X	

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA		ALTA	X
CASOS DE USO ASOCIADOS							
Incluye a: EsSeguro, EsPuenteCasa, EsPuente, EsCasa, CasillaEtapaFinal, PuedeMoverCasa, PuestoLibreCaa							
Extiende de: Tablero							
Extendido por: Tablero							
Precondiciones: Se debe de haber asignado las coordenadas generales del juego para el movimiento libre y consistente de las fichas							
Postcondiciones: Las fichas recorren correctamente sus casillas asignadas							
DATOS DE ENTRADA							
NOMBRE		TIPO		VALIDACIONES			
Casillas		Evento		Se asignan con anterioridad las coordenadas de las casillas generales para todos los jugadores, estas coordenadas junto con las anteriores tendrán la función de hacer el movimiento posible dentro del tablero			
DATOS DE SALIDA							
NOMBRE		TIPO		VALIDACIONES			
Casillas		Evento		Los movimientos de cada una de las fichas funcionan correctamente teniendo en cuenta las coordenadas anteriores			
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario aprecia el movimiento correctamente			El sistema asigna las posiciones correctamente, por lo tanto, la visualización de las posiciones se debería de ver bien			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El usuario ve que las fichas no recorren bien el tablero o no funcionan			El sistema no muestra correctamente el movimiento de las fichas en general, por lo tanto, crea un disgusto en los jugadores al no apreciar el correcto funcionamiento de sus fichas			
Requerimientos Especiales: Asignar correctamente las coordenadas generales							
Riesgos: Mostrar incorrectamente los movimientos							
Criterios de Aceptación: Se debe mostrar correctamente el recorrido de las fichas por todo el tablero							
Pantalla Propuesta:							
FIRMAS							
Juan Jaraba		Politecnico		Darmabel			

3.30. Formato Especificación Caso de Uso 30 - SeleccionarJugadores

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF30	Nombre Caso de Uso	SeleccionarJugadores				
Propósito: Seleccionar jugadores de la partida							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				X			
Jugador				X			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	X	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: HabilitarColoresFicha							
Extiende de:Sistema							
Extendido por:Jugador							
Precondiciones: Antes de iniciar el juego el jugador deberá escoger la cantidad de jugadores que desea en el juego, por defecto estarán todos activos							
Postcondiciones: El jugador tendrá los jugadores seleccionados sin importar la casa							
DATOS DE ENTRADA							
NOMBRE		TIPO		VALIDACIONES			
SeleccionarJugadores		Evento		El jugador podrá seleccionar los jugadores deseados, luego de seleccionarlos el juego empezará solo con los jugadores recibidos anteriormente seleccionados			
DATOS DE SALIDA							
NOMBRE		TIPO		VALIDACIONES			
SeleccionarJugadores		Evento		En la partida solo se aprecian los jugadores que fueron seleccionados, se puede seleccionar solo un jugador pero para el juego se necesitan mínimo 2 jugadores			
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario observa que solo tiene fichas los jugadores antes de que iniciara el juego			El sistema solo muestra las fichas de las casas seleccionadas para jugar antes de iniciar el juego			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El usuario ve que la opcion para seleccionar jugadores no sive, ya que las casas no seleccionadas aparecen habilitadas en el juego			El sistema muestra de manera erronea a los jugadores que seleccionaron ciertas casas, por lo tanto no cumple con su función			

Requerimientos Especiales: N/A		
Riesgos: Mostrar incorrectamente los jugadores a la hora de ser seleccionados		
Criterios de Aceptación: Se debe mostrar correctamente las casas que han sido seleccionadas sin errores		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.31. Formato Especificación Caso de Uso 31 - HabilitarColoresFicha

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF31		Nombre Caso de Uso		HabilitarColoresFicha		
Propósito: Ayuda para ver los colores y nombres de las fichas a la hora de seleccionar los jugadores deseados para jugar							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES			SIMPLE		MEDIO		COMPLEJO
Sistema					x		
Jugador					x		
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	x	ALTA
CASOS DE USO ASOCIADOS							
Incluye a: SeleccionarJugadores							
Extiende de: SeleccionarJugadores							
Extendido por:Jugador							
Precondiciones: Al seleccionar los jugadores que se desean en la partida, se puede apreciar el color de la ficha y el nombre de la misma para saber que fichas se seleccionan							
Postcondiciones: Las fichas e imágenes de referencia ayudan al jugador a saber con qué color de ficha desea jugar							
DATOS DE ENTRADA							
NOMBRE	TIPO			VALIDACIONES			
HabilitarColoresFicha	Evento			Se asignan los nombres de las fichas junto a una referencia de la misma para así obtener de manera correcta la ficha que se desea jugar sin errores			
DATOS DE SALIDA							
NOMBRE	TIPO			VALIDACIONES			
HabilitarColoresFicha	Evento			Las fichas al jugar Parchidos corresponden a las fichas seleccionadas anteriormente gracias a las imagenes y nombres de referencia			
FLUJO DE TRABAJO NORMAL							

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



PASO	USUARIO	SISTEMA
1	El usuario aprecia las fichas que eligió correctamente	El sistema asigna las fichas correctamente en sus casas correspondientes
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario ve que las fichas no son del color ni personaje que el eligió	El sistema no muestra correctamente las fichas y el color del personaje que el jugador decidió elegir
Requerimientos Especiales: Asignar correctamente las imágenes y nombres de los personajes frente a Seleccionar Jugadores correctamente para evitar malentendidos		
Riesgos: Mostrar incorrectamente los jugadores		
Criterios de Aceptación: Se debe mostrar correctamente las fichas seleccionadas		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.32. Formato Especificación Caso de Uso 32 - SaltarTurno

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF32		Nombre Caso de Uso		SaltarTurno			
Propósito: Permite saltar turno a cualquier jugador cuando lo vea favorable								
Alcance: Se realiza para cada uno de los jugadores de Parchidos								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema					x			
Jugador					x			
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	x	ALTA	
CASOS DE USO ASOCIADOS								
Incluye a: Jugador Extiende de: Jugador Extendido por: Tablero								
Precondiciones: El jugador debe de haber empezado una partida								
Postcondiciones: El jugador luego de iniciar la partida podrá saltar turno siempre y cuando sea su turno de jugar								
DATOS DE ENTRADA								
NOMBRE		TIPO		VALIDACIONES				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



SaltaTurno	Evento	Al presionar el boton saltar turno se establece el valor de dadosActivos en 0 y el valor de ValorSeleccionado en -1
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
SaltaTurno	Evento	Al tener los valores predeterminados se entiende que el jugador ya lanzo o hizo su jugada, por lo tanto, cede el siguiente turno
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	El usuario aprecia que al presionar Pasa Turno el circulo cede a la siguiente persona	El sistema asigna hace el traspaso del turno correctamente
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario observa que al presionar Pasar Turno no cede el turno	El sistema no cede el turno cuando se lo piden por medio del botón
Requerimientos Especiales: N/A		
Riesgos: No skippear el turno cuando se le requiere al sistema		
Criterios de Aceptación: Se debe skippear correctamente el turno de un jugador si así lo desea el		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.33. Formato Especificación Caso de Uso 33 - BotonDado

Proyecto	Parchís en C#			Fecha	8-06-2023
Código Caso de Uso	RF33	Nombre Caso de Uso	BotonDado		
Propósito: Permitir que el jugador seleccione que dado desea tomar para asignárselo a una ficha					
Alcance: Se realiza para cada uno de los jugadores de Parchidos					
ACTORES		SIMPLE	MEDIO	COMPLEJO	
Sistema			x		
Jugador			x		

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



COMPLEJIDAD DEL CASO DE USO		BAJA	MEDIA	x	ALTA
CASOS DE USO ASOCIADOS					
Incluye a: Jugador					
Extiende de: Jugador					
Extendido por: Tablero					
Precondiciones: El jugador debe de haber empezado una partida					
Postcondiciones: El jugador luego de iniciar la partida y lanzar los dados podrá seleccionar su dado					
DATOS DE ENTRADA					
NOMBRE	TIPO	VALIDACIONES			
BotonDado	Evento	Al presionar el dado de preferencia obtendrá el valor			
DATOS DE SALIDA					
NOMBRE	TIPO	VALIDACIONES			
BotonDado	Evento	Una vez seleccionado el dado, se tiene que seleccionar la ficha a mover			
FLUJO DE TRABAJO NORMAL					
PASO	USUARIO	SISTEMA			
1	El usuario aprecia que al presionar el dado y posar el mouse sobre una ficha, la ficha se mueve la posición que indicaba el dado	El sistema deja al jugador seleccionar el dado con su respectivo valor para mover la ficha a elegir			
FLUJO DE TRABAJO ALTERNATIVO 1					
PASO	USUARIO	SISTEMA			
1	El usuario aprecia que al seleccionar el dado no funciona, por lo tanto, no puede mover alguna ficha	El sistema no funciona al querer seleccionar el dado para asignarlo a la ficha			
Requerimientos Especiales: El jugador debe lanzar los dados y tener fichas en juego para poder seleccionar el dado con el valor a mover					
Riesgos: No funciona en la selección del dado, por lo tanto, el juego no funciona para mover fichas					
Criterios de Aceptación: Se debe seleccionar el valor a desear correctamente para poder asignárselo a la ficha elegida					
Pantalla Propuesta:					
FIRMAS					
Juan Jaraba	Politecnico	Darmabel			

3.34. Formato Especificación Caso de Uso 34 – DiceGrafth

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF34	Nombre Caso de Uso	DiceGrafth				
Propósito: Mostrar los dados con su respectivo valor							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				x			
Jugador				x			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	x	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							
Extiende de: Tablero							
Extendido por: Sistema, Jugador							
Precondiciones: El jugador debe de haber abierto Parchidos							
Postcondiciones: El jugador al iniciar Parchidos podrá apreciar los dados en el tablero							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
DiceGrafth	Evento		Al iniciar el juego se podrán apreciar los dados				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
DiceGrafth	Evento		Se aprecian los dados con sus respectivos valores				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario aprecia los dados con sus respectivos valores			El sistema muestra los dados correctamente			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El usuario no aprecia los dados correctamente			El sistema no funciona al querer mostrar los dados con sus respectivas caras			
Requerimientos Especiales: Asignar correctamente las caras de los dados con el valor							
Riesgos: Que no se muestre o no funcione							
Criterios de Aceptación:							

Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.35. Formato Especificación Caso de Uso 35 – GetCoin

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF35	Nombre Caso de Uso	GetCoin				
Propósito: Permitir seleccionar la ficha a jugar							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				x			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	x	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							
Extiende de: Tablero							
Extendido por: Sistema							
Precondiciones: El jugador debe de haber lanzado los dados para seleccionar una ficha, de ahí el método GetCoin							
Postcondiciones: El jugador obtiene la posición de la ficha a mover en el juego, esta ficha dependiendo del color tiene un valor							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
GetCoin	Evento		Se obtiene la posición de la ficha				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
GetCoin	Evento		Se obtiene la posición de la ficha				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO		SISTEMA				
1	Las fichas se aprecian correctamente		El sistema suma baseID y id que representa la posición de la ficha dentro del equipo				

FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	Las fichas no se aprecian correctamente	El sistema no funciona al querer mostrar los dados con sus respectivas caras
Requerimientos Especiales: N/A		
Riesgos: N/A		
Criterios de Aceptación: N/A		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.36. Formato Especificación Caso de Uso 35 – PreviewCoin

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF36		Nombre Caso de Uso		PreviewCoin		
Propósito: Crear copia de la ficha a mover							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES			SIMPLE		MEDIO		COMPLEJO
Sistema					x		
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	x	ALTA
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							
Extiende de: Tablero							
Extendido por: Sistema							
Precondiciones: El jugador debe de haber lanzado los dados para seleccionar una ficha, al seleccionar la ficha se mostrará el posible movimiento							
Postcondiciones: El jugador obtiene la futura posición de la ficha al seleccionar una ficha con un valor							
DATOS DE ENTRADA							
NOMBRE		TIPO		VALIDACIONES			
PreviewCoin		Evento		Al seleccionar la ficha que se desea mover se mostrará una ficha fantasma			
DATOS DE SALIDA							

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



NOMBRE		TIPO	VALIDACIONES
PreviewCoin		Evento	Se muestra la posición de la ficha fantasma, antes de esto hay que seleccionar un dado y posar el mouse sobre la ficha que se desea mover para que pueda representarse en el tablero
FLUJO DE TRABAJO NORMAL			
PASO	USUARIO		SISTEMA
1	Se aprecia la ficha fantasma correctamente antes de mover la ficha		El sistema muestra la ficha fantasma correctamente
FLUJO DE TRABAJO ALTERNATIVO 1			
PASO	USUARIO		SISTEMA
1	La ficha fantasma no se aprecia o se ve		El sistema no funciona al querer mostrar la ficha fantasma
Requerimientos Especiales: Programar correctamente el funcionamiento de la ficha fantasma			
Riesgos: N/A			
Criterios de Aceptación: N/A			
Pantalla Propuesta:			
FIRMAS			
Juan Jaraba		Politecnico	Darmabel

3.37. Formato Especificación Caso de Uso 37 – HidePreviewCoin

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF37		Nombre Caso de Uso	HidePreviewCoin			
Propósito: Ocultar la ficha fantasma							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES			SIMPLE		MEDIO		COMPLEJO
Sistema					x		
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	x	ALTA
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Extiende de: Tablero		
Extendido por: Sistema		
Precondiciones: El jugador debe posar el mouse sobre la ficha		
Postcondiciones: El jugador luego de posar el mouse sobre la ficha y quitarlo verá que la ficha fantasma se ha ocultado		
DATOS DE ENTRADA		
NOMBRE	TIPO	VALIDACIONES
HidePreviewCoin	Evento	La ficha aparece al existir una ficha en estado de seleccion
DATOS DE SALIDA		
NOMBRE	TIPO	VALIDACIONES
HidePreviewCoin	Evento	La ficha desaparece en el momento en que la ficha ya no ha sido seleccionada
FLUJO DE TRABAJO NORMAL		
PASO	USUARIO	SISTEMA
1	Las fichas se aprecian correctamente en el momento adecuado	El sistema oculta correctamente la ficha fantasma
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	Las fichas no se aprecian correctamente en el momento adecuado, además no se oculta cuando es necesario	El sistema no funciona al querer ocultar la ficha fantasma
Requerimientos Especiales: Ocultar correctamente la ficha fantasma al ver que ya no hay una ficha común seleccionada		
Riesgos: N/A		
Criterios de Aceptación: La ficha fantasma se oculta correctamente		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.38. Formato Especificación Caso de Uso 38 – CoinClick

Proyecto	Parchís en C#	Fecha	8-06-2023
-----------------	---------------	--------------	-----------

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Código Caso de Uso	RF38	Nombre Caso de Uso	CoinClick		
Propósito: Presionar la ficha que se quiere mover					
Alcance: Se realiza para cada uno de los jugadores de Parchidos					
ACTORES		SIMPLE	MEDIO		COMPLEJO
Sistema			x		
Jugador		x			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	ALTA
			x		
CASOS DE USO ASOCIADOS					
Incluye a: Jugador Extiende de: Mover Extendido por: Sistema					
Precondiciones: El jugador debe de haber lanzado los dados, luego de lanzarlos podrá asignar el valor de movimiento a la ficha solo cuando de clic en ella					
Postcondiciones: El jugador al dar clic a la ficha con un valor previamente escogido de los dados, vera que su movimiento es completamente exitoso, si la ficha no se mueve es porque habrá un puente					
DATOS DE ENTRADA					
NOMBRE	TIPO	VALIDACIONES			
CoinClick	Evento	Se obtiene el valor que se le va a asignar a la ficha, al presionar en la misma, la ficha obtiene el valor y se mueve las posiciones indicadas por el jugador por medio del dado			
DATOS DE SALIDA					
NOMBRE	TIPO	VALIDACIONES			
CoinClick	Evento	Se muestra que la ficha que ha sido seleccionada se mueve correctamente por el tablero			
FLUJO DE TRABAJO NORMAL					
PASO	USUARIO	SISTEMA			
1	El movimiento de la ficha se aprecia correctamente cuando el usuario presiona sobre la misma con un valor	El sistema hace el correcto movimiento solo cuando el jugador presiona sobre la ficha para asignarle el valor de cuantas casillas va a mover			
FLUJO DE TRABAJO ALTERNATIVO 1					
PASO	USUARIO	SISTEMA			
1	Las fichas no se mueven cuando el usuario quiere asignarle un valor	El sistema no funciona al no mover la ficha seleccionada			
Requerimientos Especiales: Tener en cuenta que el movimiento solo se debe generar cuando un jugador va a asignar algún valor del dado en al ficha					
Riesgos: Si la ficha que el jugador selecciona no se mueve najo ningún termino al querer moverla por medio de un valor, el juego está mal programado					

Criterios de Aceptación: Los movimientos de las fichas al presionar sobre ellas se ejecutan correctamente solo cuando se les va a asignar un valor		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.39. Formato Especificación Caso de Uso 39 – PasaTurno_Click

Proyecto	Parchís en C#					Fecha	8-06-2023	
Código Caso de Uso	RF39		Nombre Caso de Uso		PasaTurno_Click			
Propósito: Se encarga de establecer cuando termina el juego mostrando al ganador en consola								
Alcance: Se realiza para cada uno de los jugadores de Parchidos								
ACTORES			SIMPLE		MEDIO		COMPLEJO	
Sistema					x			
COMPLEJIDAD DEL CASO DE USO			BAJA		MEDIA	x	ALTA	
CASOS DE USO ASOCIADOS								
Incluye a: Log Extiende de: Log Extendido por: Sistema								
Precondiciones: Un jugador debe de haber llegado a la meta con todas sus fichas para que este método se ejecute								
Postcondiciones: El jugador ve que el juego termina mostrando el nombre del ganador en consola								
DATOS DE ENTRADA								
NOMBRE	TIPO			VALIDACIONES				
PasaTurno_Click	Evento			Se obtiene al ganador y se detiene el juego simultáneamente				
DATOS DE SALIDA								
NOMBRE	TIPO			VALIDACIONES				
PasaTurno_Click	Evento			Al obtener el ganador se mostrará en el log, ahí se terminará el juego Parchidos				
FLUJO DE TRABAJO NORMAL								
PASO	USUARIO			SISTEMA				
1	El usuario aprecia correctamente el nombre del			El sistema establece correctamente cuando termina el juego				

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



	jugador	Parchidos mostrando el ganador en consola
FLUJO DE TRABAJO ALTERNATIVO 1		
PASO	USUARIO	SISTEMA
1	El usuario no detecta o ve quien gano	El sistema no funciona ni establece quien fue el ganador del juego aunque el jugador tenga todas las fichas en el cielo
Requerimientos Especiales: Haber especificado en el código de PasaTurno_Tick que debe terminar el juego llamando al método del tablero correspondiente además se debe mostrar al debido ganador y reestablecer el juego		
Riesgos: No se ejecuta el método correctamente y el jugador deba detener el juego manualmente		
Criterios de Aceptación: Todo debe estar correctamente programado		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

3.40. Formato Especificación Caso de Uso 40 – Ficha

Proyecto	Parchís en C#			Fecha	8-06-2023
Código Caso de Uso	RF40	Nombre Caso de Uso	Ficha		
Propósito: Se encarga de establecer las imágenes de las fichas, posicionamiento inicial y asignación a casas según su color					
Alcance: Se realiza para cada uno de los jugadores de Parchidos					

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				x			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	x	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Tablero Extiende de: Tablero Extendido por: Sistema							
Precondiciones: Disponer de sus características definidas dentro de la clase Tablero, para ser llamada correctamente							
Postcondiciones: Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero correctamente							
DATOS DE ENTRADA							
NOMBRE		TIPO		VALIDACIONES			
Ficha		Evento		Se obtienen las características de las fichas, como las imágenes, posicionamiento inicial y asignación a casas según su color			
DATOS DE SALIDA							
NOMBRE		TIPO		VALIDACIONES			
Ficha		Evento		Se muestran las imágenes, posicionamientos y asignación de las fichas a sus casas correctamente			
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario aprecia correctamente lo anteriormente mencionado (posiciones, imágenes etc)			El sistema establece correctamente las características de las fichas			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El usuario detecta que las fichas con sus imágenes y posiciones iniciales no están correctas			El sistema no funciona ni establece correctamente las características de las fichas			
Requerimientos Especiales: Asignar correctamente las características anteriormente mencionadas para el correcto funcionamiento del juego en cuestiones de interfaz							
Riesgos: No se muestran las características deseadas							
Criterios de Aceptación: Todo debe estar correctamente programado							
Pantalla Propuesta:							
FIRMAS							
Juan Jaraba		Politécnico			Darmabel		

3.41. Formato Especificación Caso de Uso 41 – Main

Proyecto	Parchís en C#				Fecha	8-06-2023	
Código Caso de Uso	RF41	Nombre Caso de Uso	Main				
Propósito: Ejecución o llamado del formato FORM1 Main							
Alcance: Se realiza para cada uno de los jugadores de Parchidos							
ACTORES		SIMPLE		MEDIO		COMPLEJO	
Sistema				x			
COMPLEJIDAD DEL CASO DE USO		BAJA		MEDIA	x	ALTA	
CASOS DE USO ASOCIADOS							
Incluye a: Tablero							
Extiende de: Tablero							
Extendido por: Sistema							
Precondiciones: Disponer del diseño completo del tablero, por ejemplo: tablero, dados, botones, check box log y ayudas visuales para una mejor comprensión del juego hacia el usuario							
Postcondiciones: Se mostrará de forma correcta todos los elementos que conforman el juego al momento de su ejecución							
DATOS DE ENTRADA							
NOMBRE	TIPO		VALIDACIONES				
Main	Evento		Se obtienen las características de las fichas, como las imágenes, posicionamiento inicial y asignación a casas según su color				
DATOS DE SALIDA							
NOMBRE	TIPO		VALIDACIONES				
Main	Evento		Mostrar correctamente toda la interfaz gráfica creada en el Forms Parchidos				
FLUJO DE TRABAJO NORMAL							
PASO	USUARIO			SISTEMA			
1	El usuario aprecia correctamente la carga de todos los recursos del Forms			El sistema ejecuta de forma exitosa todos los complementos que hacen del juego una buena experiencia			
FLUJO DE TRABAJO ALTERNATIVO 1							
PASO	USUARIO			SISTEMA			
1	El usuario no aprecia el juego correctamente, por lo tanto, no puede jugar como es debido			El sistema no ejecuta correctamente los elementos del Forms arruinando completamente la experiencia de juego			

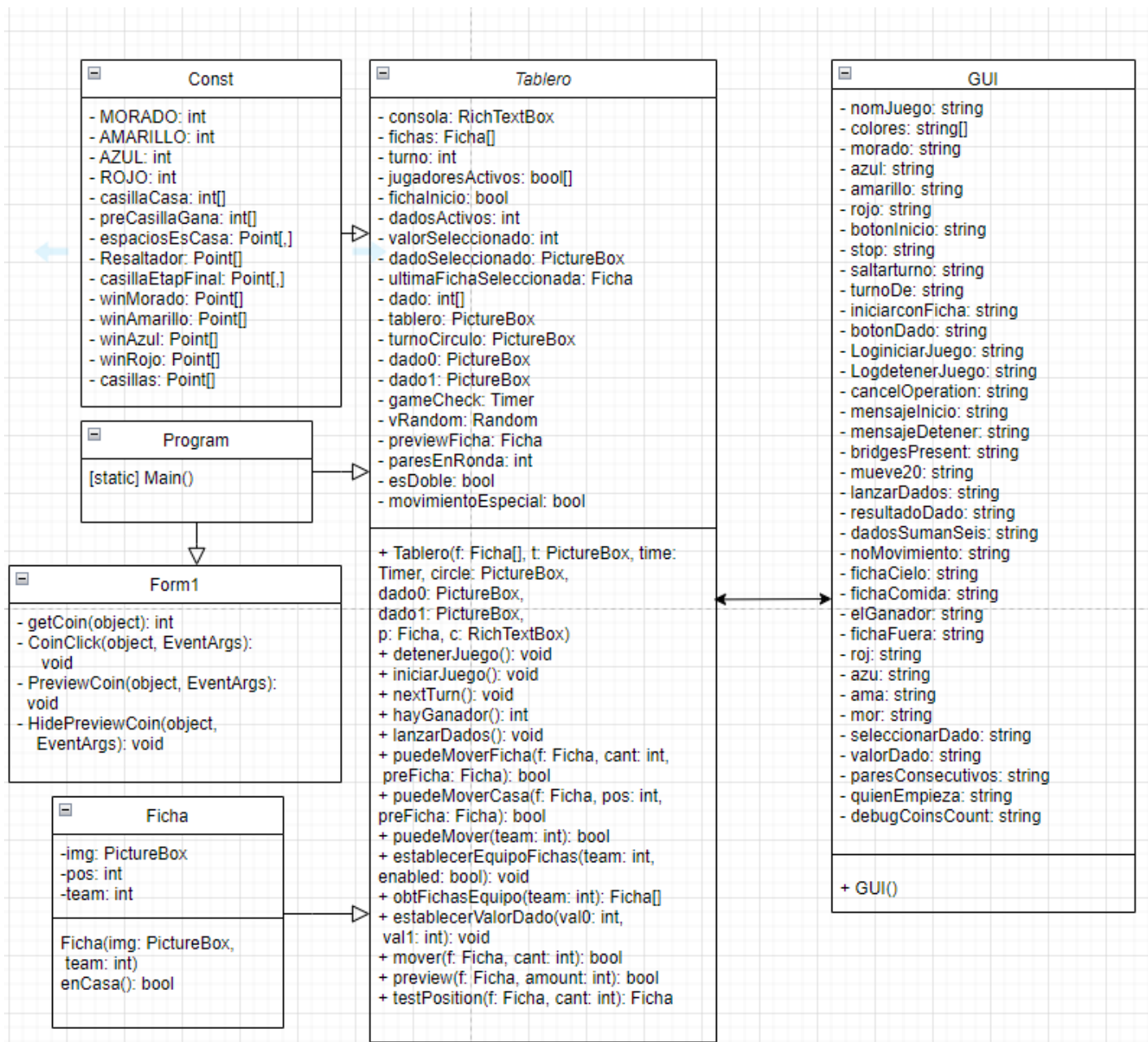
FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



Requerimientos Especiales: Asignar correctamente las características anteriormente mencionadas para el correcto funcionamiento del juego al ser ejecutado		
Riesgos: No se muestra el Forms como se espera		
Criterios de Aceptación: Todo debe estar correctamente programado		
Pantalla Propuesta:		
FIRMAS		
Juan Jaraba	Politecnico	Darmabel

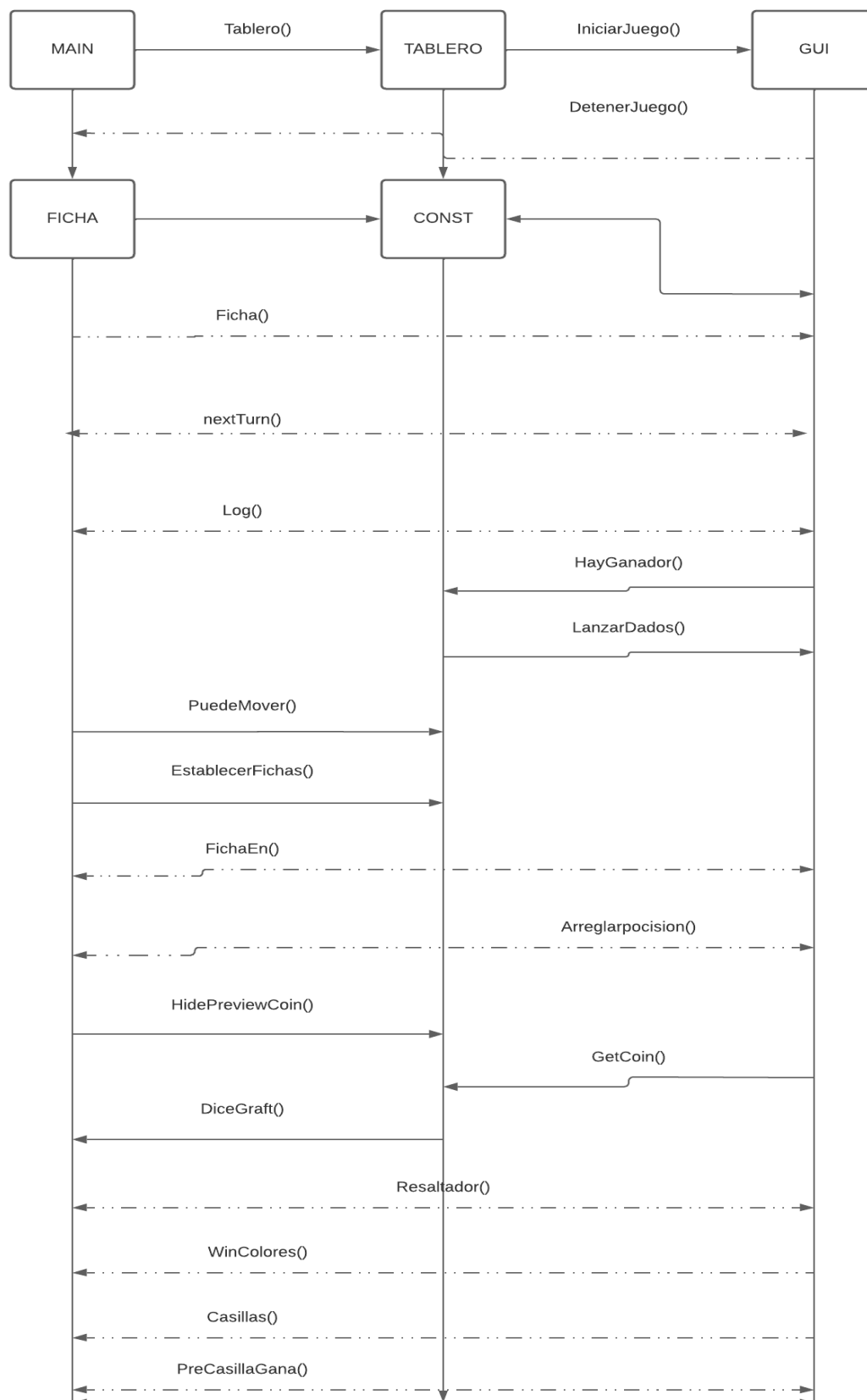
4. Diagrama de Clases

4.1. Diagrama de Clases Parchidos



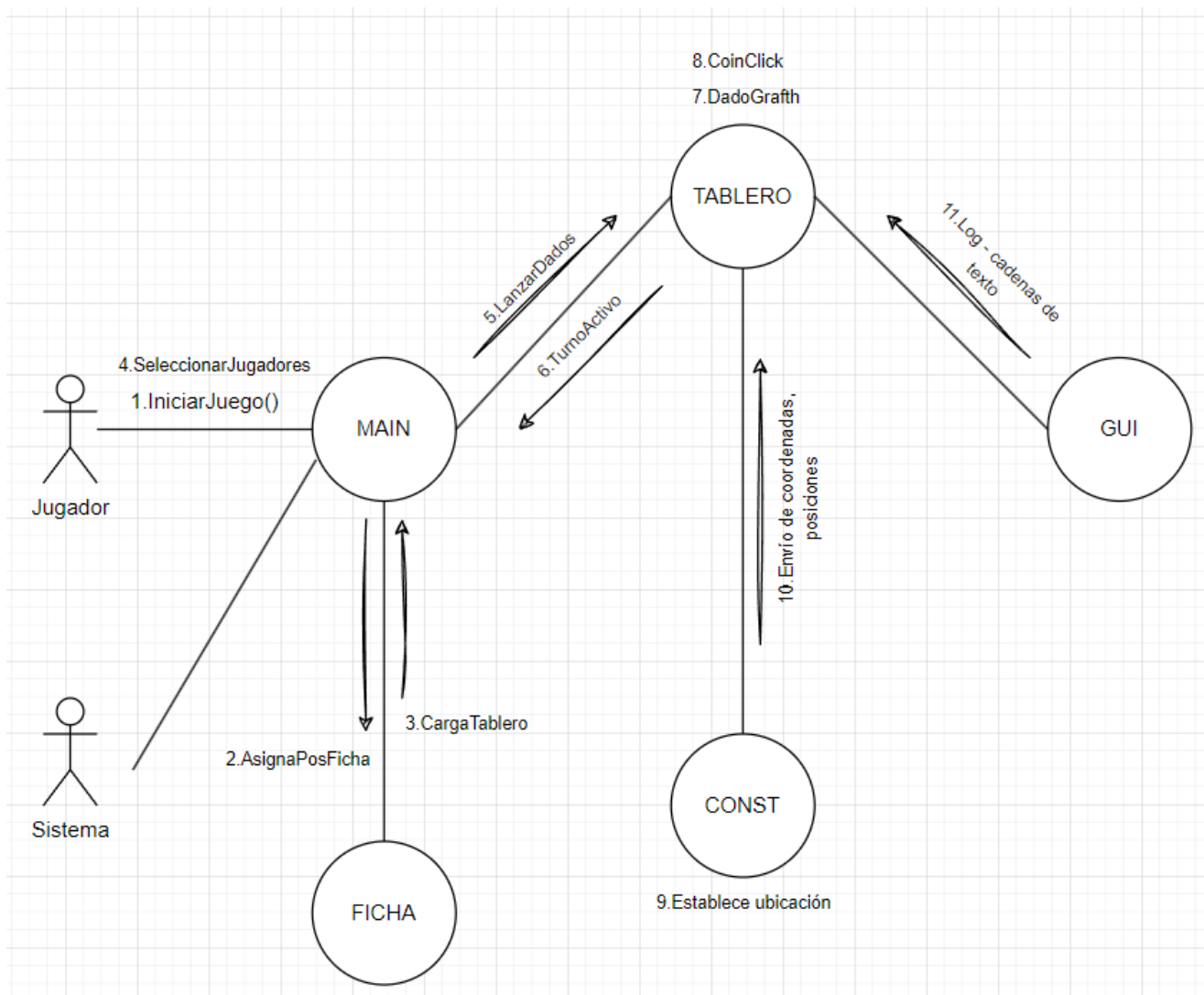
5. Diagrama de Secuencia

5.1. Diagrama de secuencia Parchidos



6. Diagrama de Colaboración

6.1. Diagrama de Colaboración Parchidos



7. Contratos de Comportamiento del Sistema

7.1. Contratos de Comportamiento Clase 1

PROYECTO	Parchidos	FECHA	19/06/023
CLASE	Ficha		
PROPÓSITO DE LA CLASE	Establece las imágenes de las fichas, posicionamiento inicial y asignación a casas según su color		
ATRIBUTOS	<ul style="list-style-type: none">No tiene		

NOMBRE CONTRATO 1	public Ficha (PictureBox Img, int equipo)
PROPÓSITO	Establece que el recurso para la visualización de la ficha es en formato imagen, un atributo para las fichas inicial, en juego y final.
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Mostrar en el tablero las imágenes adecuadas para cada Ficha en su pocion inicial, una vez en juego establecer su posición en juego en curso y cuan lleguen a cielo, establecer su estado en fuera de juego para no tenerla más en cuenta.
PRECONDICIONES	Disponer de sus características definidas dentro de la clase Tablero, para ser llamada de forma adecuada
POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta

NOMBRE CONTRATO 2	public bool EnCasa()
PROPÓSITO	Indicar que si la posición es mayor a cien es EnCasa
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Devolver un valor booleano
PRECONDICIONES	Disponer de un espacio en el tablero que se denomine EnCasa
POSTCONDICIONES	Son visibles los espacios dentro del tablero definidos como casa, cada uno con un color, que seran los colores de los jugadores y sus fichas.

FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

7.2. Contratos de Comportamiento Clase 2

PROYECTO	Parchidos	FECHA	19/06/023
CLASE	Const		
PROPÓSITO DE LA CLASE	Establecer las coordenadas y pociones de las casas, casillas especiales, casillas generales del tablero y asignacion de posicion para cada color de jugador		
ATRIBUTOS	<ul style="list-style-type: none"> • Int 		

NOMBRE CONTRATO 1	public const int Rojo
PROPÓSITO	Establecer un valor para llamar al equipo de fichas Rojo
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Si es su turno correspondiente validara el movimiento a su color respectivo, asignando cuando sea el caso el uso de la carcel y el ingreso al tramo final del tablero, que corresponde al color de la ficha.
PRECONDICIONES	Disponer de sus características definidas dentro de la clase Tablero, para ser llamada de forma adecuada
POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta, así como el ingreso al tramo final según su color-

NOMBRE CONTRATO 2	public const int Morado
PROPÓSITO	Establecer un valor para llamar al equipo de fichas Morado
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Si es su turno correspondiente validara el movimiento a su color respectivo, asignando cuando sea el caso el uso de la carcel y el ingreso al tramo final del tablero, que corresponde al color de la ficha.
PRECONDICIONES	Disponer de sus características definidas dentro de la clase Tablero, para ser llamada de forma adecuada.
POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta, así como el ingreso al tramo final según su color.

NOMBRE CONTRATO 3	public const int Amarillo
PROPÓSITO	Establecer un valor para llamar al equipo de fichas Amarillo

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



REFERENCIA A CASOS DE USO	Tablero
SALIDA	Si es su turno correspondiente validara el movimiento a su color respectivo, asignando cuando sea el caso el uso de la carcel y el ingreso al tramo final del tablero, que corresponde al color de la ficha.
PRECONDICIONES	Disponer de sus características definidas dentro de la clase Tablero, para ser llamada de forma adecuada.
POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta, así como el ingreso al tramo final según su color

NOMBRE CONTRATO 4	public const int Azul
PROPÓSITO	Establecer un valor para llamar al equipo de fichas Azul
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Si es su turno correspondiente validara el movimiento a su color respectivo, asignando cuando sea el caso el uso de la carcel y el ingreso al tramo final del tablero, que corresponde al color de la ficha.
PRECONDICIONES	Disponer de sus características definidas dentro de la clase Tablero, para ser llamada de forma adecuada.
POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta, así como el ingreso al tramo final según su color.

NOMBRE CONTRATO 5	public static int[] CasillaCasa
PROPÓSITO	Establece las casillas donde estaran ubicadas las casas dentro del tablero de juego
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Mostrará las fichas dentro de su ubicación en casa dentro del tablero, una en cada esquina del mismo.
PRECONDICIONES	Tener un tablero al cual asignarle la ubicación de las casas
POSTCONDICIONES	Se mostrarán las fichas dentro de su respectiva casa por medio de la posición y el respectivo color asignado

NOMBRE CONTRATO 6	public static int[] PreCasillaGana
PROPÓSITO	Definir las casillas donde se inicia el acceso a la etapa final del juego, las ultimas 8 casillas para enviar una ficha al cielo y ganar el juego.
REFERENCIA A CASOS DE USO	Tablero
SALIDA	En el tablero se mostrará según el color las ultimas 8 casillas para ganar el juego.
PRECONDICIONES	Disponer del diseño del tablero, asignadas las casillas, casa, especiales, cielo, pre-casillas de ganar y generales
POSTCONDICIONES	Se mostrará el correcto desplazamiento de las fichas por las casillas finales del recorrido

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



NOMBRE CONTRATO 7	public static Point[,] EspaciosEsCasa
PROPÓSITO	Establecer una posición en los ejes X e Y, donde se dispondrán las fichas de cada casa
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Se verá reflejado en el tablero de forma ordenada las fichas dentro de sus posiciones en sus respectivas casas
PRECONDICIONES	Establecer la ubicación de las casas (posición), para así poder asignar las coordenadas de forma correcta X e Y.
POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta, en su respectivo espacio dentro de la casa

NOMBRE CONTRATO 8	public static Point[] Resaltador
PROPÓSITO	Mostrar que jugador se encuentra con un turno activo, para dar orden a la partida
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Mostrar un círculo que resaltara la casa del jugador activo en partida
PRECONDICIONES	Establecer el tablero, fichas, casa, inicio de juego y un paso de turno.
POSTCONDICIONES	Dicho resaltador se posará sobre la casa del jugador con un turno activo.

NOMBRE CONTRATO 9	public static Point[,] CasillaEtapFinal
PROPÓSITO	Establecer las últimas casillas del juego, llamado etapa final
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Las fichas se desplazarán correctamente por el último tramo, solo en su color correspondiente
PRECONDICIONES	Establecer el tablero, fichas, casa, inicio de juego y un paso de turno.
POSTCONDICIONES	Las fichas se desplazarán de forma correcta por las coordenadas asignadas a las 8 últimas casillas del recorrido según su color de ficha.

NOMBRE CONTRATO 10	public static Point[] WinRojo
PROPÓSITO	Asignar las últimas 8 casillas al color Rojo
REFERENCIA A CASOS DE USO	Tablero

FACULTAD DE TÉCNICAS DE INGENIERÍA
Programa de Desarrollo de Software
Asignatura: 0553 Ingeniería de Software



SALIDA	Se desplazarán por esta zona únicamente las fichas color rojo
PRECONDICIONES	Establecer el tablero, fichas, casa, inicio de juego y un paso de turno.
POSTCONDICIONES	Las fichas se desplazarán de forma correcta por las coordenadas asignadas a las 8 últimas casillas del recorrido según su color de ficha.

NOMBRE CONTRATO 11	public static Point[] WinMorado
PROPÓSITO	Asignar las últimas 8 casillas al color Morado
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Se desplazarán por esta zona únicamente las fichas color rojo
PRECONDICIONES	Establecer el tablero, fichas, casa, inicio de juego y un paso de turno.
POSTCONDICIONES	Las fichas se desplazarán de forma correcta por las coordenadas asignadas a las 8 últimas casillas del recorrido según su color de ficha.

NOMBRE CONTRATO 12	public static Point[] WinAmarillo
PROPÓSITO	Asignar las últimas 8 casillas al color Amarillo
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Se desplazarán por esta zona únicamente las fichas color rojo
PRECONDICIONES	Establecer el tablero, fichas, casa, inicio de juego y un paso de turno.
POSTCONDICIONES	Las fichas se desplazarán de forma correcta por las coordenadas asignadas a las 8 últimas casillas del recorrido según su color de ficha.

NOMBRE CONTRATO 13	public static Point[] WinAzul
PROPÓSITO	Asignar las últimas 8 casillas al color Azul
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Se desplazarán por esta zona únicamente las fichas color rojo
PRECONDICIONES	Establecer el tablero, fichas, casa, inicio de juego y un paso de turno.
POSTCONDICIONES	Las fichas se desplazarán de forma correcta por las coordenadas asignadas a las 8 últimas casillas del recorrido según su color de ficha.

NOMBRE CONTRATO 14	public static Point[] Casillas
PROPÓSITO	Establece la ubicación en el plano X e Y, de todas las casillas generales dentro del tablero de juego
REFERENCIA A CASOS DE USO	Tablero
SALIDA	Mostrar en el tablero las imágenes adecuadas para cada Ficha durante el desplazamiento dentro del tablero, permitiendo el posicionamiento de más de una ficha en la misma casilla
PRECONDICIONES	Disponer del diseño del tablero, identificando las casillas que serán especiales, para la asignación correcta de las coordenadas de todas las demás casillas.

POSTCONDICIONES	Se mostrarán las imágenes de las fichas y su posicionamiento dentro del tablero de forma correcta
------------------------	---

FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR

7.3. Contratos de Comportamiento Clase 3

PROYECTO	Parchidos	FECHA	19/06/023
CLASE	Program		
PROPÓSITO DE LA CLASE	Indicar a el programa la correcta ejecución del FORMS1 (MAIN)		
ATRIBUTOS	<ul style="list-style-type: none"> N/A 		

NOMBRE CONTRATO 14	static void Main()
PROPÓSITO	Ejecución o llamado del formato Main
REFERENCIA A CASOS DE USO	Main
SALIDA	Mostrar de forma adecuada toda la interfaz gráfica creada en el Forms1 Main, el llamado al método MAIN
PRECONDICIONES	Disponer del diseño del tablero, dados, fichas, botones, check box, y ayudas visuales para una mejor comprensión del juego hacia el usuario.
POSTCONDICIONES	Cargara de forma adecuada todos los elementos agregados en el Forms1 al momento de su ejecucion.

FIRMAS		
Jose Aguirre	Politecnico	Darmabel Orozco
ANALISTA	CLIENTE	REVISOR