****

**Grado Universitario en Ingeniería Informática**

**2020-2021**

***Trabajo Fin de Grado***

**Adaptación del Catán a Boardgame.io**

Adrián Sanz Gómez

**Tutor/es**

Javier Ignacio Carbo Rubiera

Alejandro Cervantes Rovira

**LEGANÉS, SEPTIEMBRE 2021**

**Agradecimientos:**

Quiero dar las gracias y dedicarle este trabajo a mi familia, mis amigos y mis dos tutores Javier y Alejandro por todo el apoyo, tanto moral como físico para el desarrollo de este trabajo.

Sobre todo, muchas gracias a mi grupo de amigos de la uni: Un Rostro y BC Academy, en especial a mis dos hermanos Rubén y Fran por todas las aventuras y buenos momentos que hemos tenido en esta etapa y todo lo que aún nos queda. Gracias por estar ahí siempre. Recordad que aún no ha salido el River, camaradas.

Gracias también a Parrado por siempre ser un pilar a mi lado y ser el amigo y compañero perfecto, gracias a Mahou por siempre estar dispuesto a echar un cable, gracias a Andrés por ser tan único y en general gracias a todas y cada una de las personas que he conocido y me han hecho crecer como persona dentro de la carrera. Podría nombraros y dedicaros un párrafo a todos, pero me pasaría de extensión del TFG.

No tengo más que buenas palabras de esta universidad, los amigos que he hecho y las experiencias vividas aquí.

Por todo eso y lo que me dejo por nombrar, muchas gracias.

**RESUMEN**

Los juegos de mesa han sido, son y serán una gran forma de **ocio y entretenimiento** grupal. Algunos de ellos son simples y no necesitan mucha preparación para poder jugarlos de manera óptima. Otros, sin embargo, tienen un **factor estratégico** clave para el desarrollo de la partida, que no solo involucra al propio jugador, sino a todos en la partida.

Los **Colonos de Catán** es un gran ejemplo de este tipo de juegos. Una gran parte de la estrategia utilizada para este juego tiene que ver con la interacción con otros jugadores. El comercio, así como la capacidad de decidir una acción teniendo en mente la situación de los demás jugadores es una de las claves para lograr una buena partida en este juego.

Gracias a esta interacción, es posible adaptar este gran clásico de juego de tablero a una versión digital online y tener una manera más simple de almacenar la información recogida.

Esta información puede ser usada con propósitos de **investigación y desarrollo** en numerosas áreas, desde la Inteligencia Artificial a la Teoría de juegos aplicada.

En este proyecto se plantea usar la plataforma **BoardGame.io** para adaptar los Colonos de Catán de manera fiel y con el objetivo de poder ser usado en un futuro como un medio de recogida de datos para investigación.

**Palabras clave:** Colonos de Catán, Catán, juegos de tablero con interacción entre jugadores, adaptación, BoardGame.io, investigación, JavaScript, React.

**ABSTRACT**

Board games have been, are and will be a great form of **leisure and group entertainment**. Some of them are simpler and do not require much preparation to be able to play them optimally. Others, however, have a key **strategic factor** for the development of the game, which involves not only the player, but everyone in the game.

**Settlers of Catan** is a great example of such a game. A large part of the strategy used for this game has to do with interaction with other players. Trading, as well as the ability to decide on an action with the situation of the other players in mind is one of the keys to be able to win in this game.

Thanks to this interaction, it is possible to adapt this great board game classic to a digital online version and to have a simpler way of storing the information from the match.

This information can be used for **research and development** purposes in numerous areas, from Artificial Intelligence to applied Game t heory.

In this project we plan to use the **BoardGame.io** platform to adapt Settlers of Catan in a faithful way and with the aim of being able to use it in the future as a means of data collection for research.

**Keywords:** Settlers of Catan, Catan, player interaction board games, adaptation, BoardGame.io, investigation, JavaScript, React.

**Índice de Contenidos**

Tabla de contenido

[1. INTRODUCCIÓN - 1 -](#_Toc81741949)

[1.1 Ámbito del proyecto - 1 -](#_Toc81741950)

[1.2 Motivación del Trabajo de Fin de Grado - 1 -](#_Toc81741951)

[1.3 Estructura del Documento - 2 -](#_Toc81741952)

[2. ESTADO DEL ARTE - 4 -](#_Toc81741953)

[2.1. Document Object Model (DOM) - 4 -](#_Toc81741954)

[2.2. Entorno informático - 5 -](#_Toc81741955)

[2.2.1. JavaScript - 5 -](#_Toc81741956)

[2.2.2. React - 6 -](#_Toc81741957)

[2.2.3. Visual Studio Code - 10 -](#_Toc81741958)

[2.3. Plataformas específicas para creación de juegos. - 10 -](#_Toc81741959)

[2.3.1. Vassal - 11 -](#_Toc81741960)

[2.3.2. Tabletop Simulator - 12 -](#_Toc81741961)

[2.3.3. Board Game Arena - 13 -](#_Toc81741962)

[2.3.4. BoardGame.io - 14 -](#_Toc81741963)

[2.4. Trabajos y artículos sobre juegos de tablero - 16 -](#_Toc81741964)

[2.4.1. Neuroeducación y Juego de Mesa - 16 -](#_Toc81741965)

[2.4.2. Aplicación de los Juegos de Mesa como Estrategia Pedagógica - 18 -](#_Toc81741966)

[2.4.3. Teoría de Juegos - 19 -](#_Toc81741967)

[2.5. Inteligencia Artificial en los juegos de tablero - 20 -](#_Toc81741968)

[3. JUEGOS DE TABLERO CON INTERACCIÓN ENTRE JUGADORES - 23 -](#_Toc81741969)

[3.1. Utilidad y relevancia - 23 -](#_Toc81741970)

[3.2. Los Colonos de Catán - 23 -](#_Toc81741971)

[3.2.1. Historia - 23 -](#_Toc81741972)

[3.2.2. Reglas básicas - 24 -](#_Toc81741973)

[3.2.3. Estrategias posibles - 30 -](#_Toc81741974)

[3.3. Otros juegos de la misma área - 30 -](#_Toc81741975)

[3.3.1. Istambul - 31 -](#_Toc81741976)

[3.3.2. Agrícola - 31 -](#_Toc81741977)

[3.3.3. Alta Tensión - 32 -](#_Toc81741978)

[3.3.4. Isle of Skye - 33 -](#_Toc81741979)

[3.3.5. Carcassonne - 34 -](#_Toc81741980)

[3.3.6. Aventureros al Tren - 35 -](#_Toc81741981)

[4. OBJETIVOS - 37 -](#_Toc81741982)

[5. ANÁLISIS DEL SISTEMA - 38 -](#_Toc81741983)

[5.1. Casos de Uso - 38 -](#_Toc81741984)

[5.2. Requisitos de Usuario - 46 -](#_Toc81741985)

[5.2.1. Requisitos de Capacidad - 48 -](#_Toc81741986)

[5.2.2. Requisitos de Restricción - 57 -](#_Toc81741987)

[5.3. Requisitos de Sistema - 60 -](#_Toc81741988)

[5.3.1. Requisitos Funcionales - 61 -](#_Toc81741989)

[5.3.2. Requisitos no Funcionales - 82 -](#_Toc81741990)

[5.4. Matriz de trazabilidad - 84 -](#_Toc81741991)

[6. DISEÑO DEL SISTEMA - 89 -](#_Toc81741992)

[6.1. Diagrama de clases - 89 -](#_Toc81741993)

[6.2. Entorno de desarrollo - 90 -](#_Toc81741994)

[6.3. Diseño de funciones - 91 -](#_Toc81741995)

[6.4. Desarrollo software e interfaz - 91 -](#_Toc81741996)

[6.4.1. Tablero - 92 -](#_Toc81741997)

[6.4.2. Barra de acciones - 95 -](#_Toc81741998)

[6.4.3. Barra de cartas - 98 -](#_Toc81741999)

[6.4.4. Tabla de estado - 99 -](#_Toc81742000)

[6.5. Manual de usuario - 101 -](#_Toc81742001)

[6.5.1. Descarga e instalación del software - 101 -](#_Toc81742002)

[6.5.2. Fase inicial del juego - 103 -](#_Toc81742003)

[6.5.3. Turno del jugador - 104 -](#_Toc81742004)

[6.5.4. Fases especiales - 104 -](#_Toc81742005)

[7. BATERÍA DE PRUEBAS - 105 -](#_Toc81742006)

[7.1. Pruebas - 105 -](#_Toc81742007)

[7.2. Matriz de trazabilidad - 118 -](#_Toc81742008)

[8. PLAN DE PROYECTO - 122 -](#_Toc81742009)

[8.1. Gestión del proyecto - 122 -](#_Toc81742010)

[8.2. Presupuesto del proyecto - 123 -](#_Toc81742011)

[8.2.1. Costes de personal - 123 -](#_Toc81742012)

[8.2.2. Costes de software - 124 -](#_Toc81742013)

[8.2.3. Costes de hardware - 124 -](#_Toc81742014)

[8.2.4. Costes totales - 125 -](#_Toc81742015)

[9. CONCLUSIONES - 127 -](#_Toc81742016)

[9.1. Conclusiones del proyecto - 127 -](#_Toc81742017)

[9.2. Conclusiones personales - 127 -](#_Toc81742018)

[10. POSIBLES AMPLIACIONES DEL PROYECTO - 129 -](#_Toc81742019)

[11. MARCO REGULADOR Y ENTORNO SOCIOECONÓMICO - 130 -](#_Toc81742020)

[11.1. Marco regulador - 130 -](#_Toc81742021)

[11.1.1. Análisis de la legislación aplicable - 130 -](#_Toc81742022)

[11.1.2. Propiedad intelectual - 131 -](#_Toc81742023)

[11.2. Entorno socioeconómico - 131 -](#_Toc81742024)

[12. BIBILIOGRAFÍA - 133 -](#_Toc81742025)

[13. FUENTES - 135 -](#_Toc81742026)

[ANEXO A: GLOSARIO - 137 -](#_Toc81742027)

[ANEXO B: SUMMARY - 139 -](#_Toc81742028)

[B.1. Introduction - 139 -](#_Toc81742029)

[B.1.1. Scope of the Project - 139 -](#_Toc81742030)

[B.1.2. Motivation - 139 -](#_Toc81742031)

[B.2. State of Art - 139 -](#_Toc81742032)

[B.2.1. Document Object Model (DOM) - 140 -](#_Toc81742033)

[B.2.2. Computer Environment - 140 -](#_Toc81742034)

[B.2.3. Specific platforms for computer games develpoment - 140 -](#_Toc81742035)

[B.2.4. Articles about board games - 141 -](#_Toc81742036)

[B.3. Board game with interaction between players - 142 -](#_Toc81742037)

[B.3.1. Usefulness and relevance - 142 -](#_Toc81742038)

[B.3.2. Settlers of Catan - 143 -](#_Toc81742039)

[B.3.3. Other games in the área - 143 -](#_Toc81742040)

[B.4. Targets - 144 -](#_Toc81742041)

[B.5. System analysis - 145 -](#_Toc81742042)

[B.6. System design - 145 -](#_Toc81742043)

[B.6.1 Class diagram - 145 -](#_Toc81742044)

[B.6.2. Development environment - 146 -](#_Toc81742045)

[B.6.3. Function design - 146 -](#_Toc81742046)

[B.6.4. Software development and interface - 147 -](#_Toc81742047)

[B.7. Battery of tests - 148 -](#_Toc81742048)

[B.8. Project Management - 148 -](#_Toc81742049)

[B.9. Conclusions - 149 -](#_Toc81742050)

[B.9.1. Project conclusions - 149 -](#_Toc81742051)

[B.9.2. Personal conclusions - 149 -](#_Toc81742052)

[B.10. Posible extensions of the project - 150 -](#_Toc81742053)

**Índice de Figuras**

[Ilustración 1. Ejemplo árbol DOM - 4 -](#_Toc81742054)

[Ilustración 2. JavaScript - 5 -](#_Toc81742055)

[Ilustración 3. React - 6 -](#_Toc81742056)

[Ilustración 4. Ciclo de vida obsoleto - 8 -](#_Toc81742057)

[Ilustración 5. Ciclo de vida nuevo - 9 -](#_Toc81742058)

[Ilustración 6. Visual Studio Code - 10 -](#_Toc81742059)

[Ilustración 7. Vassal - 11 -](#_Toc81742060)

[Ilustración 8. Interfaz Vassal - 12 -](#_Toc81742061)

[Ilustración 9. Tabletop Simulator - 12 -](#_Toc81742062)

[Ilustración 10. Board Game Arena - 13 -](#_Toc81742063)

[Ilustración 11. BGA tablero - 14 -](#_Toc81742064)

[Ilustración 12. BoardGame.io - 15 -](#_Toc81742065)

[Ilustración 13. Interfaz BoardGame.io - 15 -](#_Toc81742066)

[Ilustración 14. Neuroeduación - 16 -](#_Toc81742067)

[Ilustración 15. Tipos de Inteligencia (Howard Gardner) - 17 -](#_Toc81742068)

[Ilustración 16. Corporación Universitaria Iberoamericana - 18 -](#_Toc81742069)

[Ilustración 17. Tipos de Teoría de Juegos. - 19 -](#_Toc81742070)

[Ilustración 18. Gary Kasparov vs IA (1996) - 20 -](#_Toc81742071)

[Ilustración 19. Aprendizaje automático - 21 -](#_Toc81742072)

[Ilustración 20. Klaus Teuber - 24 -](#_Toc81742073)

[Ilustración 21. Cartas de Recurso - 25 -](#_Toc81742074)

[Ilustración 22. Tablero Catán - 26 -](#_Toc81742075)

[Ilustración 23. Cartas de Desarrollo - 28 -](#_Toc81742076)

[Ilustración 24. Puerto 2x1 Ladrillo - 29 -](#_Toc81742077)

[Ilustración 25. Juego Istambul - 31 -](#_Toc81742078)

[Ilustración 26. Juego Agrícola - 32 -](#_Toc81742079)

[Ilustración 27. Juego Alta Tensión - 33 -](#_Toc81742080)

[Ilustración 28. Juego Isle of Skye - 34 -](#_Toc81742081)

[Ilustración 29. Juego Carcassonne - 35 -](#_Toc81742082)

[Ilustración 30. Juego Aventureros al Tren - 36 -](#_Toc81742083)

[Ilustración 31. Diagrama de clases - 89 -](#_Toc81742084)

[Ilustración 32. Tablero Catán - 92 -](#_Toc81742085)

[Ilustración 33. Diagrama de casillas - 93 -](#_Toc81742086)

[Ilustración 34. Imagen Círculo - 93 -](#_Toc81742087)

[Ilustración 35. Círculos de vértice - 94 -](#_Toc81742088)

[Ilustración 36. Barra de acciones - 95 -](#_Toc81742089)

[Ilustración 37. Pop-Up Robar - 96 -](#_Toc81742090)

[Ilustración 38. Pop-Up Comercio - 97 -](#_Toc81742091)

[Ilustración 39. Barra de recursos - 98 -](#_Toc81742092)

[Ilustración 40. Barra de cartas de desarrollo - 98 -](#_Toc81742093)

[Ilustración 41. Pop-Up Invento - 99 -](#_Toc81742094)

[Ilustración 42. Pop-Up Monopolio - 99 -](#_Toc81742095)

[Ilustración 43. Tabla de estado - 100 -](#_Toc81742096)

[Ilustración 44. Sitio web Node.js - 102 -](#_Toc81742097)

[Ilustración 45. Captura CMD proyecto - 103 -](#_Toc81742098)

[Ilustración 46. Aviso preparación - 104 -](#_Toc81742099)

[Ilustración 47. Plan de proyecto de software - 122 -](#_Toc81742100)

[Ilustración 48. Plan de proyecto de documentación - 123 -](#_Toc81742101)

**Índice de Tablas**

[Tabla 1. Plantilla Casos de Uso - 38 -](#_Toc81736083)

[Tabla 2. Caso de Uso 01 - 39 -](#_Toc81736084)

[Tabla 3. Caso de Uso 02 - 39 -](#_Toc81736085)

[Tabla 4. Caso de Uso 03 - 40 -](#_Toc81736086)

[Tabla 5. Caso de Uso 04 - 40 -](#_Toc81736087)

[Tabla 6. Caso de Uso 05 - 41 -](#_Toc81736088)

[Tabla 7. Caso de Uso 06 - 41 -](#_Toc81736089)

[Tabla 8. Caso de Uso 07 - 42 -](#_Toc81736090)

[Tabla 9. Caso de Uso 08 - 42 -](#_Toc81736091)

[Tabla 10. Caso de Uso 09 - 43 -](#_Toc81736092)

[Tabla 11. Caso de Uso 10 - 43 -](#_Toc81736093)

[Tabla 12. Caso de Uso 11 - 44 -](#_Toc81736094)

[Tabla 13. Caso de Uso 12 - 44 -](#_Toc81736095)

[Tabla 14. Caso de Uso 13 - 45 -](#_Toc81736096)

[Tabla 15. Caso de Uso 14 - 45 -](#_Toc81736097)

[Tabla 16. Caso de Uso 15 - 46 -](#_Toc81736098)

[Tabla 17. Caso de Uso 15 - 46 -](#_Toc81736099)

[Tabla 18. Plantilla Requisito de Usuario - 47 -](#_Toc81736100)

[Tabla 19. Requisito RQC-01 - 48 -](#_Toc81736101)

[Tabla 20. Requisito RQC-02 - 48 -](#_Toc81736102)

[Tabla 21. Requisito RQC-03 - 49 -](#_Toc81736103)

[Tabla 22. Requisito RQC-04 - 49 -](#_Toc81736104)

[Tabla 23. Requisito RQC-05 - 50 -](#_Toc81736105)

[Tabla 24. Requisito RQC-06 - 50 -](#_Toc81736106)

[Tabla 25. Requisito RQC-07 - 50 -](#_Toc81736107)

[Tabla 26. Requisito RQC-08 - 51 -](#_Toc81736108)

[Tabla 27. Requisito RQC-09 - 51 -](#_Toc81736109)

[Tabla 28. Requisito RQC-10 - 52 -](#_Toc81736110)

[Tabla 29. Requisito RQC-11 - 52 -](#_Toc81736111)

[Tabla 30. Requisito RQC-12 - 52 -](#_Toc81736112)

[Tabla 31. Requisito RQC-13 - 53 -](#_Toc81736113)

[Tabla 32. Requisito RQC-14 - 53 -](#_Toc81736114)

[Tabla 33. Requisito RQC-15 - 54 -](#_Toc81736115)

[Tabla 34. Requisito RQC-16 - 54 -](#_Toc81736116)

[Tabla 35. Requisito RQC-17 - 54 -](#_Toc81736117)

[Tabla 36. Requisito RQC-18 - 55 -](#_Toc81736118)

[Tabla 37. Requisito RQC-19 - 55 -](#_Toc81736119)

[Tabla 38. Requisito RQC-20 - 56 -](#_Toc81736120)

[Tabla 39. Requisito RQC-21 - 56 -](#_Toc81736121)

[Tabla 40. Requisito RQC-22 - 56 -](#_Toc81736122)

[Tabla 41. Requisito RQC-23 - 57 -](#_Toc81736123)

[Tabla 42. Requisito RQC-24 - 57 -](#_Toc81736124)

[Tabla 43. Requisito RQC-25 - 57 -](#_Toc81736125)

[Tabla 44. Requisito RQR-01 - 58 -](#_Toc81736126)

[Tabla 45. Requisito RQR-02 - 58 -](#_Toc81736127)

[Tabla 46. Requisito RQR-03 - 59 -](#_Toc81736128)

[Tabla 47. Requisito RQR-04 - 59 -](#_Toc81736129)

[Tabla 48. Requisito RQR-05 - 59 -](#_Toc81736130)

[Tabla 49. Requisito RQR-06 - 60 -](#_Toc81736131)

[Tabla 50. Requisito RQR-07 - 60 -](#_Toc81736132)

[Tabla 51. Plantilla Requisito de Sistema - 61 -](#_Toc81736133)

[Tabla 52. Requisito RQF-01 - 62 -](#_Toc81736134)

[Tabla 53. Requisito RQF-02 - 62 -](#_Toc81736135)

[Tabla 54. Requisito RQF-03 - 62 -](#_Toc81736136)

[Tabla 55. Requisito RQF-04 - 63 -](#_Toc81736137)

[Tabla 56. Requisito RQF-05 - 63 -](#_Toc81736138)

[Tabla 57. Requisito RQF-06 - 64 -](#_Toc81736139)

[Tabla 58. Requisito RQF-07 - 64 -](#_Toc81736140)

[Tabla 59. Requisito RQF-08 - 64 -](#_Toc81736141)

[Tabla 60. Requisito RQF-09 - 65 -](#_Toc81736142)

[Tabla 61. Requisito RQF-10 - 65 -](#_Toc81736143)

[Tabla 62. Requisito RQF-11 - 66 -](#_Toc81736144)

[Tabla 63. Requisito RQF-12 - 66 -](#_Toc81736145)

[Tabla 64. Requisito RQF-13 - 66 -](#_Toc81736146)

[Tabla 65. Requisito RQF-14 - 67 -](#_Toc81736147)

[Tabla 66. Requisito RQF-15 - 67 -](#_Toc81736148)

[Tabla 67. Requisito RQF-16 - 68 -](#_Toc81736149)

[Tabla 68. Requisito RQF-17 - 68 -](#_Toc81736150)

[Tabla 69. Requisito RQF-18 - 68 -](#_Toc81736151)

[Tabla 70. Requisito RQF-19 - 69 -](#_Toc81736152)

[Tabla 71. Requisito RQF-20 - 69 -](#_Toc81736153)

[Tabla 72. Requisito RQF-21 - 69 -](#_Toc81736154)

[Tabla 73. Requisito RQF-22 - 70 -](#_Toc81736155)

[Tabla 74. Requisito RQF-23 - 70 -](#_Toc81736156)

[Tabla 75. Requisito RQF-24 - 71 -](#_Toc81736157)

[Tabla 76. Requisito RQF-25 - 71 -](#_Toc81736158)

[Tabla 77. Requisito RQF-26 - 71 -](#_Toc81736159)

[Tabla 78. Requisito RQF-27 - 72 -](#_Toc81736160)

[Tabla 79. Requisito RQF-28 - 72 -](#_Toc81736161)

[Tabla 80. Requisito RQF-29 - 73 -](#_Toc81736162)

[Tabla 81. Requisito RQF-30 - 73 -](#_Toc81736163)

[Tabla 82. Requisito RQF-31 - 73 -](#_Toc81736164)

[Tabla 83. Requisito RQF-32 - 74 -](#_Toc81736165)

[Tabla 84. Requisito RQF-33 - 74 -](#_Toc81736166)

[Tabla 85. Requisito RQF-34 - 75 -](#_Toc81736167)

[Tabla 86. Requisito RQF-35 - 75 -](#_Toc81736168)

[Tabla 87. Requisito RQF-36 - 75 -](#_Toc81736169)

[Tabla 88. Requisito RQF-37 - 76 -](#_Toc81736170)

[Tabla 89. Requisito RQF-38 - 76 -](#_Toc81736171)

[Tabla 90. Requisito RQF-39 - 77 -](#_Toc81736172)

[Tabla 91. Requisito RQF-40 - 77 -](#_Toc81736173)

[Tabla 92. Requisito RQF-41 - 77 -](#_Toc81736174)

[Tabla 93. Requisito RQF-42 - 78 -](#_Toc81736175)

[Tabla 94. Requisito RQF-43 - 78 -](#_Toc81736176)

[Tabla 95. Requisito RQF-44 - 79 -](#_Toc81736177)

[Tabla 96. Requisito RQF-45 - 79 -](#_Toc81736178)

[Tabla 97. Requisito RQF-46 - 79 -](#_Toc81736179)

[Tabla 98. Requisito RQF-47 - 80 -](#_Toc81736180)

[Tabla 99. Requisito RQF-48 - 80 -](#_Toc81736181)

[Tabla 100. Requisito RQF-49 - 80 -](#_Toc81736182)

[Tabla 101. Requisito RQF-50 - 81 -](#_Toc81736183)

[Tabla 102. Requisito RQF-51 - 81 -](#_Toc81736184)

[Tabla 103. Requisito RQF-52 - 82 -](#_Toc81736185)

[Tabla 104. Requisito RQF-53 - 82 -](#_Toc81736186)

[Tabla 105. Requisito RQN-01 - 82 -](#_Toc81736187)

[Tabla 106. Requisito RQN-02 - 83 -](#_Toc81736188)

[Tabla 107. Requisito RQN-03 - 83 -](#_Toc81736189)

[Tabla 108. Requisito RQN-04 - 84 -](#_Toc81736190)

[Tabla 109. Requisito RQN-05 - 84 -](#_Toc81736191)

[Tabla 110. Requisito RQN-06 - 84 -](#_Toc81736192)

[Tabla 111. Matriz de trazabilidad - 88 -](#_Toc81736193)

[Tabla 112. Plantilla Pruebas - 105 -](#_Toc81736194)

[Tabla 113. Prueba 01 - 106 -](#_Toc81736195)

[Tabla 114. Prueba 02 - 106 -](#_Toc81736196)

[Tabla 115. Prueba 03 - 107 -](#_Toc81736197)

[Tabla 116. Prueba 04 - 107 -](#_Toc81736198)

[Tabla 117. Prueba 05 - 108 -](#_Toc81736199)

[Tabla 118. Prueba 06 - 108 -](#_Toc81736200)

[Tabla 119. Prueba 07 - 109 -](#_Toc81736201)

[Tabla 120. Prueba 08 - 109 -](#_Toc81736202)

[Tabla 121. Prueba 09 - 109 -](#_Toc81736203)

[Tabla 122. Prueba 10 - 110 -](#_Toc81736204)

[Tabla 123. Prueba 11 - 110 -](#_Toc81736205)

[Tabla 124. Prueba 12 - 111 -](#_Toc81736206)

[Tabla 125. Prueba 13 - 111 -](#_Toc81736207)

[Tabla 126. Prueba 14 - 111 -](#_Toc81736208)

[Tabla 127. Prueba 15 - 112 -](#_Toc81736209)

[Tabla 128. Prueba 16 - 112 -](#_Toc81736210)

[Tabla 129. Prueba 17 - 113 -](#_Toc81736211)

[Tabla 130. Prueba 18 - 113 -](#_Toc81736212)

[Tabla 131. Prueba 19 - 113 -](#_Toc81736213)

[Tabla 132. Prueba 20 - 114 -](#_Toc81736214)

[Tabla 133. Prueba 21 - 114 -](#_Toc81736215)

[Tabla 134. Prueba 22 - 115 -](#_Toc81736216)

[Tabla 135. Prueba 23 - 115 -](#_Toc81736217)

[Tabla 136. Prueba 24 - 116 -](#_Toc81736218)

[Tabla 137. Prueba 25 - 116 -](#_Toc81736219)

[Tabla 138. Prueba 26 - 116 -](#_Toc81736220)

[Tabla 139. Prueba 27 - 117 -](#_Toc81736221)

[Tabla 140. Prueba 28 - 117 -](#_Toc81736222)

[Tabla 141. Prueba 29 - 118 -](#_Toc81736223)

[Tabla 142. Prueba 30 - 118 -](#_Toc81736224)

[Tabla 143. Matriz de trazabilidad de pruebas - 121 -](#_Toc81736225)

[Tabla 144. Costes de Personal - 124 -](#_Toc81736226)

[Tabla 145. Costes de Software - 124 -](#_Toc81736227)

[Tabla 146. Costes de Hardware - 125 -](#_Toc81736228)

[Tabla 147. Costes indirectos - 125 -](#_Toc81736229)

[Tabla 148. Costes totales - 126 -](#_Toc81736230)

# INTRODUCCIÓN

## Ámbito del proyecto

Este proyecto se enmarca en el ámbito de creación o **adaptación de juegos de tablero** a un entorno digitalizado.

El Catán, o en su nombre completo, Los Colonos de Catán, es un juego de mesa de 2 a 4 jugadores en su versión estándar, cuyo objetivo es conseguir reunir 10 puntos de victoria a través de la construcción, comercio y obtención de logros.

Se trata de un **juego competitivo estratégico**, puesto que la toma de decisiones con el entorno y con los otros jugadores es clave para el desarrollo de este.

Este proyecto tiene como objetivo base la incorporación de este juego tradicional, así como sus mecánicas e interfaz, a un entorno virtual a través de la plataforma Boardgame.io, que permite la creación de juegos a través de un entorno basado en JavaScript y con una estructura base para el desarrollo del juego.

A su vez, este proyecto puede ampliarse a un ámbito de **análisis de datos e investigación**, así como de agentes automáticos, de los cuales se tratará más adelante en el documento.

## Motivación del Trabajo de Fin de Grado

En la actualidad, vivimos en un mundo cada vez más digitalizado, dónde la gran mayoría de juegos o fuentes de entretenimiento del día a día son a través de una pantalla. Sin embargo, los juegos de mesa tradicionales aún perduran y siguen siendo tan populares como antaño.

Más aun, con la llegada del Covid-19 y la situación de cuarentena y asilamiento que tuvimos que superar, el uso de este tipo de juegos aumentó y la popularidad de juegos de estrategia entretenidos se disparó, como es el caso del Catán.

El problema con los juegos de mesa basados en tablero es que no siempre se puede jugar, bien por falta de espacio para colocarlo, o bien debido a que los jugadores están físicamente en lugares distintos.

Por ello, y dentro de una motivación puramente pragmática, para este tipo de situaciones dónde se quiere disfrutar de este juego de mesa, pero no se dispone de las condiciones necesarias, se ha planteado la **adaptación a un entorno virtual** dónde se pueda disfrutar del juego únicamente necesitando acceso a internet.

Por otro lado, la motivación principal de este trabajo ha sido usar el Catán como **plataforma de investigación** para obtener datos que podrán ser usados en un futuro en otros trabajos. Cada una de las acciones que toma un jugador y el momento y circunstancia en el que las realiza pueden ser almacenadas y transformadas en información útil para obtener un modelo óptimo de juego, así como un entendimiento del razonamiento lógico del cerebro humano antes diversas situaciones. También hay un componente clave que es la aplicación de los juegos de mesa a la **educación o aprendizaje**.

Existen diversos artículos y trabajos que hablan de este concepto aplicado al Catán. En ellos se refieren a esta área como Gamificación, que es una técnica orientada al aprendizaje que busca trasladar el reglamento y funcionamiento de un juego al ámbito educativo, a través del establecimiento de objetivos y recompensas.

Además, el hecho de que un juego de estrategia se programe digitalmente, permite una mayor facilidad de la obtención de datos durante la partida. De esta forma, analizando las acciones de los jugadores y su resultado, se podría, a través de algunas herramientas, llegar a crear una guía o un agente capaz de jugar de la **manera óptima** posible.

## Estructura del Documento

El documento está organizado de la siguiente forma:

1. **Introducción:** Se describe brevemente el área al que pertenece el proyecto, así como la razón por la que se ha elegido el mismo.
2. **Estado del arte:** Se explica factores del entorno del desarrollo del proyecto, teoría básica para entenderlo, plataformas similares de desarrollo y algunos artículos sobre el área que cubre el trabajo.
3. **Juegos de tablero con interacción entre jugadores:** Se ahonda en el tipo de juego que es el Catán y la relevancia de este, describiendo algunos ejemplos de juegos similares y explicando el Catán individualmente.
4. **Objetivos:** Se detallan todos los objetivos que tiene este trabajo de manera jerárquica.
5. **Análisis del sistema:** Se analiza el proyecto definiéndolo a través de casos de uso y los diferentes tipos de requisitos.
6. **Diseño del sistema:** Se explica cómo ha sido el desarrollo del trabajo en cada una de sus partes, detallando también el funcionamiento de este y un manual de usuario para comprender su uso.
7. **Batería de pruebas:** Se definen las pruebas que se han realizado para comprobar el funcionamiento del software.
8. **Plan de proyecto:** Se detalla cómo ha sido realizada la gestión del proyecto y cuál sería su presupuesto en caso de ser una oferta profesional.
9. **Conclusiones:** Se explican las conclusiones a las que se ha llegado y los resultados obtenidos en el desarrollo del proyecto.
10. **Posibles ampliaciones del proyecto:** Se enumeran posibles líneas a mejorar y ampliar dentro del proyecto.
11. **Marco regulador y entorno socioeconómico:** Se explican las limitaciones legales y de propiedad intelectual del proyecto, así como el factor social que tiene.
12. **Bibliografía**: Se enumeran las referencias utilizadas en el desarrollo del documento.
13. **Fuentes**: Se enumeran las fuentes utilizadas durante el desarrollo del software.

# ESTADO DEL ARTE

## 2.1. Document Object Model (DOM)

Antes de poder pasar a hablar acerca de los elementos y herramientas utilizados para el trabajo, es necesario introducir el DOM, puesto que es la base sobre la que se sustenta lenguajes como HTML o JavaScript **[1].**

El DOM es una **estructura de objetos** que se genera a través del navegador cuando se carga una página web. Es modificable a través de lenguajes como JavaScript lo que permite una página dinámica con un aspecto personalizable.

Este tipo de modelo funciona a través de una **jerarquía** establecida entre los objetos que lo forman, es decir, unos objetos dependen de otros formando lo que se conocen como árboles DOM.

Diagrama

Descripción generada automáticamente

Ilustración 1. Ejemplo árbol DOM

Estos objetos representan y modelizan a todos los elementos de la página web, ya sea la ventana del propio navegador, datos internos, historial y cualquier párrafo o texto que incorpore la página.

Es W3C (Consorcio World Wide Web) quien se encarga de definir las especificaciones que incorpora el DOM, por lo que ofrece soporte en las aplicaciones. El DOM permite la modificación de páginas HTML, así como con formato XML.

Este modelo se ha ido mejorando a través de los años y de la mejora de los navegadores, implementando cada vez más características. Es por esto por lo que algunos navegadores implementan un nivel distinto de DOM que otro más modernos o de diferente estructura, lo que produce algunos **fallos de compatibilidad** entre navegadores.

## 2.2. Entorno informático

En el siguiente apartado se destallarán a través de que **lenguajes, programas y herramientas** se ha desarrollado el proyecto, así como sus principales usos en la actualidad.

### 2.2.1. JavaScript

El código está **programado usando Javascript**, aunque no la versión plana, si no la orienta a React, es decir JSX o JavaScript Syntax Extension, aunque eso se detallará más adelante.

JavaScript fue creado en 1995 y fue diseñado por Netscape Communications y la Fundación Mozilla **[2].**

JavaScript es uno de los lenguajes de programación **interpretado**, es decir, que sus instrucciones se leen y ejecutan una por una a tiempo real, en este caso a través del navegador desde donde se ejecute.

Texto, Escala de tiempo

Descripción generada automáticamente

Ilustración 2. JavaScript

Permite realizar actividades tanto simples como complejas y eso lo hace hoy en día como uno de los **estándares** para el diseño y el desarrollo de **páginas web** **[3].**

Este lenguaje goza de su popularidad debido a la gran interactividad que permite con los elementos de la página web, permitiendo así crear páginas dinámicas y multifuncionales.

La **función básica** de JavaScript es permitir a una página web estática un mayor índice de interactividad, con el objetivo de mejorar la experiencia del usuario y la facilidad del uso de la página. A su vez, una de las grandes ventajas de este lenguaje es que el peso total de la página no se ve aumentado drásticamente, lo que permite la creación de páginas con una carga relativamente **rápida**.

Esta facilidad para implementar JavaScript, así como la cantidad de funcionalidades que soporta, ha hecho que muchas páginas web que sirven como plantilla para crear videojuegos usen este lenguaje para su desarrollo, como es el caso de BoardGame.io.

### 2.2.2. React

React es una **librería** relativamente nueva (se creó en 2010) de JavaScript, creada por Facebook y que permite la mejora de las páginas web a través de la creación y montaje de componentes interactivos y reutilizables.

Actualmente es una de las bibliotecas más usadas por los desarrolladores de páginas web y ha sido usada por ejemplo para el desarrollo de Facebook o Instagram, esta última programada enteramente con React **[4].**

Imagen que contiene Logotipo

Descripción generada automáticamente

Ilustración 3. React

Una de las grandes ventajas con las que cuenta React es su capacidad de desarrollar no solo la parte del **cliente**, sino también la del **servidor**, permitiendo así un mayor control y libertad sobre el manejo de la página web.

Para poder explicar bien como se usa y para qué sirve React, es necesario hablar de los componentes y su funcionalidad. Un componente es una de las piezas que forman la interfaz del usuario, son independientes entre sí y se programan por separado en el código **[5].**

**Los componentes** se pueden crear a través de funciones o a través de las clases de JavaScript, introducidas en ECMAScript en el año 2015. A cada componente el usuario puede pasarle los argumentos que necesite, y el componente los recibirá en forma de objeto a través de un argumento llamado “props”.

Un componente puede estar compuesto de otros componentes a la hora de la renderización, de tal forma que un componente llamado por ejemplo “oración” sea básicamente una llamada a varios componentes “palabra”. Esto permite que se pueda crear cierta jerarquía que suele desembocar en un componente global llamado App, pudiendo tener un alto **grado de anidamiento** para lograr códigos simples e intuitivos.

Una de las pocas restricciones con las que cuenta React es que un componente no puede modificar el valor de ninguno de los argumentos recibidos a través de props ya que son solo de lectura, considerándose así funciones puras **[6].**

Uno de los conceptos más interesantes de React son los **ciclos de vida** de sus componentes, que en este caso representan cada una de las etapas que tiene ese componente, desde su creación hasta su destrucción, con el objetivo de optimizar recursos y eficiencia.

React es una librería que se **actualiza constantemente** y hasta la versión 16.2 se utilizaba otro ciclo de vida distinto al que se usa actualmente (va por la versión 17.0.2). En este caso 3 de sus etapas fueron “deprecadas” y sustituidas por otras dos nuevas, esto quiere decir que se pueden seguir usando dichas etapas, pero no se aconseja puesto que las actualizaciones no cuentan con ellas **[7].**

Diagrama

Descripción generada automáticamente

Ilustración 4. Ciclo de vida obsoleto

En la ilustración 15 se muestran en verde los estados que siguen vigentes y en rojo lo que están obsoletos. A continuación, se explicarán brevemente para un mayor entendimiento:

* **componentWillMount (obs):** este método se utilizaba previamente al montaje del componente y servía como inicialización síncrona del estado o carga de los datos
* **componentDidMount:** esta función se ejecuta de manera posterior al método render, por lo que todos los elementos DOM ya son accesibles. Su objetivo es la carga de datos por parte del servidor o ejecutar operaciones que necesiten algún elemento del DOM.
* **componentWillReceiveProps (obs):** esta función sirve para actualizar la lista de props de un componente en caso de que cambien. Recibe por parámetro las nuevas propsy se ejecuta siempre que un componente que ya ha sido montado obtenga nuevas propiedades.
* **shouldComponentUpdate:** esta función es ejecutada de manera previa al método render, de tal forma que si el estado o las props del componente son modificadas se pasará a ejecutar el render para su actualización.
* **componentWillUpdate (obs):** esta función obsoleta se ejecuta una vez la función shouldComponentUpdate recibe nuevas props o cambios en el estado. Se encarga de preparar al componente para la actualización, justo antes de llamar al render.
* **render:** se trata de la única función de ciclo de vida que es obligatoria ya que es la responsable de mostrar como el componente se va a ver en pantalla a través de su renderización. Esta función puede retornar diferentes tipos de valores, principalmente elementos (p.e. componentes), arrays, portales o tipos de datos (strings, ints, boolean…).
* **componentDidUpdate:** este método es ejecutado tras la función render, aunque únicamente si se ha modificado el estado del componente. Recibe por parámetro las props y el estado antiguo y comprueba si se ha actualizado o no.
* **componentWillUnmount:** esta función sirve para dejar todo limpio y cerrado previamente a la destrucción del componente, se podría decir que es la opuesta a componentWillMount.

Como se ha comentado antes, este ciclo de componentes está desactualizado. Actualmente se usa el siguiente ciclo:

Diagrama

Descripción generada automáticamente

Ilustración 5. Ciclo de vida nuevo

Este nuevo ciclo contiene 2 funciones que sustituyes y mejoran a las ya mencionadas deprecadas. Estas dos funciones son las siguientes:

* **getDerivedStateFromProps:** Este método únicamente se llama si se da el caso de que el estado del componente dependa directamente de las props. Sus argumentos son el estado y las props, y con ellos deberá devolver el objeto que pasará a ser el nuevo estado de dicho componente.
* **getSnapshotBeforeUpdate:** Esta función permite capturar el estado del componente justo en el momento previo a su renderización. El return de esta función se pasa a su vez como parámetro a componentDidUpdate y se usa para poder determinar algunos datos del DOM y si es necesario ejercer alguna acción posterior al render.

### 2.2.3. Visual Studio Code

La totalidad del código del proyecto ha sido escrito a través de Visual Studio Code. Se trata de un **editor de texto** desarrollado por Microsoft para diversos sistemas operativos (entre ellos Windows o Linux) en 2015 **[8].**

Icono

Descripción generada automáticamente

Ilustración 6. Visual Studio Code

Se trata de un editor que viene equipado con **soporte íntegro** para JavaScript, TypeScript y Node.js, aunque también cuenta con un gran archivo de librerías para otros numerosos lenguajes **[9].**

Actualmente es el editor de texto más utilizado para programar Javascript y Node.js por las facilidades que trae y la cantidad de librerías útiles para el usuario que posee.

## 2.3. Plataformas específicas para creación de juegos.

La **programación de videojuegos** ha sido un área que con los años ha pasado de estar monopolizada por unos pocos informáticos a ser un mundo mucho más abierto dónde prácticamente cualquiera con unas nociones básicas y algunos tutoriales puede montar su propio videojuego o versionar uno ya existente.

Con el paso del tiempo salen a la luz nuevas páginas o sitios web que permiten a los usuarios crear juegos usando recursos de la propia web o con documentación y tutoriales básicos para que el creador desarrolle su juego de manera **personalizada**.

Normalmente estos juegos son programados a través de JavaScript o TypeScript, ayudándose de librerías como React o entornos como Node.js.

Dentro de este apartado se resumirán algunas de las páginas principales actualmente, y el motivo del uso de BoardGane.io para este trabajo.

### 2.3.1. Vassal

Vassal es un **motor** creado para la programación y posterior uso online de diferentes juegos basados en turnos y pensados para jugar PvP (Person vs Person), es decir, únicamente entre humanos **[10].**

Forma, Flecha

Descripción generada automáticamente

Ilustración 7. Vassal

Este motor permite la jugabilidad a **tiempo real** mediante conexión a internet o vía email, permitiendo cambiar entre estos durante el transcurso de la partida.

Los juegos se programan 100% a través de Java y algunas de las características más llamativas de Vassal son su facilidad para **personalizar mapas y ajustes**, así como su funcionalidad drag-drop (arrastrar objetos con el ratón). Por otro lado, cuenta con funcionalidades útiles para investigación como la posibilidad de registrar todos los movimientos o una interfaz simple, retro y fácil de entender.

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 8. Interfaz Vassal

A la hora de elegir una plataforma para poder programar el Catán, Vassal fue una de las opciones que más se barajó debido a su simplicidad y la cantidad de tutoriales disponibles por internet. Sin embargo, se decidió dejar de lado principalmente debido a que el lenguaje de programación es Java y el proyecto se pensó desde un inició para React y JavaScript.

### 2.3.2. Tabletop Simulator

Tabletop Simulator es un **juego** en sí mismo. Se trata de un juego desarrollado por Berserk Games en 2015 (actualmente disponible en Steam) que permite la creación y el uso de otros juegos de tablero mediante un entorno de **simulación en 3D**, dónde los jugadores pueden moverse con una cámara libre e interactuar con las piezas **[11].**

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 9. Tabletop Simulator

La característica fundamental de este juego es el conocido como **Steam Workshop**, que se trata de un sistema pensado para que varios usuarios puedan colaborar en la creación de un juego y que permite la instalación sencilla de mods (extensión de un videojuego que cambia alguna de sus características) a través de un simple click.

Al igual que con Vassal, Tabletop Simulator también se pensó como un posible medio para programar el juego, sin embargo, en este caso, fue descartado rápidamente debido a que prácticamente toda la creación del juego (salvo los scripts y alguna cosa más) se hace a través de una interfaz preprogramada dónde no es necesario escribir código, y para este proyecto se buscaba crear los cimientos del juego desde la nada.

### 2.3.3. Board Game Arena

Board Game Arena (BGA) es una **plataforma online** a través de la cual se puede acceder a cientos de juegos creados por la comunidad y a los que se puede jugar de manera gratuita desde el propio navegador.

Un dibujo de un perro

Descripción generada automáticamente con confianza media

Ilustración 10. Board Game Arena

La plataforma cuenta con una extensión llamada Board Game Arena Studio **[12]** donde un usuario puede pasar a formar parte del **equipo de desarrolladores** e implementar su propio videojuego. El propio equipo de desarrolladores avisa que se trata de un software no muy simple de utilizar y se requiere conocimientos medios de 5 lenguajes de programación para poder crear un juego: JavaScript, HTML, CSS, PHP y SQL.

Se requieren esos conocimientos puesto que la creación del juego es desde cero, lo cual incluye interfaz y mecánicas (HTML, JavaScript y CSS), bases de datos (SQL) y la parte del servidor (PHP) **[13].**

**Imagen que contiene Escala de tiempo

Descripción generada automáticamente**

Ilustración 11. BGA tablero

BGA fue uno de los candidatos más estudiados para usarlo como plataforma, pero finalmente se decidió no usarlo debido a su **gran complejidad**. Los propios desarrolladores advierten que no se debe usar esta plataforma sin experiencia en creación de juegos y que es necesario un gran nivel de desarrollo de software, por lo que, para evitar perder mucho tiempo con el aprendizaje de esta plataforma, se decidió usar una más simple.

### 2.3.4. BoardGame.io

BoardGame.io es una **plataforma online** que permite la creación de juegos online a través de la utilización de una base estructural y unas librerías propias. Cualquier usuario puede crear un juego de manera gratuita escribiendo funciones simples y haciendo uso de las **variables de estado** que proporciona la librería **[14].**

Logotipo

Descripción generada automáticamente

Ilustración 12. BoardGame.io

Esta plataforma permite juegos multijugador tanto por turnos como a tiempo real, permitiendo programar tanto la parte del servidor como la del cliente. Se puede programar los juegos en JavaScript plano o en React y la plataforma cuenta con un archivo que contiene **documentación y tutoriales**, así como un **foro comunitario** donde los usuarios pueden dejar sus preguntas.

Una de las ventajas que tiene BoardGame sobre otras plataformas es que cuenta con una interfaz permanente que permite al desarrollador ver el **estado del juego** y sus variables de estado en todo momento durante el desarrollo de la partida. De este modo, se puede comprobar fácilmente si el código se ejecuta correctamente y dónde falla en caso contrario.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 13. Interfaz BoardGame.io

A diferencia de otras plataformas nombradas anteriormente, esta cuenta con la posibilidad de añadir **bots** dirigidos por Inteligencia Artificial a la partida, así como se puede añadir un log de los movimientos con posibilidad de volver hacia atrás en el estado de la partida.

Se ha elegido finalmente utilizar esta plataforma debido a la **estructura simplificada** que presentan sus juegos, estando ordenada en las variables de estado y la posibilidad de realizar eventos y fases dentro del juego, modificando de manera fácil estos estados.

## 2.4. Trabajos y artículos sobre juegos de tablero

En este apartado se comentarán algunos textos que tratan de temas relacionados con la motivación de este trabajo. En concreto se han seleccionado 3 textos de índole diversa para una mejor comparación, más concretamente una propuesta, un trabajo de fin de grado y un artículo científico.

### 2.4.1. Neuroeducación y Juego de Mesa

Este texto es una **propuesta educativa de innovación** para lograr una mejora de las competencias de los alumnos desarrollado por el grupo editorial Devir, con sede en Brasil **[15].**

La **neuroeducación** es un concepto que mezcla por un lado la neurociencia y por otro la pedagogía y la psicología, cuyo objetivo es avanzar en la investigación del funcionamiento del cerebro y como mejorar las capacidades de este.

Básicamente, la neuroeducación plantea una revolución en cuanto a sistema educativo se refiere, buscando dejar de lado el sistema tradicional y poco efectivo con el que se cuenta actualmente, el cual produce una gran tasa de fracasos y sujetos “robotizados”.

Imagen que contiene Forma

Descripción generada automáticamente

Ilustración 14. Neuroeduación

Los juegos, en este caso de tablero, aunque pueden ser de cualquier tipo, engloban los principales elementos en los que se base la neuroeducación. Estos son, por ejemplo, la necesidad de plantear desafíos, el afán de superarse, un feedback instantáneo de las acciones, un sistema de recompensas y lo que es más importante para un niño, la emoción de conseguir la victoria.

En la propuesta mencionan 3 **grupos de ventajas** o habilidades que pueden ser mejoradas con los juegos de mesa aplicados en la educación. Cada una de ellas se puede relacionar con un tipo de inteligencia, según las definidas por Howard Gardner.

Diagrama

Descripción generada automáticamente

Ilustración 15. Tipos de Inteligencia (Howard Gardner)

Por un lado, hablan de una mejora en la **aptitud verbal**, ya no solo en la comunicación, sino también a nivel de comprensión e interpretación de la información que se presenta por vía visual o auditiva en el desarrollo del juego, lo cual estaría dentro de la inteligencia lingüística-verbal, así como la interpersonal.

El segundo punto de mejora es la **aptitud numérica**, ya que en juegos que incorporen cualquier sistema medido por números, será necesario aplicar cierto razonamiento aritmético para mejorar la estrategia, lo que hace que se ejercite la parte más ligada al cálculo del cerebro. Esta formaría parte de la inteligencia lógico-matemática.

Por último, se encuentra la **aptitud espacial y el razonamiento lógico**. La primera mejora el sentido de perspectiva tridimensional, la orientación y el entendimiento de los mapas, formando parte de la inteligencia visual-espacial. El razonamiento lógico, por otro lado, ayuda a mejorar el establecimiento de conexiones y la extracción de conclusiones para resolver problemas. Como su nombre indica, estaría dentro de la inteligencia lógico-matemática.

### 2.4.2. Aplicación de los Juegos de Mesa como Estrategia Pedagógica

Este trabajo de fin de grado, cuyo título completo es **Aplicación de los Juegos de Mesa como Estrategia Pedagógica para facilitar el Aprendizaje en la Educación Infantil**, fue realizado por Luz Nancy Rodríguez en la Facultad de Ciencias Humanas y Sociales de la Corporación Universitaria Iberoamericana, en 2014 **[16].**

Logotipo, nombre de la empresa

Descripción generada automáticamente

Ilustración 16. Corporación Universitaria Iberoamericana

Este trabajo profundiza en el tema tratado por la anterior propuesta educativa ya mencionada, puesto que plantea una manera de cambiar el sistema educativo tradicional por uno basado en la estimulación cerebral por medio de la **autorrealización**. Se presenta como argumento que los juegos de mesa poseen características indispensables para estimular el aprendizaje en cualquier rango de edad (aunque se centra en niños), como puede ser una intención y objetivo didáctico, trabajo en equipo, un reglamento establecido y un sistema de recompensas.

También habla acerca de las mejoras que producen los juegos de mesa en áreas como la adaptación social, el desarrollo cognitivo, el progreso a nivel artístico o el liderazgo, que son consideradas **habilidades necesarias** para un correcto desarrollo personal y laboral.

### 2.4.3. Teoría de Juegos

La **Teoría de Juegos** es un área perteneciente a las matemáticas. En ella, se investiga estrategias óptimas basadas en el comportamiento y las acciones de los jugadores **[17].** Sobre este tema existen numerosos artículos científicos y trabajos relacionados o aplicados, aunque esta vez se comentará el artículo publicado en la página web Investigación y Ciencia, en noviembre de 2019 por la autora Daniela Ovadia **[18].**

En este artículo comentan que la teoría de juegos existe en un contexto dónde se mezcla la cooperación y la competición, puesto que las decisiones que un usuario toma no solo le afectan a él. Esta teoría es interesante porque las posibles acciones que tome un usuario pueden y, de hecho, están influenciadas por las posibles acciones que puedan tomar los demás.

Cuando se han visualizado todos los **posibles escenarios** con las acciones u opciones correspondientes, la teoría de juegos se encarga de formular todo de forma matemática, al igual que se haría en un ambiente económico.

Existen diversos tipos de Teoría de Juegos, según el número de objetivos o de decisores de las acciones:

Tabla

Descripción generada automáticamente

Ilustración 17. Tipos de Teoría de Juegos.

Normalmente la Teoría de Juegos se usa en un **ámbito económico**, pero en este caso se puede extrapolar a los juegos de tablero y la interacción de los jugadores para lograr la victoria.

Este ámbito es realmente interesante y permite que de un simple juego de mesa por turnos se puedan extraer una gran cantidad de información que permite crear un perfil del jugador y su habilidad para tomar decisiones que involucren otras personas.

Así mismo, dicha información se podría usar para un estudio con el objetivo de obtener una **estrategia óptima** para un juego determinado, pudiendo ser posible la creación de una Inteligencia Artificial (p.e. mediante redes neuronales) que logre simular una interacción óptima según las posibilidades que ofrezca el juego.

## 2.5. Inteligencia Artificial en los juegos de tablero

Desde que se fue desarrollando el sector de la inteligencia artificial ya por la década de los 80, sé plantearon diferentes áreas dónde podría acoplarse para mejorar o rivalizar contra **seres humanos**.

Los juegos de mesa fue uno de esos áreas, dónde a través de la creación de bots guiados por IA se ha conseguido vencer a grandes maestros mundiales en su juego, como fue el caso del GM Garry Kasparov en 1996, que fue derrotado en ajedrez por una IA.

Un grupo de personas haciendo gestos con la cara de una persona

Descripción generada automáticamente con confianza baja

Ilustración . Gary Kasparov vs IA (1996)

Se han creado IAs capaces de **jugar y dominar** gran parte de los juegos de mesa más conocidos. La mayor parte de estos bots aprenden a través de **aprendizaje automático**, a base de gran cantidad de datos de partidas reales que usan para ir mejorando, usando diferentes técnicas como aprendizaje supervisado, aprendizaje no supervisado o aprendizaje por refuerzo.

Este tipo de conocimientos también se ha aplicado a juegos de ámbito estratégico, como es el caso del Catán. Existen numerosos artículos que describen la creación de alguna IA para poder jugar de manera óptima o competitiva al Catán, también usadas como método de enseñanza o para entender mejor el juego y sus posibilidades.

Uno de estos artículos es **Intelligent Tutoring Agent for Settlers of Catan [19],** dónde se describe como se ha creado una IA a través del uso de CLIPS para este juego.

A través de este lenguaje se crea un conjunto de reglas para decidir que negociaciones realizar a lo largo del juego. Esta IA se crea con el principal objetivo de ayudar a los niños el funcionamiento de la negociación y como y cuando se debe realizar. De esta forma se ayuda al desarrollo cognitivo y de toma de decisiones de los más jóvenes, todo en un ambiente basado en un juego y con la motivación de ganar el mismo.

Diagrama

Descripción generada automáticamente

Ilustración . Aprendizaje automático

Se trata de un artículo que resume muy bien una de las posibilidades de la creación de este TFG. Este trabajo se presentó como **base** para la posible realización de más artículos y proyectos de esta índole, pues la creación de etas IAs necesitan de numerosos datos y BoardGame.io puede llegar a ser una plataforma ideal para la recolección de estos a través de partidas reales.

# JUEGOS DE TABLERO CON INTERACCIÓN ENTRE JUGADORES

## 3.1. Utilidad y relevancia

Como ya se ha comentado en los artículos mencionados, dejando de lado la parte de ocio y entretenimiento que generan este tipo de juegos, existen numerosas **ventajas y beneficios** a nivel psicológico en los juegos que implican interacción entre los jugadores.

Ya sea a nivel competitivo o a nivel cooperativo, este tipo de juegos fuerzan a los jugadores a tomar decisiones que repercutirán en el estado de su juego, de tal forma que una decisión errada que tome un jugador puede hacer que pierda la partida más adelante. Esto obliga a los jugadores a pensar en una **estrategia**, que no solo engloba al propio jugador sino al resto de los presentes.

Por ejemplo, en un juego como el Catán, que permite comercio, el uso estratégico de esta funcionalidad es clave para poder ganar, puesto que el desarrollo del jugador será mucho mayor si comercia por los recursos que necesita. Sin embargo, el jugador debe tener en cuenta que en un comercio ambos participantes salen beneficiados y su habilidad como comerciante puede desembocar en que ayude demasiado a los oponentes y perder por ello.

Este factor y muchos otros que necesitan de interacción entre los jugadores hace que los usuarios mejoren y desarrollen su capacidad de **toma de decisiones**, con el objetivo de un beneficio propio, es decir, el área que abarca la Teoría de Juegos**,** más concretamente en la Teoría de Juegos Cooperativos.

## 3.2. Los Colonos de Catán

### 3.2.1. Historia

Los Colonos de Catán o simplemente Catán es toda una **saga de juegos** creada por el inventor de juegos de mesa alemán Klaus Teuber **[20]** en 1995. Se trata del primer juego con estilo alemán que se ha popularizado fuera del continente europeo, pues actualmente cuenta con traducciones hasta en 18 idiomas distintos.

Un hombre mayor

Descripción generada automáticamente con confianza media

Ilustración 20. Klaus Teuber

Es tanta la popularidad de esta saga que se sigue actualizando hoy en día, y actualmente cuenta con **5 grandes expansiones** traducidas al español (muchas más y más cuantiosas en otros países), varias ampliaciones para permitir hasta 6 jugadores en el juego base y algunas de las expansiones, diferentes ediciones especiales con eventos únicos, 3 diferentes juegos de mesa con el mismo nombre (cartas, dados y junior) y numerosos subproductos o “spinoffs” de la saga.

Actualmente el juego ha vendido más de **20 millones de copias** (cifra de 2018), generando millones de euros y ha conseguido premios como Juego del año (Spiel des Jahres) en 1995, Origins Award en 1996, Spiel der Spiele en 2001 o Games Magazine Hall of Fame en 2006.

El juego base tiene una duración media de **1 hora**, pero este tiempo se puede alargar dependiendo de la familiaridad de los jugadores con el juego y el tiempo de toma de decisiones, así como el número de jugadores o si se está usando alguna expansión.

### 3.2.2. Reglas básicas

Para poder entender la parte de análisis e implementación del sistema es necesario conocer, al menos teóricamente, **las reglas** de este juego **[21]**. Se trata de un juego de estrategia, por lo que lo que haga cada jugador depende únicamente de sus decisiones para afrontar la partida.

#### 3.2.2.1. Objetivo

El juego base se basa en la construcción de poblados, ciudades y carreteras con el objetivo de conseguir puntos de victoria. Para ganar se necesitan un total de **10 puntos de victoria**, una vez alguien los consiga se acaba el juego automáticamente.

#### 3.2.2.2. Recursos

Para realizar cualquier acción dentro del juego se usan diferentes **recursos** (madera, ladrillo, lana, grano y mineral), que dependiendo del tipo de construcción o acción requerirá diferentes y diferente número.

Imagen que contiene caja, alimentos

Descripción generada automáticamente

Ilustración 21. Cartas de Recurso

Para obtener recursos es necesario tener construcciones colindantes a los terrenos de dicho recurso. También se pueden obtener recursos mediante el comercio con el banco o con otros jugadores, así como con la utilización de cartas de desarrollo, las cuales, después de comprarlas por recursos, ofrecen algún tipo de ventaja para el jugador que las usa, similar a lo que en otros juegos es un comodín.

#### 3.2.2.3. Tablero

El Catán se desarrolla sobre un tablero compuesto de **19 hexágonos de terreno**, cada uno de ellos tiene un número asociado del 2 al 12 y un recurso que produce.

Los terrenos pueden ser granja que ofrece **grano**; montaña, que ofrece **mineral**; pasto, que ofrece **lana**; bosque, que ofrece **madera**; cantera, que ofrece **ladrillo** y el desierto, que no ofrece ningún recurso.

Mapa

Descripción generada automáticamente

Ilustración 22. Tablero Catán

A su vez, algunos de los terrenos colindantes al mar tienen puertos de comercio asociados, que se explicarán más adelante.

#### 3.2.2.4. Etapa de recolección

Al inicio de cada turno el jugador correspondiente tira 2 dados y el número que salga será el que determine los recursos ese turno.

De esta forma, un jugador que tenga un **poblado colindante** a un hexágono con el número que ha salido, robará del banco una carta del recurso de dicho hexágono. Esta regla se aplica a cada construcción de cada jugador, de tal forma que por cada poblado colindante se roba una carta y por cada ciudad colindante se roban dos.

Por estadística y probabilidad, con dos dados de 6 caras, no todos los números del 2 al 12 tienen la **misma probabilidad** de salir, por ello los números 6 y 8 se pintan como números rojos para indicar su mayor ratio de salida, y otros como el 12 o el 2 tienen muchas menos probabilidades de salir.

#### 3.2.2.5. Ladrón

El **número 7**, por otro lado, siendo uno de los más comunes, tiene otra función, puesto que, si sale, cualquier jugador con más de 7 cartas de recursos debe descartarse de la mitad (se redondea para abajo en caso impar), y el jugador que haya tirado puede **mover el ladrón** a otra casilla que el decida.

El ladrón es un elemento importante en el Catán, puesto que condiciona mucho el desarrollo de la partida. Se trata de una pieza que se coloca inicialmente en el terreno de desierto (no ofrece ningún recurso) y que anula el número de la casilla donde se encuentre.

Por ejemplo, basándose en la *Ilustración 3. Tablero de Catán* mostrada anteriormente, el ladrón se encuentra en la casilla de ladrillo con el número 10. De esta forma, el jugador rojo que debería robar 2 cartas de ladrillo (ciudad) y el naranja que debería robar 4 (dos ciudades) no roban ninguna.

#### 3.2.2.6. Construcciones

Existen **3 tipos de construcción** en el juego base, cada una con un coste asociado diferente:

* **Poblados:** Dan un punto de victoria y se construyen en los vértices de cada hexágono, se pueden construir un máximo de 5.
* **Ciudades:** Dan 2 puntos de victoria y se construyen encima de los poblados (destruyendo el poblado), se pueden construir un máximo de 5.
* **Carreteras:** Se construyen en las aristas y sirven para conectar las otras construcciones, se pueden construir un máximo de 15.

Para poder construir poblados por el tablero, es necesario que estos estén **conectados** mediante carreteras con los poblados iniciales, y se debe cumplir la regla de distancia de construcción.

Los **poblados iniciales** son los que se colocan al empezar el juego de manera gratuita para contar con un desarrollo inicial. Se colocan dos por cada jugador, así como una carretera colindante a cada uno, y solo es necesario tener en cuenta la regla de distancia.

La regla de distancia de construcción dice que para poder construir un poblado nuevo es necesario dejar dos aristas de espacio con cualquier otro poblado o ciudad, ya sea del mismo jugador o de otro.

#### 3.2.2.7. Cartas de desarrollo

Otro de los factores importantes comentados antes son las **cartas de desarrollo**. A priori se consiguen como si fuera cualquier otra construcción, pagando los recursos necesarios, sin embargo, lo que obtienes es una carta aleatoria de desarrollo.

Esta carta puede ser de diferentes tipos:

* **Caballero:** Permite mover el ladrón a una casilla que se elija (14 cartas).
* **Invento:** Permite robar dos recursos a elegir del banco de recursos (2 cartas).
* **Monopolio:** Roba todas las cartas de un recurso elegido a los demás jugadores (2 cartas).
* **Carreteras:** Permite construir dos carreteras gratis (2 cartas).
* **Punto de Victoria:** Otorga un punto de victoria (5 cartas).

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración 23. Cartas de Desarrollo

Hay un total de 25 cartas de desarrollo y solo se puede usar una por turno.

#### 3.2.2.8. Comercio

Una de las claves que distingue este juego de otros similares es la **capacidad de negociación** de comercio que ofrece. Existen dos tipos de comercio:

* Con el banco: Se pueden cambiar 4 recursos iguales por 1 a elegir del banco.
* Con otros jugadores: Dos jugadores pueden comerciar con la cantidad y tipo de cartas de recursos que ellos deseen.

Únicamente puede comerciar el jugador que esté jugando en ese turno y si hay un comercio entre jugadores, debe ser él uno de los participantes.

Como se ha comentado antes, algunos hexágonos tienen **puertos de comercio** asociados. Estos puertos ofrecen ventajas comerciales al jugador que tenga una construcción encima del puerto (poblado o ciudad). Esta ventaja puede ser o bien sustituir el 4x1 en comercio con el banco por un 3x1 general o bien permite un 2x1 con un tipo de recurso específico.



Ilustración 24. Puerto 2x1 Ladrillo

#### 3.2.2.9. Cartas de misión

Otra forma de conseguir puntos de victoria es a través de las **dos misiones** que ofrece el juego base, estas son conseguir el mayor ejército o conseguir la mayor ruta comercial.

* **El mayor ejército:** El jugador que tenga esta carta obtiene 2 puntos de victoria extra. Se otorga al primer jugador que use **3 cartas de caballero**. A partir de entonces se puede obtener si otro jugador consigue más cartas de caballero usadas que el dueño de la carta.
* **La mayor ruta comercial:** El jugador que tenga esta carta obtiene 2 puntos de victoria extra. Se otorga al primer jugador que consiga una ruta (sin cortar por otro jugador) de **4 carreteras** seguidas. A partir de entonces se puede obtener si otro jugador logra una ruta más larga que el dueño de la carta. Debe ser una ruta única sin bucles, en caso de ramificaciones se contará la más larga.

### 3.2.3. Estrategias posibles

Puesto que se trata de un juego de toma de decisiones, no hay una estrategia más válida que otra, pero sí que se pueden obtener algunas **guías básicas** para mejorar el juego de alguien principiante.

Una de las guías principales es tener en cuenta los números de las casillas donde se construye e intentar conseguir los números más comunes y recursos variados para poder realizar todo tipo de construcciones.

Será conveniente pensar en si merece la pena centrarse en la obtención de cartas de desarrollo con el objetivo de obtener la carta del mayor ejército, así como intentar conseguir la carta de mayor ruta comercial, o si por otro lado es más conveniente centrarse en la expansión y construcción de poblados y ciudades.

Es muy complicado lograr tener la ventaja en todos estos ámbitos durante la partida por lo que depende de la **estrategia del jugador** elegir como balancea su juego para llegar a la victoria, teniendo siempre en cuenta el desarrollo de los otros jugadores y el comercio que pueda llegar a tener con ellos.

## 3.3. Otros juegos de la misma área

Los juegos de estrategia son actualmente una de las áreas **más populares** y reclamadas entre los juegos de mesa, puesto que permiten ejercitar la capacidad de decisión de los jugadores, así como lograr formar un ambiente familiar o competitivo según se prefiera.

Dentro del ámbito en el que profundiza el Catán, existen otros juegos similares que merece la pena nombrar, puesto que han ayudado en el auge de este sector del entretenimiento **[22]**.

### 3.3.1. Istambul

**Istambul** es un juego de 2 a 5 jugadores con temática **estratégica financiera**. En este juego con partidas aproximadamente de media hora los jugadores deberán liderar su grupo de piezas, compuestas por un comerciante y cuatro asistentes.

Con estas piezas los jugadores deberán recorrer sabiamente las 16 ubicaciones del gran bazar de Estambul, dejando a los asistentes donde consideren para ir realizando comercios tras cargar las carretas desde los almacenes y conseguir así los rubíes necesarios para ganar la partida.

Se trata de un juego fácil de entender y que permite a sus jugadores elaborar estrategias de manera individual para poder derrotar a los oponentes.

Este juego tiene similitud con respecto al Catán en que cada jugador deberá realizar una estrategia en forma de recorrido y acciones para no quedarse sin asistentes y conseguir los rubíes de la manera más rápida posible

Imagen que contiene foto, oso, diferente, montón

Descripción generada automáticamente

Ilustración 25. Juego Istambul

### 3.3.2. Agrícola

**Agrícola** es un juego de mesa que sitúa a los jugadores en la Europa del año 1670. Se trata de un ambiente en dónde el jugador se pone en la piel de un **granjero** y su esposa, los cuales tendrán que ir tomando decisiones en torno a las cosechas y como afecta ésta a la economía.

Es un juego de complejidad variable donde la dificultad radica en la estrategia elegida, que a su vez va en función de las cartas que posea el jugador. Se pueden realizar acciones a través de turnos que pueden variar en recolección, progreso de la casa, incluso tener más hijos para ayudar con el trabajo.

Dependiendo del tipo de partida y la complejidad a la que quieran jugar los participantes, la partida puede durar desde media hora hasta dos horas y media.

Es un juego que tiene gran similitud con el Catán entorno al ambiente en el que se desarrolla, puesto que ambos están orientados a una etapa similar en la historia y en los dos los jugadores deberán tomar decisiones relacionadas con las **actividades comerciales** de ese tiempo.

Mapa

Descripción generada automáticamente

Ilustración 26. Juego Agrícola

### 3.3.3. Alta Tensión

**Alta tensión** es un juego de mesa que pone a los jugadores a experimentar de primera mano la gestión de una **red de suministros de energía**. Los jugadores deberán marcar una ruta por la cual otorgarán energía a las ciudades y competirán por el monopolio de esta.

El juego se basa en la compra de plantas de energía, las cuales se pueden mejorar a través de la adquisición de nuevos modelos, que estarán disponibles según se avance en la partida.

Para poder mantener estas plantas, los jugadores deberán conseguir diferentes tipos de materias primas, lo que convierte el juego en una competencia entre los participantes por conseguir eficiencia y riqueza.

Guarda similitud con el Catán en el manejo y adquisición de los recursos, necesarios para el avance y el desarrollo de cada estrategia.

Mapa

Descripción generada automáticamente con confianza media

Ilustración 27. Juego Alta Tensión

### 3.3.4. Isle of Skye

**Isle of Skye** desarrolla su juego en la isla con el mismo nombre, la cual dentro de la historia es considerada como una de las más bellas del planeta.

En el juego, cada participante se convierte en un **jefe de clan** cuyo objetivo primordial será construir un reino que les permita obtener la mayor cantidad de puntos posible.

Las partidas se basan en las fichas de puntuación que se utilicen, por lo que cada juego es distinto y tiene como consecuencia que los jugadores deban elaborar diferentes estrategias para poder ganar.

El principal factor del juego es el dinero y la gestión de este para la mejora del reino, que determina al ganador del juego. Es importante el hecho de saber comerciar y establecer precios puesto que es un factor decisivo en el desarrollo de la partida.

Este juego es parecido al Catán respecto a la **capacidad de negociación** con otros jugadores.

Dibujo de ingeniería

Descripción generada automáticamente con confianza media

Ilustración 28. Juego Isle of Skye

### 3.3.5. Carcassonne

**Carcassonne** es un juego basado en la **colocación de fichas** para formar un tablero. Los jugadores se sitúan en la época de la antigua Francia y deberán elaborar una estrategia por turnos para decidir dónde colocar cada ficha del terreno y si merece la pena colocar alguna pieza encima.

Al ir completando el terreno se van formando ciudades y carreteras que permiten a los jugadores que tengan piezas encima obtener puntos para lograr la victoria.

Cada jugador deberá elegir si merece la pena mejorar sus territorios o perjudicar a los otros jugadores.

Tiene un factor muy parecido al Catán en el **nivel estratégico**, puesto que las decisiones sobre si defender o atacar marcan el desarrollo de la partida y las estrategias de los demás jugadores.



Ilustración 29. Juego Carcassonne

### 3.3.6. Aventureros al Tren

**Aventureros al Tren** es un juego que se basa en la competitividad por la **empresa ferroviaria** de Norteamérica. Se trata de un juego fácil de aprender basado en la obtención de recursos a través de vagones que se usarán para ir expandiendo las rutas ferroviarias.

Se consiguen puntos en base a la longitud de las rutas, así como al cumplir ciertos objetivos secundarios. Se trata de partidas entre media y una hora que permiten a los jugadores mantener la tensión por conseguir la gloria en este sector.

Guarda bastante relación con el Catán, ya que ambos se basan en la expansión de os jugadores en base a las rutas que construyen y se premia la longitud de estas.

Diagrama

Descripción generada automáticamente

Ilustración 30. Juego Aventureros al Tren

# OBJETIVOS

En este punto se muestran detallados los **objetivos y subobjetivos** que se pretenden alcanzar con la realización de este proyecto.

* **Creación de una adaptación detallada del juego de mesa Los Colonos de Catán en la plataforma BoardGame.io**. Es el objetivo principal de este trabajo y está compuesto a su vez de:
  + Creación de las mecánicas del Catán a través de la estructura de BoardGame.io usando React y JavaScript.
  + Creación de una interfaz interactuable para el Catán a través de JavaScript, React y CSS.
  + (Opcional) Creación de un servidor para poder jugar al Catán en línea desde diferentes dispositivos.
  + (Opcional) Creación de un bot o IA capaz de jugar al Catán en BoardGame.io
* **Creación de un sistema de almacenamiento de la información (log).** Se busca almacenar la información de los jugadores de cada turno en archivo de texto.
* **Realización y validación de pruebas de cada una de las funcionalidades del juego.** Se busca que la adaptación sea lo más parecido a la versión original del juego.
* **Incorporación de los requisitos planteados en el análisis.** El juego deberá cumplir con todos los requisitos de usuario y de sistema definidos en el análisis del sistema.

# ANÁLISIS DEL SISTEMA

En este apartado se especificará detalladamente el **sistema de información** usado para satisfacer las necesidades de los usuarios al utilizar el software propuesto. Para ello se definirán los diferentes requisitos que debe tener el juego programado, así como los casos de uso que se han visto necesarios para corroborar el correcto funcionamiento de este.

## 5.1. Casos de Uso

Un **caso de uso** se define como una acción que realizar el usuario sobre o a través del software. A continuación, se definirán diferentes casos que cubran todas las funcionalidades del sistema siguiendo la siguiente plantilla:

|  |  |
| --- | --- |
| **CU-00** | |
| **Caso de Uso** |  |
| **Actores** |  |
| **Objetivos** |  |
| **Precondiciones** |  |
| **Postcondiciones** |  |

Tabla 1. Plantilla Casos de Uso

Donde:

* **CU-XX:** identifica de manera unívoca el caso de uso, siendo XX el número del caso.
* **Caso de Uso:** resumen de longitud breve que indica en que consiste el caso de uso.
* **Actores**: indica el usuario o usuarios que realizan el caso de uso, en este caso puede ser 1 o 2 jugadores.
* **Objetivos:** funcionalidad que es espera conseguir con el caso de uso.
* **Precondiciones:** condiciones que deben ser cumplidas antes de empezar el caso de uso.
* **Postcondiciones:** condiciones que se tienen que dar cuando el caso de uso haya sido realizado.

**Casos de Uso:**

|  |  |
| --- | --- |
| **CU-01** | |
| **Caso de Uso** | Crear partida nueva |
| **Actores** | Un jugador |
| **Objetivos** | Empezar una partida del Catán |
| **Precondiciones** | * El usuario debe haber ejecutado el juego |
| **Postcondiciones** | * Se crea una partida con el número de jugadores que elija el usuario * Los jugadores elegidos tienen el nombre elegido por el usuario |

Tabla 2. Caso de Uso 01

|  |  |
| --- | --- |
| **CU-02** | |
| **Caso de Uso** | Colocar las estructuras iniciales |
| **Actores** | Un jugador |
| **Objetivos** | El jugador colocará dos poblados y dos carreteras según las normas del Catán. |
| **Precondiciones** | * El usuario debe haber iniciado la partida |
| **Postcondiciones** | * Se crearán las estructuras iniciales para el jugador dónde haya seleccionado en el tablero |

Tabla 3. Caso de Uso 02

|  |  |
| --- | --- |
| **CU-03** | |
| **Caso de Uso** | Tirar los dados |
| **Actores** | Un jugador |
| **Objetivos** | El jugador realizará la acción de tirar los dados pulsando en el icono correspondiente |
| **Precondiciones** | * Debe ser el turno del jugador * Ya ha sido superada la fase de establecimientos iniciales * El turno debe estar en la etapa “Tirar dados” |
| **Postcondiciones** | * Los dados se “tiran” seleccionando un número que se actualizará en el estado del juego * Se acaba la etapa “Tirar Dados” |

Tabla 4. Caso de Uso 03

|  |  |
| --- | --- |
| **CU-04** | |
| **Caso de Uso** | Construir carretera |
| **Actores** | Un jugador |
| **Objetivos** | El jugador construirá una carretera en un lugar disponible del tablero siguiendo las reglas del juego |
| **Precondiciones** | * Debe ser el turno del jugador * El jugador debe tener los recursos necesarios * Debe estar en la etapa libre del turno * El jugador debe de tener carreteras disponibles para construir |
| **Postcondiciones** | * Se construye una carretera del jugador en la casilla del tablero que se haya elegido * Se actualiza el estado de la partida * Se actualiza el estado del jugador |

Tabla 5. Caso de Uso 04

|  |  |
| --- | --- |
| **CU-05** | |
| **Caso de Uso** | Construir poblado |
| **Actores** | Un jugador |
| **Objetivos** | El jugador construirá un poblado en un lugar disponible siguiendo las reglas del juego |
| **Precondiciones** | * Debe ser el turno del jugador * El jugador debe de tener los recursos necesarios * Debe estar en la etapa libre del turno * El jugador debe tener poblados disponibles para construir |
| **Postcondiciones** | * Se construye un poblado del jugador en la casilla del tablero que se haya elegido * Se actualiza el estado de la partida * Se actualiza el estado del jugador |

Tabla 6. Caso de Uso 05

|  |  |
| --- | --- |
| **CU-06** | |
| **Caso de Uso** | Construir ciudad |
| **Actores** | Un jugador |
| **Objetivos** | El jugador construirá una ciudad en un lugar disponible siguiendo las reglas del juego |
| **Precondiciones** | * Debe ser el turno del jugador * El jugador debe de tener los recursos necesarios * Debe estar en la etapa libre del turno * El jugador debe tener ciudades disponibles para construir |
| **Postcondiciones** | * Se construye una ciudad del jugador en la casilla del tablero que se haya elegido * Se actualiza el estado de la partida * Se actualiza el estado del jugador |

Tabla 7. Caso de Uso 06

|  |  |
| --- | --- |
| **CU-07** | |
| **Caso de Uso** | Comprar carta de desarrollo |
| **Actores** | Un jugador |
| **Objetivos** | El jugador compra una carta de desarrollo siguiendo las reglas del juego |
| **Precondiciones** | * Debe ser el turno del jugador * El jugador debe de tener los recursos necesarios * Debe estar en la etapa libre del turno * Debe haber cartas de desarrollo disponibles en el mazo |
| **Postcondiciones** | * Se añade la primera carta de desarrollo del mazo de cartas al jugador * Se actualiza el estado de la partida * Se actualiza el estado del jugador |

Tabla 8. Caso de Uso 07

|  |  |
| --- | --- |
| **CU-08** | |
| **Caso de Uso** | Usar carta de desarrollo Caballero |
| **Actores** | Un jugador |
| **Objetivos** | El jugador usará la carta de desarrollo Caballero para desplazar al ladrón a otra casilla del tablero |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa libre del turno * El jugador debe tener al menos una carta de desarrollo Caballero |
| **Postcondiciones** | * El jugador mueve el ladrón a donde desee del tablero y aplica los efectos correspondientes * Se actualiza el estado de la partida * Se actualiza el estado del jugador * Se elimina la carta de desarrollo utilizada |

Tabla 9. Caso de Uso 08

|  |  |
| --- | --- |
| **CU-09** | |
| **Caso de Uso** | Usar carta de desarrollo Invento |
| **Actores** | Un jugador |
| **Objetivos** | El jugador usará la carta de desarrollo Invento para robar dos cartas de recurso a elección del banco |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa libre del turno * El jugador debe tener al menos una carta de desarrollo Invento |
| **Postcondiciones** | * El jugador selecciona dos cartas de recursos del banco y se le añaden a su inventario * Se actualiza el estado de la partida * Se actualiza el estado del jugador * Se elimina la carta de desarrollo utilizada |

Tabla 10. Caso de Uso 09

|  |  |
| --- | --- |
| **CU-10** | |
| **Caso de Uso** | Usar carta de desarrollo Monopolio |
| **Actores** | Un jugador |
| **Objetivos** | El jugador usará la carta de desarrollo Monopolio para robar todas las cartas de un recurso elegido de los demás jugadores |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa libre del turno * El jugador debe tener al menos una carta de desarrollo Monopolio |
| **Postcondiciones** | * El jugador selecciona un recurso y todas las cartas de ese recurso que posean los demás jugadores sin transferidas al jugador que ha utilizado la carta * Se actualiza el estado de la partida * Se actualiza el estado del jugador * Se elimina la carta de desarrollo utilizada |

Tabla 11. Caso de Uso 10

|  |  |
| --- | --- |
| **CU-11** | |
| **Caso de Uso** | Usar carta de desarrollo Carreteras |
| **Actores** | Un jugador |
| **Objetivos** | El jugador usará la carta de desarrollo Carreteras para construir dos carreteras gratis siguiendo las reglas del juego |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa libre del turno * El jugador debe tener al menos una carta de desarrollo Carreteras |
| **Postcondiciones** | * Se construyen dos carreteras en los lugares elegidos por el jugador * Se actualiza el estado de la partida * Se actualiza el estado del jugador * Se elimina la carta de desarrollo utilizada |

Tabla 12. Caso de Uso 11

|  |  |
| --- | --- |
| **CU-12** | |
| **Caso de Uso** | Comercio con banco |
| **Actores** | Un jugador |
| **Objetivos** | El jugador comerciará 4 cartas de recurso de un mismo tipo por 1 que elija del banco |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa libre del turno * El jugador debe tener al menos 4 cartas de recurso iguales |
| **Postcondiciones** | * Se almacenan en el banco las 4 cartas iguales del jugador * Se incluye la carta seleccionada por el jugador a su inventario * Se actualiza el estado de la partida * Se actualiza el estado del jugador |

Tabla 13. Caso de Uso 12

|  |  |
| --- | --- |
| **CU-13** | |
| **Caso de Uso** | Comercio con jugador |
| **Actores** | Dos jugadores |
| **Objetivos** | Un jugador propondrá un trueque que el otro jugador podrá aceptar o denegar |
| **Precondiciones** | * Debe ser el turno del jugador que ofrece el comercio * Debe estar en la etapa libre del turno * El jugador debe tener al menos 1 carta de recurso para negociar |
| **Postcondiciones** | * Se realiza el intercambio de recursos si el receptor acepta la oferta * Se elimina la petición de comercio si el receptor rechaza la oferta * Se actualiza el estado de la partida * Se actualiza el estado de ambos jugadores |

Tabla 14. Caso de Uso 13

|  |  |
| --- | --- |
| **CU-14** | |
| **Caso de Uso** | Mover ladrón |
| **Actores** | Un jugador |
| **Objetivos** | El jugador seleccionará una casilla para colocar el ladrón siguiendo las reglas del juego |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa “Tirar Dados” * Debe haber salido un 7 en los dados |
| **Postcondiciones** | * La posición del ladrón se actualiza a la casilla elegida por el jugador * Se actualiza el estado de la partida |

Tabla 15. Caso de Uso 14

|  |  |
| --- | --- |
| **CU-15** | |
| **Caso de Uso** | Descartarse |
| **Actores** | Un jugador |
| **Objetivos** | El jugador se descartará de la mitad de sus cartas |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa “Tirar Dados” * El jugador debe tener al menos 8 cartas de recurso |
| **Postcondiciones** | * Se eliminan las cartas de recursos seleccionadas por el jugador y se devuelven al banco * Se actualiza el estado de la partida * Se actualiza el estado del jugador |

Tabla 16. Caso de Uso 15

|  |  |
| --- | --- |
| **CU-16** | |
| **Caso de Uso** | Finalizar turno |
| **Actores** | Un jugador |
| **Objetivos** | El jugador acabará su turno dejando paso al siguiente jugador |
| **Precondiciones** | * Debe ser el turno del jugador * Debe estar en la etapa libre del turno |
| **Postcondiciones** | * Se actualiza el estado de la partida * Se acaba el turno del jugador y pasa a ser el turno del siguiente jugador en la lista |

Tabla 17. Caso de Uso 15

## 5.2. Requisitos de Usuario

En este apartado se definirán los requisitos asociados a las **necesidades del cliente**. En este caso, el término cliente se refiere a un usuario estándar que quiera jugar a la versión adaptada del Catán.

Técnicamente, al ser un proyecto creado sin ninguna solicitud, no existen requisitos de usuario, sin embargo, se ha decidido incorporar requisitos simulando una **oferta profesional** por el desarrollo del software.

Dentro de los requisitos de usuario se separarán en dos tipos: de capacidad y de restricción.

Para la identificación de cada uno de los requisitos, se va a seguir la siguiente plantilla:

|  |  |
| --- | --- |
| **Identificador** | **RQT-XX** |
| **Nombre** |  |
| **Prioridad** |  |
| **Necesidad** |  |
| **Estabilidad** |  |
| **Verificabilidad** |  |
| **Descripción** |  |

Tabla 18. Plantilla Requisito de Usuario

Donde:

* **Identificador:** Código que identifica de manera unívoca al requisito, donde T indica el tipo de requisito (Capacidad/Restricción) y XX representa a un número de dos dígitos.
* **Nombre:** Título del requisito que resume su funcionalidad.
* **Prioridad:** Indica la importancia de este requisito para el funcionamiento general del software. Puede ser Alta, Media o Baja.
* **Necesidad:** Indica la importancia de este requisito desde la perspectiva del cliente. Puede ser Esencial, Deseable u Opcional.
* **Estabilidad:** Indica si el requisito puede ser modificado durante del desarrollo del software. Puede ser Estable o Inestable.
* **Verificabilidad:** Indica si la adaptación del requisito puede ser fácilmente verificable una vez el software esté listo. Puede ser Alta, Media o Baja.
* **Descripción:** Breve resumen del requisito que detalla de manera simple su funcionalidad.

### 5.2.1. Requisitos de Capacidad

Los **requisitos de capacidad** detallan el funcionamiento que debe tener el software para poder cumplir las necesidades del usuario.

|  |  |
| --- | --- |
| **Identificador** | **RQC-01** |
| **Nombre** | Número de jugadores |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de elegir el número de jugadores que tendrá la partida |

Tabla 19. Requisito RQC-01

|  |  |
| --- | --- |
| **Identificador** | **RQC-02** |
| **Nombre** | Nombre de jugadores |
| **Prioridad** | Media |
| **Necesidad** | Deseable |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de elegir el nombre de jugadores que tendrá la partida |

Tabla 20. Requisito RQC-02

|  |  |
| --- | --- |
| **Identificador** | **RQC-03** |
| **Nombre** | Establecimientos iniciales |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Cada uno de los jugadores deberá poder elegir dónde colocar los poblados y carreteras iniciales |

Tabla 21. Requisito RQC-03

|  |  |
| --- | --- |
| **Identificador** | **RQC-04** |
| **Nombre** | Seleccionar carreteras |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de elegir mediante la interfaz la casilla de carretera que apunte con el ratón |

Tabla 22. Requisito RQC-04

|  |  |
| --- | --- |
| **Identificador** | **RQC-05** |
| **Nombre** | Seleccionar intersecciones |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de elegir mediante la interfaz la casilla de poblado/ciudad que apunte con el ratón |

Tabla 23. Requisito RQC-05

|  |  |
| --- | --- |
| **Identificador** | **RQC-06** |
| **Nombre** | Seleccionar hexágono |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de elegir mediante la interfaz la casilla de terreno que apunte con el ratón |

Tabla 24. Requisito RQC-06

|  |  |
| --- | --- |
| **Identificador** | **RQC-07** |
| **Nombre** | Tirar dados |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de tirar los dados al inicio de su turno |

Tabla 25. Requisito RQC-07

|  |  |
| --- | --- |
| **Identificador** | **RQC-08** |
| **Nombre** | Construir carretera |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de construir una carretera siguiendo las reglas del juego y que se actualice el estado de la partid |

Tabla 26. Requisito RQC-08

|  |  |
| --- | --- |
| **Identificador** | **RQC-09** |
| **Nombre** | Construir poblado |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de construir un poblado siguiendo las reglas del juego y que se actualice el estado de la partida |

Tabla 27. Requisito RQC-09

|  |  |
| --- | --- |
| **Identificador** | **RQC-10** |
| **Nombre** | Construir ciudad |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de construir una ciudad siguiendo las reglas del juego y que se actualice el estado de la partida |

Tabla 28. Requisito RQC-10

|  |  |
| --- | --- |
| **Identificador** | **RQC-11** |
| **Nombre** | Comprar carta de desarrollo |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de comprar una carta de desarrollo siguiendo las reglas del juego y que se actualice el estado de la partida |

Tabla 29. Requisito RQC-11

|  |  |
| --- | --- |
| **Identificador** | **RQC-12** |
| **Nombre** | Comercio con banco |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de comerciar con el banco si se cumplen las condiciones para ello |

Tabla 30. Requisito RQC-12

|  |  |
| --- | --- |
| **Identificador** | **RQC-13** |
| **Nombre** | Comercio con jugadores |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de comerciar con otro jugador mediante una oferta que podrá ser rechazada o aceptada |

Tabla 31. Requisito RQC-13

|  |  |
| --- | --- |
| **Identificador** | **RQC-14** |
| **Nombre** | Usar caballero |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de usar la carta de desarrollo caballero si se cumplen las condiciones para ello |

Tabla 32. Requisito RQC-14

|  |  |
| --- | --- |
| **Identificador** | **RQC-15** |
| **Nombre** | Usar invento |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de usar la carta de desarrollo invento si se cumplen las condiciones para ello |

Tabla 33. Requisito RQC-15

|  |  |
| --- | --- |
| **Identificador** | **RQC-16** |
| **Nombre** | Usar monopolio |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de usar la carta de desarrollo monopolio si se cumplen las condiciones para ello |

Tabla 34. Requisito RQC-16

|  |  |
| --- | --- |
| **Identificador** | **RQC-17** |
| **Nombre** | Usar carreteras |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de usar la carta de desarrollo carreteras si se cumplen las condiciones para ello |

Tabla 35. Requisito RQC-17

|  |  |
| --- | --- |
| **Identificador** | **RQC-18** |
| **Nombre** | Finalizar turno |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de terminar el turno cuando desee oportuno |

Tabla 36. Requisito RQC-18

|  |  |
| --- | --- |
| **Identificador** | **RQC-19** |
| **Nombre** | Descartarse |
| **Prioridad** | Media |
| **Necesidad** | Deseable |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de descartarse de la mitad de sus cartas cuando sea necesario |

Tabla 37. Requisito RQC-19

|  |  |
| --- | --- |
| **Identificador** | **RQC-20** |
| **Nombre** | Colocar ladrón |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de colocar el ladrón en una casilla a través de la interfaz cuando sea necesario |

Tabla 38. Requisito RQC-20

|  |  |
| --- | --- |
| **Identificador** | **RQC-21** |
| **Nombre** | Estado de la partida |
| **Prioridad** | Baja |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El usuario deberá ser capaz de ver el estado de la partida en cualquier momento de esta |

Tabla 39. Requisito RQC-21

|  |  |
| --- | --- |
| **Identificador** | **RQC-22** |
| **Nombre** | Mapa aleatorio |
| **Prioridad** | Media |
| **Necesidad** | Deseable |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Al crear una partida nueva se generará un mapa con los terrenos y números distribuidos de manera aleatoria |

Tabla 40. Requisito RQC-22

|  |  |
| --- | --- |
| **Identificador** | **RQC-23** |
| **Nombre** | Victoria |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | La partida terminará cuando uno de los jugadores alcance los 10 puntos de victoria |

Tabla 41. Requisito RQC-23

|  |  |
| --- | --- |
| **Identificador** | **RQC-24** |
| **Nombre** | Inventario jugador |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador podrá ver sus cartas de recurso y desarrollo en su interfaz cuando sea su turno |

Tabla 42. Requisito RQC-24

|  |  |
| --- | --- |
| **Identificador** | **RQC-25** |
| **Nombre** | Desarrollo de partida |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener las etapas y las fases dentro del turno establecidas en las reglas |

Tabla 43. Requisito RQC-25

### 5.2.2. Requisitos de Restricción

Los **requisitos de restricción** indican como debe de ser la implementación de la funcionalidad y que condiciones debe cumplir.

|  |  |
| --- | --- |
| **Identificador** | **RQR-01** |
| **Nombre** | JavaScript |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser desarrollado a través del lenguaje de programación JavaScript |

Tabla 44. Requisito RQR-01

|  |  |
| --- | --- |
| **Identificador** | **RQR-02** |
| **Nombre** | React.js |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser desarrollado usando la biblioteca React.js para JavaScript |

Tabla 45. Requisito RQR-02

|  |  |
| --- | --- |
| **Identificador** | **RQR-03** |
| **Nombre** | CSS |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | La interfaz del juego deberá ser reproducida a través del lenguaje de páginas de estilo CSS |

Tabla 46. Requisito RQR-03

|  |  |
| --- | --- |
| **Identificador** | **RQR-04** |
| **Nombre** | BoardGame.io |
| **Prioridad** | Alta |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser desarrollado de tal forma que sea compatible en la plataforma BoardGame.io |

Tabla 47. Requisito RQR-04

|  |  |
| --- | --- |
| **Identificador** | **RQR-05** |
| **Nombre** | Rango de jugadores |
| **Prioridad** | Media |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá poder jugarse de 2 a 4 jugadores |

Tabla 48. Requisito RQR-05

|  |  |
| --- | --- |
| **Identificador** | **RQR-06** |
| **Nombre** | Control de Versiones |
| **Prioridad** | Baja |
| **Necesidad** | Opcional |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser desarrollado utilizando control de versiones en Github |

Tabla 49. Requisito RQR-06

|  |  |
| --- | --- |
| **Identificador** | **RQR-07** |
| **Nombre** | Fuentes |
| **Prioridad** | Baja |
| **Necesidad** | Esencial |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Todas las fuentes de recursos o información utilizadas para el desarrollo del software deberán ser documentadas |

Tabla 50. Requisito RQR-07

## 5.3. Requisitos de Sistema

Los requisitos del sistema definen las necesidades que presenta el sistema para poder lograr el proyecto deseado.

Dentro de esos requisitos existen los **requisitos funcionales** que definen el funcionamiento del sistema que recogen las necesidades del usuario detalladas en los requisitos de capacidad; y los **requisitos no funcionales**, que definen las características y condiciones del sistema y que recogen lo presentado en los requisitos de restricción.

Para la identificación de cada uno de los requisitos, se va a seguir la siguiente plantilla:

|  |  |
| --- | --- |
| **Identificador** | **RQT-XX** |
| **Nombre** |  |
| **Prioridad** |  |
| **Estabilidad** |  |
| **Verificabilidad** |  |
| **Descripción** |  |
| **Origen** |  |

Tabla 51. Plantilla Requisito de Sistema

Donde:

* **Identificador:** Código que identifica de manera unívoca al requisito, donde T indica el tipo de requisito (Funcional/No Funcional) y XX representa a un número de dos dígitos.
* **Nombre:** Título del requisito que resume su funcionalidad.
* **Prioridad:** Indica la importancia de este requisito para el funcionamiento general del software. Puede ser Alta, Media o Baja.
* **Estabilidad:** Indica si el requisito puede ser modificado durante del desarrollo del software. Puede ser Estable o Inestable.
* **Verificabilidad:** Indica si la adaptación del requisito puede ser fácilmente verificable una vez el software esté listo. Puede ser Alta, Media o Baja.
* **Descripción:** Breve resumen del requisito que detalla de manera simple su funcionalidad.
* **Origen:** Requisito/s de usuario en los que se basa. Cada requisito de usuario debe tener al menos 1 requisito de sistema que lo contemple.

### 5.3.1. Requisitos Funcionales

|  |  |
| --- | --- |
| **Identificador** | **RQF-01** |
| **Nombre** | Mínimo de jugadores |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener un mínimo de 2 jugadores |
| **Origen** | RQC-01,RQR-05 |

Tabla 52. Requisito RQF-01

|  |  |
| --- | --- |
| **Identificador** | **RQF-02** |
| **Nombre** | Máximo de jugadores |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener un máximo de 4 jugadores |
| **Origen** | RQC-01,RQR-05 |

Tabla 53. Requisito RQF-02

|  |  |
| --- | --- |
| **Identificador** | **RQF-03** |
| **Nombre** | Nombre de jugadores |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá aceptar una cadena alfanumérica como nombre para cada uno de los jugadores |
| **Origen** | RQC-02 |

Tabla 54. Requisito RQF-03

|  |  |
| --- | --- |
| **Identificador** | **RQF-04** |
| **Nombre** | Casillas de intersecciones |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener casillas en cada uno de los vértices de los hexágonos de terreno, de tal forma que sean interactuables según se requiera |
| **Origen** | RQC-05 |

Tabla 55. Requisito RQF-04

|  |  |
| --- | --- |
| **Identificador** | **RQF-05** |
| **Nombre** | Casillas de carreteras |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener casillas en cada una de las aristas de los hexágonos de terreno, de tal forma que sean interactuables según se requiera |
| **Origen** | RQC-04 |

Tabla 56. Requisito RQF-05

|  |  |
| --- | --- |
| **Identificador** | **RQF-06** |
| **Nombre** | Casillas de terreno |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener casillas de terreno seleccionables en caso de que se requiera por un evento del juego |
| **Origen** | RQC-06 |

Tabla 57. Requisito RQF-06

|  |  |
| --- | --- |
| **Identificador** | **RQF-07** |
| **Nombre** | Tirar los dados |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá tener un icono clicable que ejecute la acción de tirar los dados |
| **Origen** | RQC-07 |

Tabla 58. Requisito RQF-07

|  |  |
| --- | --- |
| **Identificador** | **RQF-08** |
| **Nombre** | Tirada con valor 7 |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Si la tirada de los dados es 7, el jugador deberá poder colocar el ladrón. A su vez cualquier jugador con más de 7 cartas deberá descartarse la mitad |
| **Origen** | RQC-07, RQC-20, RQC-19 |

Tabla 59. Requisito RQF-08

|  |  |
| --- | --- |
| **Identificador** | **RQF-09** |
| **Nombre** | Colocar ladrón |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego deberá dejar escoger al jugador que le corresponda dónde colocar el ladrón, clicando en una casilla de terreno |
| **Origen** | RQC-20 |

Tabla 60. Requisito RQF-09

|  |  |
| --- | --- |
| **Identificador** | **RQF-10** |
| **Nombre** | Descarte |
| **Prioridad** | Media |
| **Estabilidad** | Inestable |
| **Verificabilidad** | Alta |
| **Descripción** | El jugador deberá descartarse de la mitad de sus cartas si las condiciones lo requieren. Se descartarán de forma aleatoria. |
| **Origen** | RQC-19 |

Tabla 61. Requisito RQF-10

|  |  |
| --- | --- |
| **Identificador** | **RQF-11** |
| **Nombre** | Construir poblado |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El jugador podrá construir un poblado en una casilla de vértice clicando en ella siempre que cumpla las condiciones |
| **Origen** | RQC-09 |

Tabla 62. Requisito RQF-11

|  |  |
| --- | --- |
| **Identificador** | **RQF-12** |
| **Nombre** | Regla de proximidad |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir un poblado a menos de 2 aristas de distancia de otro poblado |
| **Origen** | RQC-09 |

Tabla 63. Requisito RQF-12

|  |  |
| --- | --- |
| **Identificador** | **RQF-13** |
| **Nombre** | Insuficientes recursos poblado |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir un poblado y no tenga los recursos necesarios |
| **Origen** | RQC-09 |

Tabla 64. Requisito RQF-13

|  |  |
| --- | --- |
| **Identificador** | **RQF-14** |
| **Nombre** | Límite de poblados |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador podrá construir un máximo de 5 poblados y se le dará un aviso si intenta construir más |
| **Origen** | RQC-09 |

Tabla 65. Requisito RQF-14

|  |  |
| --- | --- |
| **Identificador** | **RQF-15** |
| **Nombre** | Intersección no disponible |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir un poblado en una casilla que ya tenga otro edificio construido |
| **Origen** | RQC-09 |

Tabla 66. Requisito RQF-15

|  |  |
| --- | --- |
| **Identificador** | **RQF-16** |
| **Nombre** | Construir carretera |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El jugador podrá construir una carretera en una casilla de arista clicando en ella siempre que cumpla las condiciones |
| **Origen** | RQC-08 |

Tabla 67. Requisito RQF-16

|  |  |
| --- | --- |
| **Identificador** | **RQF-17** |
| **Nombre** | Regla de colindancia |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir una carretera sin conexión a ninguna de sus construcciones |
| **Origen** | RQC-08 |

Tabla 68. Requisito RQF-17

|  |  |
| --- | --- |
| **Identificador** | **RQF-18** |
| **Nombre** | Insuficientes recursos carretera |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir una carretera y no tenga los recursos necesarios |
| **Origen** | RQC-08 |

Tabla 69. Requisito RQF-18

|  |  |
| --- | --- |
| **Identificador** | **RQF-19** |
| **Nombre** | Límite de carreteras |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador podrá construir un máximo de 15 carreteras y se le dará un aviso si intenta construir más |
| **Origen** | RQC-08 |

Tabla 70. Requisito RQF-19

|  |  |
| --- | --- |
| **Identificador** | **RQF-20** |
| **Nombre** | Carretera no disponible |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir una carretera en una casilla que ya tenga otra carretera construida |
| **Origen** | RQC-08 |

Tabla 71. Requisito RQF-20

|  |  |
| --- | --- |
| **Identificador** | **RQF-21** |
| **Nombre** | Construir ciudad |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El jugador podrá construir una ciudad en una casilla de vértice clicando en ella siempre que cumpla las condiciones |
| **Origen** | RQC-10 |

Tabla 72. Requisito RQF-21

|  |  |
| --- | --- |
| **Identificador** | **RQF-22** |
| **Nombre** | Poblado incorrecto |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir una ciudad encima de un poblado de otro jugador |
| **Origen** | RQC-10 |

Tabla 73. Requisito RQF-22

|  |  |
| --- | --- |
| **Identificador** | **RQF-23** |
| **Nombre** | Insuficientes recursos ciudad |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir una ciudad y no tenga los recursos necesarios |
| **Origen** | RQC-10 |

Tabla 74. Requisito RQF-23

|  |  |
| --- | --- |
| **Identificador** | **RQF-24** |
| **Nombre** | Límite de ciudades |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador podrá construir un máximo de 4 ciudades y se le dará un aviso si intenta construir más |
| **Origen** | RQC-10 |

Tabla 75. Requisito RQF-24

|  |  |
| --- | --- |
| **Identificador** | **RQF-25** |
| **Nombre** | Intersección no disponible |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente construir una ciudad en una casilla que ya tenga otro edificio construido |
| **Origen** | RQC-10 |

Tabla 76. Requisito RQF-25

|  |  |
| --- | --- |
| **Identificador** | **RQF-26** |
| **Nombre** | Establecimientos iniciales |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego permitirá a cada jugador construir dos poblados y dos carreteras de manera gratuita antes de empezar la partida. Durante esta fase no se podrá realizar ninguna otra acción |
| **Origen** | RQC-03 |

Tabla 77. Requisito RQF-26

|  |  |
| --- | --- |
| **Identificador** | **RQF-27** |
| **Nombre** | Cartas iniciales |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará a cada jugador las cartas de recurso correspondientes al segundo poblado inicial que construyan |
| **Origen** | RQC-03 |

Tabla 78. Requisito RQF-27

|  |  |
| --- | --- |
| **Identificador** | **RQF-28** |
| **Nombre** | Comprar carta de desarrollo |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego tendrá un icono clicable por los jugadores que les dará una carta de desarrollo del mazo de cartas siempre que tengan los recursos necesarios |
| **Origen** | RQC-11 |

Tabla 79. Requisito RQF-28

|  |  |
| --- | --- |
| **Identificador** | **RQF-29** |
| **Nombre** | Insuficientes recursos carta de desarrollo |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente comprar una carta de desarrollo sin los recursos necesarios |
| **Origen** | RQC-11 |

Tabla 80. Requisito RQF-29

|  |  |
| --- | --- |
| **Identificador** | **RQF-30** |
| **Nombre** | Límite de cartas de desarrollo |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso a un jugador si intenta comprar una carta de desarrollo y no quedan en el mazo |
| **Origen** | RQC-11 |

Tabla 81. Requisito RQF-30

|  |  |
| --- | --- |
| **Identificador** | **RQF-31** |
| **Nombre** | Mazo de desarrollo |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego contará con un mazo de 25 cartas de desarrollo: 14 de caballero, 2 de monopolio, 2 de invento, 2 de carreteras y 5 puntos de victoria |
| **Origen** | RQC-11 |

Tabla 82. Requisito RQF-31

|  |  |
| --- | --- |
| **Identificador** | **RQF-32** |
| **Nombre** | Mazo de recursos |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego contará con un mazo de recursos de 19 cartas por cada recurso |
| **Origen** | RQC-22, RQC-21 |

Tabla 83. Requisito RQF-32

|  |  |
| --- | --- |
| **Identificador** | **RQF-33** |
| **Nombre** | Insuficientes cartas de recurso |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador cuando intente robar una carta de recurso, pero no queden en el mazo |
| **Origen** | RQC-22, RQC-21 |

Tabla 84. Requisito RQF-33

|  |  |
| --- | --- |
| **Identificador** | **RQF-34** |
| **Nombre** | Usar caballero |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego permitirá usar la carta de caballero y aplicar su efecto clicando en ella si el jugador cumple las condiciones |
| **Origen** | RQC-14 |

Tabla 85. Requisito RQF-34

|  |  |
| --- | --- |
| **Identificador** | **RQF-35** |
| **Nombre** | Usar invento |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego permitirá usar la carta de invento y aplicar su efecto clicando en ella si el jugador cumple las condiciones |
| **Origen** | RQC-15 |

Tabla 86. Requisito RQF-35

|  |  |
| --- | --- |
| **Identificador** | **RQF-36** |
| **Nombre** | Usar monopolio |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego permitirá usar la carta de monopolio y aplicar su efecto clicando en ella si el jugador cumple las condiciones |
| **Origen** | RQC-16 |

Tabla 87. Requisito RQF-36

|  |  |
| --- | --- |
| **Identificador** | **RQF-37** |
| **Nombre** | Usar carreteras |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego permitirá usar la carta de carreteras y aplicar su efecto clicando en ella si el jugador cumple las condiciones |
| **Origen** | RQC-17 |

Tabla 88. Requisito RQF-37

|  |  |
| --- | --- |
| **Identificador** | **RQF-38** |
| **Nombre** | Una carta por turno |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso al jugador si este intenta usar más de una carta de desarrollo por turno |
| **Origen** | RQC-14, RQC-15, RQC-16, RQC-17 |

Tabla 89. Requisito RQF-38

|  |  |
| --- | --- |
| **Identificador** | **RQF-39** |
| **Nombre** | Carta de mayor ejército |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego otorgará la carta de mayor ejército al primer jugador que use 3 cartas de caballero. Desde ese momento la carta la tendrá el que más caballeros use |
| **Origen** | RQC-21 |

Tabla 90. Requisito RQF-39

|  |  |
| --- | --- |
| **Identificador** | **RQF-40** |
| **Nombre** | Carta mayor ruta comercial |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego otorgará la carta de mayor ruta comercial al primer jugador que logre una ruta de 4 carreteras seguidas. Desde ese momento la carta la tendrá el jugador con la ruta más larga |
| **Origen** | RQC-21 |

Tabla 91. Requisito RQF-40

|  |  |
| --- | --- |
| **Identificador** | **RQF-41** |
| **Nombre** | Condición victoria |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego terminará cuando un jugador llegue a 10 puntos de victoria, siendo este el ganador |
| **Origen** | RQC-23 |

Tabla 92. Requisito RQF-41

|  |  |
| --- | --- |
| **Identificador** | **RQF-42** |
| **Nombre** | Carta de punto de victoria |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Cuando un jugador robe una carta de desarrollo Punto de Victoria, el juego le aumentará en 1 sus puntos de victoria |
| **Origen** | RQC-21 |

Tabla 93. Requisito RQF-42

|  |  |
| --- | --- |
| **Identificador** | **RQF-43** |
| **Nombre** | Fin de turno |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador podrá clicar en el icono correspondiente a Fin de Turno para terminar su turno |
| **Origen** | RQC-18 |

Tabla 94. Requisito RQF-43

|  |  |
| --- | --- |
| **Identificador** | **RQF-44** |
| **Nombre** | Tabla estado |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego tendrá a la vista en todo momento una tabla que muestre el estado de la partida para cada jugador |
| **Origen** | RQC-21 |

Tabla 95. Requisito RQF-44

|  |  |
| --- | --- |
| **Identificador** | **RQF-45** |
| **Nombre** | Cartas de recurso de jugador |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | La interfaz del juego mostrará las cartas de recurso que posee un jugador en su turno |
| **Origen** | RQC-24 |

Tabla 96. Requisito RQF-45

|  |  |
| --- | --- |
| **Identificador** | **RQF-46** |
| **Nombre** | Comercio con banco |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego cambiará 1 recurso elegido por el jugador a cambio de 4 cartas iguales |
| **Origen** | RQC-12 |

Tabla 97. Requisito RQF-46

|  |  |
| --- | --- |
| **Identificador** | **RQF-47** |
| **Nombre** | Insuficientes recursos comercio |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El juego dará un aviso a un jugador si este intenta comerciar con el banco sin tener los recursos necesarios para el cambio |
| **Origen** | RQC-12 |

Tabla 98. Requisito RQF-47

|  |  |
| --- | --- |
| **Identificador** | **RQF-48** |
| **Nombre** | Comercio con jugador emisor |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador podrá realizar una oferta de comercio a otro jugador especificando que ofrece y que solicita |
| **Origen** | RQC-13 |

Tabla 99. Requisito RQF-48

|  |  |
| --- | --- |
| **Identificador** | **RQF-49** |
| **Nombre** | Comercio con jugador receptor |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Un jugador que ha recibido una oferta podrá aceptar o rechazar la oferta propuesta |
| **Origen** | RQC-13 |

Tabla 100. Requisito RQF-49

|  |  |
| --- | --- |
| **Identificador** | **RQF-50** |
| **Nombre** | Fases del turno |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El turno del juego constará de una fase inicial de tirar los dados y una libre. En la fase de tirar los dados y las de eventos como colocar el ladrón el jugador no podrá realizar otras acciones |
| **Origen** | RQC-25 |

Tabla 101. Requisito RQF-50

|  |  |
| --- | --- |
| **Identificador** | **RQF-51** |
| **Nombre** | Etapas del juego |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | Una partida constará de una etapa inicial dónde se elige el número y nombre de los jugadores, otra etapa de construcciones iniciales y la etapa libre |
| **Origen** | RQC-25 |

Tabla 102. Requisito RQF-51

|  |  |
| --- | --- |
| **Identificador** | **RQF-52** |
| **Nombre** | Cartas de desarrollo de jugador |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | La interfaz del juego mostrará las cartas de desarrollo que posee un jugador en su turno |
| **Origen** | RQC-24 |

Tabla 103. Requisito RQF-52

|  |  |
| --- | --- |
| **Identificador** | **RQF-53** |
| **Nombre** | Mapa aleatorio |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El tablero tendrá una distribución aleatoria cuando se cree una nueva partida |
| **Origen** | RQC-22 |

Tabla 104. Requisito RQF-53

### 5.3.2. Requisitos no Funcionales

|  |  |
| --- | --- |
| **Identificador** | **RQN-01** |
| **Nombre** | JavaScript versión |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser programado usando JavaScript, con la versión 1.7 |
| **Origen** | RQR-01 |

Tabla 105. Requisito RQN-01

|  |  |
| --- | --- |
| **Identificador** | **RQN-02** |
| **Nombre** | React.js versión |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser programado usando la librería React.js, con la versión 17.0 |
| **Origen** | RQR-02 |

Tabla 106. Requisito RQN-02

|  |  |
| --- | --- |
| **Identificador** | **RQN-03** |
| **Nombre** | CSS versión |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | La interfaz del software deberá ser programado usando hoja de estilos CSS, con la versión CSS3 |
| **Origen** | RQR-03 |

Tabla 107. Requisito RQN-03

|  |  |
| --- | --- |
| **Identificador** | **RQN-04** |
| **Nombre** | Boardgame.io |
| **Prioridad** | Alta |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser programado usando las librerías que ofrece la plataforma Boargame.io, versión 0.44 |
| **Origen** | RQR-04 |

Tabla 108. Requisito RQN-04

|  |  |
| --- | --- |
| **Identificador** | **RQN-05** |
| **Nombre** | Github versión |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | El software deberá ser desarrollado usando control de versiones con Github versión 2.31 |
| **Origen** | RQR-06 |

Tabla 109. Requisito RQN-05

|  |  |
| --- | --- |
| **Identificador** | **RQN-06** |
| **Nombre** | Bibliografía |
| **Prioridad** | Media |
| **Estabilidad** | Estable |
| **Verificabilidad** | Alta |
| **Descripción** | La memoria técnica del proyecto deberá tener un apartado de bibliografía para indicar de dónde viene la información documentada |
| **Origen** | RQR-07 |

Tabla 110. Requisito RQN-06

## 5.4. Matriz de trazabilidad

Para comprobar de una manera más visual que todos los requisitos de usuario están contemplados dentro del sistema por uno o más requisitos del sistema, a continuación, se mostrará una tabla conocida como **matriz de trazabilidad**.

En ella se muestran en las **columnas** los códigos identificativos de los requisitos de usuario, mientras que en las **filas** se muestran los de los requisitos de sistema. El color rojo en una casilla indica que ambos requisitos están relacionados y por tanto se cumple la norma.

Este apartado podría ser demostrado mirando el apartado **Origen** dentro de los requisitos de sistema, pero para una mayor claridad se muestra también en la tabla.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RQC-01** | **RQC-02** | **RQC-03 RQC-01** | **RQC-04** | **RQC-05 RQC-01** | **RQC-06** | **RQC-07** | **RQC-08** | **RQC-09** | **RQC-10** | **RQC-11** | **RQC-12** | **RQC-13** | **RQC-14** | **RQC-15** | **RQC-16 RQC-0** | **RQC-17** | **RQC-18** | **RQC-19** | **RQC-20** | **RQC-21** | **RQC-22** | **RQC-23** | **RQC-24** | **RQC-25** | **RQR-01** | **RQR-02** | **RQR-03** | **RQR-04** | **RQR-05** | **RQR-06** | **RQR-07** |
| **F01** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F02** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F03** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F04** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F05** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F06** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F07** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F08** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F09** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F11** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F12** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F13** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F14** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F15** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F16** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F17** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F18** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F19** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F21** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F22** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F23** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F24** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F25** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F26** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F27** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F28** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F29** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F30** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F31** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F32** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F33** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F34** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F35** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F36** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F37** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F38** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F39** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F40** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F41** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F42** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F43** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F44** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F45** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F46** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F47** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F48** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F49** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F50** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F51** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F52** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **F53** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **N01** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **N02** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **N03’3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **N04** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **N05** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **N06** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Tabla 111. Matriz de trazabilidad

Como se puede ver, todos los requisitos de usuario están contemplados por al menos un requisito de sistema.

# DISEÑO DEL SISTEMA

## 6.1. Diagrama de clases

En este apartado se explicarán cada una de las **clases** (archivos) en la que se divide el proyecto. A continuación, se muestra un diagrama:

Diagrama

Descripción generada automáticamente

Ilustración 31. Diagrama de clases

El proyecto contiene algunas clases más pero no son relevantes para la utilización del software por lo que no se muestran en el diagrama.

Como se puede ver, los archivos se dividen en extensión .js (JavaScript) para aquellos destinados al **funcionamiento y mecánicas** del software, así como algunos archivos de datos; y la extensión .css (CSS) para aquellos que describen la **interfaz** del archivo .js correspondiente.

Dentro de estos archivos se encuentran:

* **Index.js:** En este archivo se renderiza la aplicación. A su vez también se puede modificar para ajustar el juego multijugador a través de servidor.
* **Index.css:** Define las características generales de la página (tipo de letra, fuente, etc).
* **Game.js:** Es el archivo dónde se define todas las mecánicas del juego. Incluye las etapas, fases y todas las acciones permitidas. Cuenta con numerosas funciones y también define G, que es la variable estado de la partida y la que se encarga de almacenar la información cada turno.
* **placeData.js:** Se trata de un archivo que define a través de una lista todas las casillas vértice del tablero y las conexiones entre ellas. Esto se puede realizar así ya que las casillas y sus IDs son siempre los mismos. Se deja sin indicar el tipo y el dueño pues se modifica durante la partida.
* **roadData.js:** Tiene el mismo formato que placeData.js, su objetivo es definir todas las carreteras y las conexiones entre ellas.
* **App.js:** Este archivo es el que lanza el Cliente del software. Aquí se definen el número de jugadores y se elige si se va a jugar en local o remoto.
* **App.css:** Es un archivo predefinido que aporta BoardGame para dar un logo y unas características básicas a la aplicación.
* **Board.js:** En este archivo se define el tablero y todas las interacciones con la interfaz. También se muestran las acciones reprogramadas para poder ser usadas a través de la interfaz.
* **Board.css:** Aquí se define todas las características visuales del software incluidas en el tablero. El mapa, la tabla o la barra de acciones son descritas en este css.
* **Casilla.js:** Se trata de un componente definido a través de una función. Board.js crea cada uno de los hexágonos llamando a este componente. En este archivo se define todo lo que compone un hexágono de terreno.
* **Casilla.css:** Es la hoja de estilos asociada a cada una de las casillas. Es la parte gráfica de los hexágonos, así como los números y círculos que los componen.

## 6.2. Entorno de desarrollo

Con respecto al entorno informático y herramientas asociadas al **desarrollo del software**, se han tomado las siguientes decisiones:

* En cuanto al **Sistema Operativo**, se ha decidido trabajar en Windows. Puesto que para lanzar el proyecto solo es necesario instalar las librerías y tener acceso a una consola, se ha elegido Windows por su simplicidad y facilidad para instalar programas (Visual Studio Code). A su vez, en Windows es más simple acceder a distintos navegadores para comprobar la compatibilidad del software.
* Como **editor de código** se ha decidido usar Visual Studio Code. Este editor cuenta con facilidades intrínsecas para desarrollo de JavaScript y de React, así como capacidad de conectar con GitHub para llevar a cabo un sistema de control de versiones. Visual Studio tiene un gran número de librerías para facilitar la programación, así como documentación para solucionar dudas.
* Como **lenguaje de programación** se ha decidido usar JavaScript. Este lenguaje permite el desarrollo dinámico de las páginas web, así como numerosas funcionalidades diseñadas para el desarrollo web. A su vez, se ha decidido usar CSS para la definición de la interfaz del software debido a la experiencia y cantidad de documentación disponible vía online.
* Para la **estructura del software** se decidió usar la librería React.js, que permite una organización jerárquica y un desarrollo más avanzado del sitio web.
* Por último, como **plataforma de desarrollo** de videojuegos se decidió usar BoardGame.io. Las razones de esta decisión están descritas en el punto [**2.3.4.**](#_2.3.4._BoardGame.io) del documento.

## 6.3. Diseño de funciones

Todo el juego se basa en el **diseño y uso de funciones**, que son las que implementan cada una de las mecánicas y las funcionalidades para que el software funcione.

Para el desarrollo de las funciones se ha usado un **sistema de documentación** dentro del propio código. Cada función tiene 3 apartados para documentarla:

* **Estado:** Indica en que parte de su desarrollo está la función. Puede ser EN PROCESO, TERMINADA SIN REVISAR Y TERMINADA REVISADA.
* **To-Do:** En caso de que la función este en el estado EN PROCESO, aquí se pondría lo que falta por hacer para terminar la función.
* **Función:** Aquí se define brevemente el objetivo y lo que hace la función.

El código está compuesto por más de 5000 líneas de código y se usan un total de 75 funciones en los archivos Game.js, Board.js y Casilla.js.

## 6.4. Desarrollo software e interfaz

En este apartado se explicarán las **decisiones de diseño** de cada uno de los apartados del juego, así como las funciones principales que se usan para el desarrollo.

Para facilitar la explicación, se ha decidido separar el software en diferentes partes.

### 6.4.1. Tablero

La **construcción del tablero** está dividida en dos partes, al igual que el resto del código: la parte funcional o mecánica y la interfaz o diseño gráfico. El tablero está formado por 19 hexágonos con un terreno y un número asociado. En este caso el terreno está representado por el color (verde-bosque, verde claro-prado, amarillo-campo de trigo, rojo-cantera, gris-montaña).

Imagen que contiene Forma

Descripción generada automáticamente

Ilustración 32. Tablero Catán

En el juego se diferencian 3 tipos de casillas interactuables: las **casillas vértice** o intersecciones, que son dónde se pueden construir pueblos y ciudades; las **casillas carretera** o aristas, que son donde se pueden construir carreteras; y las **casillas terreno**, que son las que otorgan recursos y dónde se puede colocar el ladrón.

Estas casillas están representadas por **3 listas** definidas en Game.js y en los archivos de datos mencionados placeData.js y roadData.js. Para poder vincular cada una de las casillas con su elemento correspondiente en la lista es necesario mantener una **numeración única** a través de IDs, que en este caso corresponden a su posición en la lista.

Diagrama

Descripción generada automáticamente

Ilustración 33. Diagrama de casillas

Una vez creada la infraestructura de las casillas es necesario **crear la interfaz** y vincularla a las listas. El tablero del Catán está formado por hexágonos, que han sido representados a través de código CSS y ordenados y colocados por filas a través de un bucle y propiedades de estilo.

Al crear las casillas mediante el componente Casilla.js se le pasa como parámetro, entre otros, el id para poder vincularlo a la lista de las casillas de terreno. En cuanto a las casillas vértice y arista fue necesario crear dos arrays auxiliares para vincular las listas. Estos arrays contenían **imágenes circulo**, que son las que aparecen al pulsar en construir poblado, ciudad o carretera.

Forma

Descripción generada automáticamente con confianza media

Ilustración 34. Imagen Círculo

Estas imágenes están colocadas en la pantalla siguiendo una **fórmula matemática** puesto que el tablero cumple cierta simetría. A cada imagen más adelante se le asigna ciertas propiedades onClick para poder realizar las acciones correspondientes.

Imagen que contiene Forma

Descripción generada automáticamente

Ilustración 35. Círculos de vértice

Para el caso de colocar el ladrón sucede lo mismo salvo que no aparecen círculos, en este caso las casillas de terreno se “desbloquean” y permiten ser clicadas para poder mover el ladrón.

Las imágenes correspondientes a los poblados, ciudades y carreteras funcionan de manera similar a los círculos. Son **arrays de imágenes** previamente creados y colocados, que, cuando se requiera, se modifica su propiedad de estilo “display” para que aparezcan en la pantalla.

La interfaz está ligada a las variables del juego G y ctx, por lo que al modificar el estado realizando una acción clicando en una casilla, el estado del juego es **automáticamente actualizado** gracias al render simultáneo de React.

### 6.4.2. Barra de acciones

Las **acciones** en este juego están ligadas a la etapa del juego o fase del turno en la que se encuentre el jugador, ya que hay ciertas situaciones dónde no se pueden usar todas las acciones disponibles.

Según la estructura que plantea BoardGame.io, las acciones se llaman “moves” y a no ser que se cree una interfaz propia, a los moves se acceder desde la propia interfaz de BoardGame, que aparece al lanzar el juego a la derecha. Sin embargo, esta interfaz es “para desarrolladores”, el objetivo final es que se acceda a las posibles acciones a través de la interfaz creada para el juego.

Los botones de la barra de acciones se pueden clicar en cualquier momento de la partida, pero solo tendrán efecto si se está en la fase o etapa correspondiente.

Por ello se ha creado la siguiente **barra de acciones**:

Rectángulo

Descripción generada automáticamente con confianza media

Ilustración 36. Barra de acciones

Por orden, las acciones que realizan son las siguientes:

1. **Tirar los dados:**

Esta acción está únicamente disponible al inicio del turno y solo se puede realizar una vez por turno. Al pulsar la variable del estado G correspondiente al número de la tirada, a través de la función ***throwDice*** se actualiza con un número ponderado aleatorio del 2 al 12 (simulando la tirada de 2 dados de 6 caras) y se reparten los recursos a los jugadores que tengan construcciones al lado de terrenos con dicho número.

También si el número es un 7 se activa un algoritmo que comprueba si hay que descartarse a través de ***checkDiscard*** y mueve al jugador actual a la fase ***placeRobber.*** En esta fase solo será posible realizar la acción de mover el ladrón, mediante un clic en alguno de los círculos de los terrenos. En caso de mover el ladrón donde haya algún jugador construido a los alrededores se mostrará el siguiente pop-up:

Diagrama

Descripción generada automáticamente

Ilustración 37. Pop-Up Robar

El jugador podrá clicar en el jugador que desee robar la carta. El pop-up muestra todos los jugadores de la partida, pero solo funcionará con los jugadores que estén disponibles al lado del nuevo ladrón.

1. **Construir poblado:**

Este icono se muestra del color del jugador actual. Una vez se pulse se mostrarán los círculos de construcción encima de las casillas de vértice para que el jugador pueda colocar un poblado donde quiera a través de la función ***buildSettlement***. Se comprobarán las condiciones a través de funciones secundarias, si cumple las condiciones se construirá actualizando el estado de la partida y los recursos; si no se cumplen las condiciones se mostrará un aviso con el inconveniente. Esta acción se puede usar en la fase libre del turno todas las veces que se requiera. Este mismo icono también se usa en la fase de construcción de los primeros establecimientos para desempeñar la función ***buildFirstSettlement***.

1. **Construir ciudad:**

Este icono se muestra del color del jugador actual. Sucede igual a la acción de construir poblado, pero con una ciudad, usando la función ***buildCity***. Solo se puede usar en la fase libre del turno.

1. **Construir carretera:**

Este icono se muestra del color del jugador actual. Una vez se pulse se mostrarán los círculos de construcción encima de las casillas de arista. Funciona igual a las dos acciones anteriores, pero con las condiciones especiales de las carreteras. Utiliza la función ***buildRoad*** o ***buildFirstRoad*** en el caso de los establecimientos iniciales.

1. **Comprar carta de desarrollo:**

Al clicar en este icono, si se cumplen las condiciones de recursos y de disponibilidad, el jugador que haya clicado recibirá una carta de desarrollo del mazo de cartas a través de la función ***buyDevCard***, que se la añadirá a su barra de cartas de desarrollo dónde podrá utilizarla. Esta acción se puede realizar todas las veces que se requiera durante la fase libre del turno.

1. **Comercio:**

Al clicar en este icono, se mostrará un pop-up dónde el jugador podrá seleccionar si comerciar con el banco o con otro de los jugadores:

Cuadrado

Descripción generada automáticamente con confianza media

Ilustración 38. Pop-Up Comercio

Si elige comerciar con el **banco**, se abrirá otro pop-up para que el usuario elija el recurso que quiera ofrecer al banco. Después podrá elegir cual tomar del deck general.

Por otro lado, si elige negociar con **otro jugador**, deberá escribir su nombre y luego escribir la oferta y la demanda. El jugador seleccionado deberá escribir Si o No para aceptar o rechazar la determinada oferta.

En caso de aceptar el comercio, se comprobará que el jugador receptor tenga los recursos necesarios para el intercambio, de ser así se realizará de forma automática.

1. **Terminar turno:**

Esta acción finaliza el turno del jugador que lo clica a través de la función **endTurn.** Se puede usar en cualquier momento de la fase libre del turno, es decir, una vez se hayan tirado los dados.

Se pueden realizar más acciones en el turno a través del uso de las cartas de desarrollo, pero están explicadas en el siguiente apartado.

### 6.4.3. Barra de cartas

Las barras de cartas se usan para **mostrar por pantalla** las cartas de recursos y de desarrollo que tiene un jugador. Se actualizan constantemente por lo que al usar cartas de desarrollo o gastar recursos en alguna acción, las cartas usadas desaparecen de la barra.

Se ha decidido separar las cartas en dos barras, la de **recursos** y la de **desarrollo**, ya que el conteo de cartas para el estado de la partida solo tiene en cuenta las de recursos.

Captura de pantalla de varios colores

Descripción generada automáticamente con confianza media

Ilustración 39. Barra de recursos

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Ilustración 40. Barra de cartas de desarrollo

En ambas barras, al poner el clic en una de las cartas, ésta se hace más grande y se muestra un texto con el nombre de esta. En el caso de las cartas de desarrollo al clicar en ellas y siempre que se cumpla la condición de usar una por turno se activará su efecto:

* **Caballero**: Funciona exactamente igual a si saliera un 7 en la tirada de dados con la excepción de que no se activa la función ***checkDiscard.***
* **Invento:** Al clicar en esta carta de desarrollo se abrirá un pop-up dos veces seguidas donde el jugador podrá elegir cada una de las dos cartas de recursos que el decida del banco:

Captura de pantalla de un video juego

Descripción generada automáticamente

Ilustración 41. Pop-Up Invento

* **Monopolio:** Al hacer clic en este icono se muestra un pop-up similar al del invento. El jugador podrá elegir el recurso que quiere robar de los otros jugadores y al clicar en uno de ellos, todas las cartas de ese recurso que tengan los demás jugadores se traspasarán al jugador que usó la carta.

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 42. Pop-Up Monopolio

* **Carreteras:** Al usar esta carta clicando en ella, se le añadirán al jugador los recursos necesarios para construir dos carreteras y se le moverá a una etapa especial dónde solo podrá salir al construirlas y dónde solo se puede usar esa acción, de esta forma el jugador está obligado a construir las carreteras.

### 6.4.4. Tabla de estado

La **tabla de estado** es una interfaz creada para reflejar algunas de las variables e información que almacena la variable general G. Se trata de una tabla con una fila por jugador y dos filas extra para las cartas de misión.

Interfaz de usuario gráfica, Diagrama, Texto, Aplicación

Descripción generada automáticamente

Ilustración 43. Tabla de estado

Se mostrarán los nombres de cada jugador debajo de su color correspondiente siempre que este jugador esté en la partida. Por ejemplo, en la **Ilustración 40** mostrada arriba solo hay 3 jugadores en la partida, por lo que la fila del jugador blanco no tiene nombre y nunca modificará sus valores.

Las columnas dentro de las 4 primeras filas, por orden significan:

1. **Puntos de Victoria:** El número de puntos que el jugador tiene, si llega a 10 se acaba la partida. También se actualiza cuando recibe una carta de punto de victoria.
2. **Número de poblados**: Indica el número de poblados que tiene construidos ese jugador.
3. **Número de ciudades:** Indica el número de ciudades que tiene construidas ese jugador.
4. **Número de carreteras:** Indica el número de carreteras que tiene construidas ese jugador.
5. **Número de cartas:** Indica el número de cartas de recursos totales que tiene le jugador. Si posee más de 7 y sale el número 7 en la tirada, deberá descartarse de la mitad.

Las otras dos filas son la carta de **mayor ruta comercial** y la carta de **mayor ejército**. El color de cada número indica al jugador al que pertenece.

Por un lado, la fila de la ruta comercial indica el número de carreteras conectadas que tiene ese jugador, y la fila del mayor ejército indica el número de caballeros usados por ese jugador. La tabla no indica quién tiene la carta directamente, pero se puede ver por el que tenga el mayor valor y fijándose en los puntos de victoria.

## 6.5. Manual de usuario

Para comprender mejor como jugar a esta adaptación y el uso de cada botón de la interfaz, se recomienda leer el apartado [**6.4. Desarrollo software e interfaz**](#_6.4._Desarrollo_software)**.**

En este apartado se resumirá que debe hacer el jugador en caso de que no se lo indique directamente el juego. Para ello se redactará que hacer **según la etapa o fase** del juego en la que se encuentre.

### 6.5.1. Descarga e instalación del software

El código del juego está disponible en el repositorio público de Github, el enlace se encuentra en [**12. FUENTES**](#_12._FUENTES)**.**

Para poder ejecutar el código es necesario **tener instalado** Node.js y npm. Para ello será necesario instalarlo desde el siguiente enlace:

**Nodejs (Sitio oficial):** <https://nodejs.org/en/download/>

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Ilustración 44. Sitio web Node.js

Una vez instalado se puede comprobar la correcta instalación desde un sistema de comando ejecutando:

* **node -v**
* **npm -v**

Si al usar el comando de iniciación del juego sigue dando error, es posible que se solucione usando el siguiente comando:

* **npm install**

Una vez descargado el proyecto e instaladas las librerías, será necesario **colocarse en la carpeta** catan-react, que es la que contiene el código:

Texto

Descripción generada automáticamente

Ilustración 45. Captura CMD proyecto

Una vez situado en la carpeta ejecute el siguiente comando:

* **npm start**

El juego debería abrirse en **localhost:3000** de manera correcta después de estos pasos.

### 6.5.2. Fase inicial del juego

Al crear partida, el juego solicitará al usuario que indique el **número de jugadores** y posteriormente el **nombre** de cada uno.

Una vez empiece la partida, cada jugador solo podrá realizar **2 acciones**: Construir Poblado y Construir Carretera. Esto es lo que se denomina Fase de Preparación.

Primero cada jugador construye un poblado y luego una carretera (en ese orden) y el juego saltará automáticamente al siguiente jugador. Si la construcción no es válida, el juego dará un aviso del porqué y se podrá repetir la acción correspondiente. Cuando todos los jugadores tengan su poblado y su carretera comenzará la segunda ronda de construcción dónde se repetirá el orden, primero poblado y luego carretera.

Una vez todos los jugadores tengan dos poblados y dos carreteras el juego dará un aviso y **empezará la partida**.

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Ilustración 46. Aviso preparación

### 6.5.3. Turno del jugador

Una vez la partida haya empezado los turnos serán normales. Según le llega el turno al jugador lo único que puede hacer es **Tirar los Dados.**

Cuando haya tirado los dados empezará la **fase libre del turno** (a no ser que saque un 7), dónde podrá realizar cualquier acción de la barra de acciones (salvo tirar los dados de nuevo) o usar una carta de desarrollo.

Si el jugador considera que no quiere hacer más en su turno, deberá pulsar el botón de **Fin de Turno** y el turno del siguiente jugador comenzará.

### 6.5.4. Fases especiales

Dentro de un turno puede ser que el jugador sea “movido” a fases de eventos o de acciones, dónde el jugador **solo** podrá realizar ciertas acciones. Estas fases pueden ser debidas a que haya salido un 7 en los dados, en cuyo caso el jugador deberá colocar el ladrón; usar las cartas de desarrollo caballero o invento; o en el caso de que haya comercio. Todas estas excepciones están descritas en el apartado [**6.4.2 Barra de acciones**](#_6.4.2._Barra_de) y [**6.4.3 Barra de cartas**](#_6.4.3._Barra_de)**.**

# BATERÍA DE PRUEBAS

En este apartado se describirán las pruebas realizadas para el **correcto funcionamiento** del sistema y sus requisitos. Todas las pruebas han sido realizadas usando el software desarrollado para el trabajo.

## 7.1. Pruebas

El objetivo de las pruebas es cubrir todos los requisitos del sistema de manera satisfactoria. Las pruebas seguirán la siguiente plantilla:

|  |  |
| --- | --- |
| **PR-XX** | |
| **Descripción** |  |
| **Requisitos previos** |  |
| **Acciones** |  |
| **Resultado esperado** |  |
| **Resultado** |  |
| **Origen** |  |

Tabla 112. Plantilla Pruebas

Donde:

* **Descripción:** Breve resumen sobre el objetivo de la prueba.
* **Requisitos previos:** Condiciones que se tienen que cumplir para empezar la prueba.
* **Acciones:** Pasos a realizar en la prueba.
* **Resultado esperado:** Resultado hipotético como consecuencia de una prueba exitosa.
* **Resultado:** Conclusión de la prueba, puede ser “Favorable” o “No favorable”.
* **Origen**: Requisito(s) del sistema que ha(n) dado lugar a la prueba.

**Pruebas:**

|  |  |
| --- | --- |
| **PR-01** | |
| **Descripción** | Se busca inicializar el juego, eligiendo un número de jugadores y asignando sus nombres |
| **Requisitos previos** | * Inicializar el programa |
| **Acciones** | 1. Elegir un número entre 2 y 4 para los jugadores. 2. Asignar un nombre alfanumérico a cada jugador |
| **Resultado esperado** | Se inicia la fase de construcciones iniciales con el número de jugadores elegido y sus nombres actualizados |
| **Resultado** | Favorable |
| **Origen** | RQF-01, RQF-02, RQF-03, RQF-51, RQF-53 |

Tabla 113. Prueba 01

|  |  |
| --- | --- |
| **PR-02** | |
| **Descripción** | Se busca completar la fase de establecimientos iniciales construyendo los poblados y carreteras correspondientes a los jugadores elegidos |
| **Requisitos previos** | * Haber completado la fase de elección de jugadores |
| **Acciones** | 1. Construir un poblado para el jugador del turno 2. Construir una carretera para el jugador del turno 3. Repetir 1 y 2 hasta que se termine la fase |
| **Resultado esperado** | Sale el aviso de fin de fase y se han construido todos los establecimientos en los lugares que se ha elegido, actualizando el estado de la partida |
| **Resultado** | Favorable |
| **Origen** | RQF-03, RQF-04, RQF-11, RQF-16, RQF-26, RQF-27 |

Tabla 114. Prueba 02

|  |  |
| --- | --- |
| **PR-03** | |
| **Descripción** | Se busca comprobar que funciona la acción de tirar los dados si la tirada no es 7. |
| **Requisitos previos** | * Haber terminado la fase de establecimientos iniciales * Estar en el inicio del turno * Que el número de los dados no sea 7 |
| **Acciones** | 1. Pulsar el botón de tirar los dados 2. Comprobar que se hayan robado las cartas correspondientes en caso de ser necesario |
| **Resultado esperado** | El estado del juego se actualiza y los jugadores reciben las cartas de recursos correspondientes al número que haya salido en los dados |
| **Resultado** | Favorable |
| **Origen** | RQF-07, RQF-45 |

Tabla 115. Prueba 03

|  |  |
| --- | --- |
| **PR-04** | |
| **Descripción** | Se busca provocar que en los dados salga un 7 para comprobar la funcionalidad de descarte y de colocar el ladrón |
| **Requisitos previos** | * Que 1 o más jugadores tengan más de 7 cartas de recursos * Tirar los dados y que el resultado sea 7 |
| **Acciones** | 1. Elegir una casilla de terreno distinta a donde está el actual ladrón 2. Elegir un jugador para robarle la carta de entre los que tengan construido al lado del nuevo ladrón |
| **Resultado esperado** | El/Los jugadores con más de 7 cartas se descartan automáticamente de la mitad de sus cartas de manera aleatoria. El ladrón se actualiza de posición y una de las cartas del jugador elegido es traspasada al jugador del turno correspondiente |
| **Resultado** | Favorable |
| **Origen** | RQF-06, RQF-08, RQF-09, RQF-10 |

Tabla 116. Prueba 04

|  |  |
| --- | --- |
| **PR-05** | |
| **Descripción** | Se busca provocar el aviso de proximidad en la construcción de un poblado |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en construir poblado 2. Elegir una casilla colindante a un poblado ya construido |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-12 |

Tabla 117. Prueba 05

|  |  |
| --- | --- |
| **PR-06** | |
| **Descripción** | Se busca provocar el aviso de falta de recursos en la construcción de un poblado |
| **Requisitos previos** | * Estar en la fase libre del turno * No tener recursos suficientes |
| **Acciones** | 1. Clicar en construir poblado 2. Elegir una casilla viable para construir |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-13 |

Tabla 118. Prueba 06

|  |  |
| --- | --- |
| **PR-07** | |
| **Descripción** | Se busca provocar el aviso de límite de poblados en la construcción de un poblado |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener 5 poblados construidos |
| **Acciones** | 1. Clicar en construir poblado 2. Elegir una casilla válida |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-14 |

Tabla 119. Prueba 07

|  |  |
| --- | --- |
| **PR-08** | |
| **Descripción** | Se busca provocar el aviso de terreno no disponible en la construcción de un poblado |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en construir poblado 2. Elegir una casilla no conectada a ninguna carretera del jugador |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-15 |

Tabla 120. Prueba 08

|  |  |
| --- | --- |
| **PR-09** | |
| **Descripción** | Se busca construir una carretera de manera normal y conseguir la carta de mayor ruta comercial |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener 3 carreteras consecutivas * Nadie posee la carta de mayor ruta comercial |
| **Acciones** | 1. Clicar en construir poblado 2. Elegir una casilla colindante a un poblado ya construido |
| **Resultado esperado** | Se construye una carretera del jugador que ha clicado en el lugar indicado y se actualiza la tabla de estado de partida. La carta de mayor ruta comercial es entregada al jugador. |
| **Resultado** | Favorable |
| **Origen** | RQF-05, RQF-16, RQF-40 |

Tabla 121. Prueba 09

|  |  |
| --- | --- |
| **PR-10** | |
| **Descripción** | Se busca provocar el aviso de carreteras no colindantes en la construcción de una carretera |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en construir carretera 2. Elegir una casilla que no conecte a ninguna de las construcciones del jugador actual |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-17 |

Tabla 122. Prueba 10

|  |  |
| --- | --- |
| **PR-11** | |
| **Descripción** | Se busca provocar el aviso de insuficientes recursos en la construcción de una carretera |
| **Requisitos previos** | * Estar en la fase libre del turno * No tener suficientes recursos para construir una carretera |
| **Acciones** | 1. Clicar en construir carretera 2. Elegir una casilla válida para construir carretera |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-18 |

Tabla 123. Prueba 11

|  |  |
| --- | --- |
| **PR-12** | |
| **Descripción** | Se busca provocar el aviso de límite de carreteras en la construcción de una carretera |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener 15 carreteras construidas |
| **Acciones** | 1. Clicar en construir carretera 2. Elegir una casilla válida para construir carretera |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-19 |

Tabla 124. Prueba 12

|  |  |
| --- | --- |
| **PR-13** | |
| **Descripción** | Se busca provocar el aviso de proximidad en la construcción de un poblado |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en construir carretera 2. Elegir una casilla con una carretera ya construida |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-20 |

Tabla 125. Prueba 13

|  |  |
| --- | --- |
| **PR-14** | |
| **Descripción** | Se busca construir una ciudad de manera normal y ganar la partida |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener 9 puntos de victoria |
| **Acciones** | 1. Clicar en construir ciudad 2. Elegir una casilla donde haya un poblado del jugador actual |
| **Resultado esperado** | Se construye una ciudad encima del poblado y se actualiza la tabla de estado. Se termina el juego |
| **Resultado** | Favorable |
| **Origen** | RQF-21, RQF-41 |

Tabla 126. Prueba 14

|  |  |
| --- | --- |
| **PR-15** | |
| **Descripción** | Se busca provocar el aviso de poblado incorrecto en la construcción de una ciudad |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en construir ciudad 2. Elegir una casilla con un poblado de otro jugador |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-22 |

Tabla 127. Prueba 15

|  |  |
| --- | --- |
| **PR-16** | |
| **Descripción** | Se busca provocar el aviso de insuficientes recursos en la construcción de una ciudad |
| **Requisitos previos** | * Estar en la fase libre del turno * No tener suficientes recursos para construir una ciudad |
| **Acciones** | 1. Clicar en construir ciudad 2. Elegir una casilla válida para construir ciudad |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-23 |

Tabla 128. Prueba 16

|  |  |
| --- | --- |
| **PR-17** | |
| **Descripción** | Se busca provocar el aviso de límite de ciudades en la construcción de una ciudad |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener 4 ciudades construidas |
| **Acciones** | 1. Clicar en construir ciudad 2. Elegir una casilla válida para construir ciudad |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-24 |

Tabla 129. Prueba 17

|  |  |
| --- | --- |
| **PR-18** | |
| **Descripción** | Se busca provocar el aviso de terreno no disponible en la construcción de una ciudad |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en construir ciudad 2. Elegir una casilla que no tenga poblado construido |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-25 |

Tabla 130. Prueba 18

|  |  |
| --- | --- |
| **PR-19** | |
| **Descripción** | Se busca comprar una carta de desarrollo y que sea un punto de victoria |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en comprar carta de desarrollo 2. Debe de ser un punto de victoria, de lo contrario volver al punto 1 |
| **Resultado esperado** | El jugador recibe una carta de desarrollo de punto de victoria que le aparecerá en la zona de cartas de desarrollo. A su vez, la tabla de estado se actualizará sumando 1 a los puntos de victoria |
| **Resultado** | Favorable |
| **Origen** | RQF-28, RQF-52, RQF-42, RQF-44 |

Tabla 131. Prueba 19

|  |  |
| --- | --- |
| **PR-20** | |
| **Descripción** | Se busca que salte el aviso por falta de recursos al comprar una carta de desarrollo |
| **Requisitos previos** | * Estar en la fase libre del turno * No tener recursos suficientes para comprar una carta de desarrollo |
| **Acciones** | 1. Clicar en comprar carta de desarrollo |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-29 |

Tabla 132. Prueba 20

|  |  |
| --- | --- |
| **PR-21** | |
| **Descripción** | Se busca comprar una carta de desarrollo y que no queden en el mazo de cartas de desarrollo |
| **Requisitos previos** | * Estar en la fase libre del turno * Que no queden cartas de desarrollo para comprar |
| **Acciones** | 1. Clicar en comprar carta de desarrollo |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-30, RQF-31 |

Tabla 133. Prueba 21

|  |  |
| --- | --- |
| **PR-22** | |
| **Descripción** | Se busca que salte el aviso por falta de cartas de recursos en el mazo de recursos |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener una carta Invento * Que al menos un recurso del banco no tenga cartas para robar |
| **Acciones** | 1. Clicar en Invento 2. Elegir un recurso que no haya en el banco |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida. La carta de desarrollo desaparece de la mano del jugador. |
| **Resultado** | Favorable |
| **Origen** | RQF-32, RQF-33, RQF-35 |

Tabla 134. Prueba 22

|  |  |
| --- | --- |
| **PR-23** | |
| **Descripción** | Se busca usar la carta caballero y el mayor ejército |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener una carta Caballero * Tener 2 cartas de caballero usadas * Que nadie posea la carta de mayor ejército |
| **Acciones** | 1. Clicar en Caballero 2. Elegir una casilla para mover el ladrón |
| **Resultado esperado** | El ladrón se mueve a la casilla seleccionada. La carta de desarrollo desaparece de la mano del jugador. La carta de mayor ejército es entregada al jugador. |
| **Resultado** | Favorable |
| **Origen** | RQF-34, RQF-39 |

Tabla 135. Prueba 23

|  |  |
| --- | --- |
| **PR-24** | |
| **Descripción** | Se busca usar la carta de carreteras |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener una carta Carreteras |
| **Acciones** | 1. Clicar en Carreteras 2. Comprar dos carreteras de manera normal y válida |
| **Resultado esperado** | Las dos carreteras se construyen de manera normal y se actualiza la tabla de estado. La carta de desarrollo desaparece de la mano del jugador. |
| **Resultado** | Favorable |
| **Origen** | RQF-37 |

Tabla 136. Prueba 24

|  |  |
| --- | --- |
| **PR-25** | |
| **Descripción** | Se busca usar la carta monopolio |
| **Requisitos previos** | * Estar en la fase libre del turno * Tener una carta Monopolio |
| **Acciones** | 1. Clicar en Monopolio 2. Elegir un recurso |
| **Resultado esperado** | Todas las cartas del recurso elegido que tengan los rivales en la mano se transfieren al jugador, actualizando la tabla de estado. La carta de desarrollo desaparece de la mano del jugador. |
| **Resultado** | Favorable |
| **Origen** | RQF-36 |

Tabla 137. Prueba 25

|  |  |
| --- | --- |
| **PR-26** | |
| **Descripción** | Se busca que salte el aviso por usar más de una carta de desarrollo por turno |
| **Requisitos previos** | * Estar en la fase libre del turno * Haber usado una carta de desarrollo en el turno |
| **Acciones** | 1. Clicar en alguna carta de desarrollo |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-38 |

Tabla 138. Prueba 26

|  |  |
| --- | --- |
| **PR-27** | |
| **Descripción** | Se busca comerciar con el banco |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en Comerciar 2. Clicar en Banco 3. Elegir una oferta |
| **Resultado esperado** | El banco recibe las 4 cartas del recurso que ofrezca el jugador y el jugador recibe la carta que ha elegido. |
| **Resultado** | Favorable |
| **Origen** | RQF-46 |

Tabla 139. Prueba 27

|  |  |
| --- | --- |
| **PR-28** | |
| **Descripción** | Se busca que salte el aviso de falta de recursos al comerciar con el banco |
| **Requisitos previos** | * Estar en la fase libre del turno * No tener recursos suficientes para comerciar con el banco |
| **Acciones** | 1. Clicar en Comerciar 2. Clicar en Banco 3. Elegir una oferta |
| **Resultado esperado** | El sistema da un aviso al jugador de que la acción es inválida |
| **Resultado** | Favorable |
| **Origen** | RQF-47 |

Tabla 140. Prueba 28

|  |  |
| --- | --- |
| **PR-29** | |
| **Descripción** | Se busca probar el botón de fin de turno |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en Fin de turno |
| **Resultado esperado** | El sistema otorga el turno al siguiente jugador |
| **Resultado** | Favorable |
| **Origen** | RQF-43, RQF-50 |

Tabla 141. Prueba 29

|  |  |
| --- | --- |
| **PR-30** | |
| **Descripción** | Se busca comerciar con otro jugador |
| **Requisitos previos** | * Estar en la fase libre del turno |
| **Acciones** | 1. Clicar en Comerciar 2. Clicar en Jugador 3. Elegir una oferta 4. Aceptar/Rechazar la oferta |
| **Resultado esperado** | Se produce el intercambio ofertado entre los jugadores elegidos para el comercio. Se actualiza la tabla de estado. |
| **Resultado** | Favorable |
| **Origen** | RQF-48, RQF-49 |

Tabla 142. Prueba 30

## 7.2. Matriz de trazabilidad

En la siguiente tabla se mostrará las relaciones entre las pruebas diseñadas y los requisitos funcionales del sistema. Todos los requisitos del sistema **deben estar cubiertos** al menos por una prueba.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PR-01** | **PR-02** | **PR-03 RQC-01** | **PR-04** | **PR-05 RQC-01** | **PR-06** | **PR-07** | **PR-08** | **PR-09** | **PR-10** | **PR-11** | **PR-12** | **PR-13** | **PR-14** | **PR-15** | **PR-16 RQC-0** | **PR-17** | **PR-18** | **PR-19** | **PR-20** | **PR-21** | **PR-22** | **PR-23** | **PR-24** | **PR-25** | **PR-26** | **PR-27** | **PR-28** | **PR-29** | **PR-30** |
| **01** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **02** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **03** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **04** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **05** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **06** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **07** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **08** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **09** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **11** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **12** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **13** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **14** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **15** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **16** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **17** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **18** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **19** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **21** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **22** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **23** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **24** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **25** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **26** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **27** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **28** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **29** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **30** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **31** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **32** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **33** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **34** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **35** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **36** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **37** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **38** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3938** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **40** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **41** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **42** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **43** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **44** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **45** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **46** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **47** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **48** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **49** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **50** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **51** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **52** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **53** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Tabla 143. Matriz de trazabilidad de pruebas

Como se puede ver, **todos** los requisitos del sistema funcionales están cubiertos por al menos una prueba, por lo que se completa la verificabilidad del sistema.

# PLAN DE PROYECTO

En este apartado se recoge la organización que se ha llevado a cabo para las tareas necesarias para el desarrollo del proyecto. A su vez, también se incluye el presupuesto necesario para poder llevar a cabo el trabajo.

## 8.1. Gestión del proyecto

El proyecto fue **acordado y cerrado** el 26 de octubre de 2020 junto a los **tutores** Javier Ignacio Carbo Rubiera y Alejandro Cervantes Rovira, aunque no fue hasta el 11 de enero de 2021 que se empezó con el desarrollo de este.

El proyecto fue dividido en dos grandes áreas que desarrollar: el código y la documentación.

Se decidió **no** realizar un desarrollo simultáneo de las dos áreas debido a que gran parte de la memoria depende del trabajo realizado y durante los primeros meses era algo incierto el camino a recorrer para su realización.

A continuación, se muestra un desglose de las tareas necesarias para el **desarrollo del software**, así como el tiempo invertido y las fechas de su realización:



Ilustración 47. Plan de proyecto de software

La parte de la memoria se empezó a desarrollar a mediados de mayo de 2021, una vez el proyecto ya tenía gran parte desarrollada y se habían esclarecido los objetivos.

A continuación, se muestra un desglose de la **parte de la memoria**, no se muestra fecha de inicio o de fin puesto que los apartados se fueron rellenando de manera simultánea y muchos fueron actualizados o modificados días después de su redacción:



Ilustración 48. Plan de proyecto de documentación

## 8.2. Presupuesto del proyecto

En este punto se muestra una aproximación de lo que podría costar un proyecto como este en un **ambiente profesional**. Estos costes quedan divididos en 3 apartados: los costes de personal, costes en software y costes en hardware. El IVA no ha sido incluido en estos costes.

### 8.2.1. Costes de personal

Gracias a las tablas generadas en la gestión de proyecto, se pueden extraer las horas dedicadas y separarlas según la función que desempeñen:

* Preparación y estudio: 90 horas.
* Diseño de Software: 82 horas.
* Diseño web (Interfaz): 28 horas.
* Documentación: 58 horas.

Una vez establecidos los roles y aproximando los **salarios medios** en España para este tipo de trabajos, se puede estimar los costes generados:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Apellidos, Nombre** | **Cargo** | **Coste / Hora [€/hora]** | **Horas trabajadas** | **Coste [€]** |
| Sanz Gómez, Adrián | Desarrollador de Software | 22 | 148 | 3.256,00 |
| Sanz Gómez, Adrián | Diseñador web | 18 | 28 | 504,00 |
| Sanz Gómez, Adrián | Programador | 17 | 82 | 1.394,00 |
| **TOTAL:** | | | 258 | 5.154,00 |

Tabla 144. Costes de Personal

En estos costes de personal se incluyen gastos derivados de la seguridad social, dietas, transporte, etc…

### 8.2.2. Costes de software

Prácticamente todo el software utilizado para el proyecto salvo el sistema operativo y las herramientas usadas para la documentación son **software libre** y gratuito.

|  |  |
| --- | --- |
| **Software** | **Precio [€]** |
| Windows 10 Home | 145,00 |
| Microsoft Office 365 | 69,00 |
| **TOTAL:** | **214,00** |

Tabla 145. Costes de Software

### 8.2.3. Costes de hardware

A lo largo del proyecto se han usado algunos **elementos hardware** para programar y realizar la documentación. No se ha utilizado ningún hardware de carácter específico, pero sí algunos dispositivos.

Puesto que no sería lógico solicitar el importe completo de dichos dispositivos, se ha aplicado la siguiente fórmula:

Con una duración del proyecto de 7 meses:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recurso** | **Precio [€]** | **Vida útil** | **% Uso** | **Coste imputable [€]** |
| REDSMOKE - RTX™ 3060 TI | 1.479,00 | 70 meses | 100 | 147,90 |
| Ratón G502 Hero Logitech | 51,99 | 120 meses | 100 | 3,03 |
| Teclado Mars Gaming MRK0 | 13,90 | 240 meses | 100 | 0,40 |
| TOTAL: | | | | 151,33 |

Tabla 146. Costes de Hardware

### 8.2.4. Costes totales

Una vez nombrados todos los costes anteriores, es necesario sumar los **gastos indirectos** procedentes del suministro eléctrico, el material de oficina y el servicio de internet (el coste de la electricidad y el internet se ha calculado en base a los dispositivos hardware y contrato de internet utilizados):

|  |  |
| --- | --- |
| **Recurso** | **Precio [€]** |
| Electricidad (0,20 €/kWh) | 46,87 |
| Internet (39,95 €/ mes) | 279,65 |
| Material de oficina | 6,00 |
| **TOTAL:** | **332,52** |

Tabla 147. Costes indirectos

Con todos los gastos ya definidos se muestra el **desglose final**. Se tiene en cuenta un margen de 10% para cualquier improvisto, así como un margen de ganancias de un 22% aplicado al coste del proyecto con margen de imprevistos aplicado:

|  |  |
| --- | --- |
| **Concepto** | **Precio [€]** |
| **Costes de personal** | 5.154,00 |
| **Costes de software** | 214,00 |
| **Costes de hardware** | 151,33 |
| **Costes indirectos** | 332,52 |
| **Subtotal** | 5.851,85 |
| **Margen de imprevistos (10%)** | 585,18 |
| **Subtotal con margen de imprevistos** | 6.437,03 |
| **Beneficio (22%) / Total sin IVA** | 7.724,43 |
| **TOTAL (IVA incluido):** | **9.346,56** |

Tabla 148. Costes totales

En total, el proyecto debería contar con un presupuesto de 9.346,56 euros.

# CONCLUSIONES

En este apartado se resumirá qué es lo que se ha conseguido de los **objetivos planteados** al inicio del proyecto, poniendo un punto y aparte en el desarrollo de este. También se exponen las conclusiones a las que se ha llegado, así como la experiencia personal en la realización del trabajo.

## 9.1. Conclusiones del proyecto

Teniendo en consideración el **objetivo principal** del proyecto, la motivación del desarrollo del software y el resto de los objetivos y submetas planteados en el punto [**4. Objetivos**](#_OBJETIVOS)**,** se puede afirmar que:

* Se ha conseguido crear una **adaptación realista** de la gran mayoría de funcionalidades del juego original Colonos de Catán.
* Se ha logrado utilizar la **plataforma BoardGame.io** y las características y librerías que presenta para las mecánicas y funcionalidad del juego.
* El proyecto ha sido correctamente desarrollado usando la **estructura ofrecida por React.js**, haciendo uso de características propias como los componentes o el estado.
* Se ha podido realizar un **análisis amplio** que abarque todas o casi todas las funcionalidades y necesidades del sistema.
* Se ha realizado un **diseño de sistema y de software** que cumple todos los requisitos de usuario y sistema planteados en un inicio en el apartado de análisis de sistema.
* Se ha realizado una **batería de pruebas** que valida y comprueba todos los requisitos funcionales del sistema.
* BoardGame.io almacena todas las acciones y turnos en un log accesible por lo que se puede **almacenar los datos** y copiarlos a un documento.

## 9.2. Conclusiones personales

Este proyecto ha supuesto un gran **reto personal y profesional** para mí, ya que ha sido mi primer gran proyecto realizado individualmente. Ha supuesto un gran número de horas de estudio previo puesto que no había realizado ningún trabajo con JavaScript o con React.js hasta la fecha.

Ha sido un largo camino de autodidactismo y búsqueda de información a través de internet y de conocidos y amigos, que ha desembocado en un resultado, a mi parecer, muy satisfactorio.

He necesitado introducirme en gran cantidad de foros, la mayoría en otros idiomas, e interactuar con los usuarios para poder solucionar dudas muy específicas e incluso contratar algún curso para acelerar mi aprendizaje de React y JavaScript.

Considero que el Catán es uno de los **mejores** juegos de mesa de tablero jamás creado y ha sido todo un privilegio poder realizar mi trabajo de fin de grado sobre este juego, así como poder orientarlo hacia un objetivo tan interesante como es la investigación del comportamiento humano en las interacciones de la Teoría de Juegos.

Es necesario volver a agradecer a todos los que me han **ayudado e inspirado** en el desarrollo del TFG, así como a mis dos tutores Javier y Alejandro por ayudarme y orientarme para poder dar lo mejor de mí y lograr sacar adelante este proyecto que a priori parecía simple, pero al final se convirtió en un proyecto que ha durado mucho más de lo que había anticipado.

Por último, BoardGame.io ha resultado ser una plataforma muy **divertida e interesante** para programar el juego. Sí que es verdad que tiene una comunidad muy pequeña y es complicado conseguir información avanzada acerca de ciertas funcionalidades de la plataforma, pero eso a su vez hace que sea mucho más gratificante lograr terminar un proyecto de esta magnitud, partiendo desde cero.

# POSIBLES AMPLIACIONES DEL PROYECTO

Este proyecto tiene varias **posibles expansiones** dependiendo del área que se quiera agrandar:

* Si se quiere ampliar la parte **mecánica y de entretenimiento** del juego, se pueden programar las expansiones que tiene el juego Colonos de Catán, así como las versiones de 5-6 jugadores para el juego base o sus expansiones.
* Sería interesante también trabajar en la parte de **agentes automáticos** y crear un bot que jugara de manera automática en la plataforma BoardGame.io. Este bot podría en un futuro jugar siguiendo una heurística o mediante aprendizaje automático. Usando los datos obtenidos en diferentes partidas se podría programar incluso una red neuronal que use como entradas los diferentes elementos del estado.
* Como bien se dijo en la motivación, la parte de **investigación** es una gran área en la que se puede trabajar. Con los datos obtenidos en el log a lo largo de varias partidas se pueden realizar diferentes estudios acerca del comportamiento humano en interacciones con otros jugadores en según qué circunstancias. Esta investigación podría fusionar parte de la Teoría de Juegos ya mencionada en el trabajo junto a la Inteligencia Artificial y los agentes automáticos.

# MARCO REGULADOR Y ENTORNO SOCIOECONÓMICO

En este apartado se describen las **leyes y reglas** a las que está sujeto este trabajo, así como las circunstancias socioeconómicas sobre las que se desarrolla.

## Marco regulador

### 11.1.1. Análisis de la legislación aplicable

El desarrollo de este proyecto se basa en el uso de JavaScript apoyado por la librería React, HTML y CSS. Por un lado, el uso de JavaScript es **completamente libre** y no tiene ninguna licencia ni restricción asociada.

HTML y CSS se tratan de lenguajes de **Formato abierto**, es decir, publicada a través de una organización sin ninguna restricción legal ni económica y por tanto de libre uso y distribución.

Por su parte, React cuenta con una **licencia MIT**, que se trata de una licencia aplicada al software libre permisiva, lo que implica capacidad de modificación, pocos límites en la reutilización y una buena compatibilidad de licencia.

Como plataforma de desarrollo, BoardGame.io cuenta también con una licencia MIT, por lo que tampoco es necesario registrar ninguna limitación legal.

Dentro de las regulaciones éticas y profesionales este proyecto se encuentra dentro del ámbito de **ingeniería de software**, y, por tanto, debe cumplir los cuatro principios en los que se basa cualquier proyecto de software:

1. **Confidencialidad:** Será necesario respetar la confidencialidad de los usuarios, clientes o empresa de la que forme parte.
2. **Competencia:** Será necesario ser realista y consciente de las habilidades y conocimientos que se posean y poder aceptar las responsabilidades que conllevan.
3. **Derechos de Copyright:** Será necesario ser conocedor de los derechos de propiedad intelectual asociados al proyecto y defenderlos correctamente.
4. **Uso legítimo del hardware:** Será necesario utilizar de forma correcta el equipo informático y no ocasionar problemas a los demás usuarios.

### 11.1.2. Propiedad intelectual

En cuanto a la **propiedad intelectual** es necesario diferenciar entre la propiedad intelectual que tiene el juego Colonos de Catán y la del software creado.

Por un lado, el juego Colonos de Catán cuenta con **licencia de copyright** de la empresa Catan GMBH, © CATAN GmbH 2021. Según se indica en el apartado de Copyright de su página web, el desarrollo de creación para uso personal o creaciones únicas hechas a mano son parte de la categoría “fair use” (uso legítimo) y por tanto está permitido su uso, aunque no su venta o comercialización.

Para cualquier duda, todos los derechos de distribución y uso del juego Colonos de Catán recaen sobre la empresa Catan GmbH y su director Klaus Teuber.

En cuanto al **desarrollo del software** en este proyecto, según el artículo 4 del apartado 4 de la ley de Patentes publicada en la Ley 24/2015 el 25 de Julio, se describe:

*“No se considerarán invenciones en el sentido de los apartados anteriores, en particular:*

*…*

*c) Los planes, reglas y métodos para el ejercicio de actividades intelectuales, para juegos*

*o para actividades económico-comerciales, así como los programas de ordenadores.”*

Según este artículo, el software generado por este proyecto **no** puede llegar a ser patentado ni a tener derechos de propiedad intelectual.

## Entorno socioeconómico

La parte económica del entorno de desarrollo del proyecto queda cubierta en el apartado de presupuestos y costes [**7.2 Presupuesto del proyecto**](#_7.2._Presupuesto_del)**.**

Por otro lado, el juego los Colonos de Catán, así como otros juegos de estrategia similares, buscan hacer una **labor lúdica y de entretenimiento** para la sociedad. Este es un factor clave para entender su importancia en la industria del entretenimiento actualmente.

Moralmente, este tipo de juegos ayudan a las personas a mejorar su comunicación y empatía por el resto de los jugadores, puesto que es necesario tener en cuenta los objetivos y situación de estos para poder jugar de forma óptima.

Adaptar este juego a una **versión digital** es una forma de hacer más accesible los beneficios que tiene jugar a este tipo de juegos multijugador, así como intentar acercar los juegos de tablero clásicos a nuevas generaciones.

# BIBILIOGRAFÍA

- [1] Miguel Ángel Álvarez. “Qué es el DOM”. Desarrollo Web. <https://desarrolloweb.com/articulos/que-es-el-dom.html> (accedido jul. 15, 2021)

- [2] Wikimedia Foundation. “JavaScript”. Wikipedia. <https://es.wikipedia.org/wiki/JavaScript> (accedido jul. 15, 2021)

- [3] Artículo sin autor. “JavaScript”. NeoAttack. <https://neoattack.com/neowiki/javascript/> (accedido jul. 15, 2021)

- [4] Artículo sin autor. “Qué es y cómo funciona React”. Next\_u. <https://www.nextu.com/blog/que-es-y-como-funciona-react-js/> (accedido jul. 15, 2021)

- [5] Artículo sin autor. “Empezando React”. React. <https://es.reactjs.org/docs/> (accedido jul. 15, 2021)

- [6] Artículo sin autor. “Componentes y Props”. React. <https://es.reactjs.org/docs/components-and-props.html> (accedido ago. 9, 2021)

- [7] Oscar Javier Blancarte. “Ciclo de vida de los Componentes”. Reactive Programming. <https://reactiveprogramming.io/blog/es/react/ciclo-de-vida-de-los-componentes> (accedido ago. 9, 2021)

- [8] Wikimedia Foundation. “Visual Studio Code”. Wikipedia. <https://es.wikipedia.org/wiki/Visual_Studio_Code> (accedido ago. 9, 2021)

- [9] Artículo sin autor. “Visual Studio Code Getting Started”. Visual Studio Code. <https://code.visualstudio.com/docs> (accedido ago. 9, 2021)

- [10] Artículo sin autor. “Vassal Overview”. Vassal. <http://www.vassalengine.org/wiki/Overview> (accedido ago. 10, 2021)

- [11] Artículo sin autor. “Tabletop Simulator, el videojuego que te permite crear un juego de mesa desde 0”. Generación Young. <https://www.generacionyoung.com/zona-gamers/noticias/tabletop-simulator-juego-mesa/> (accedido ago. 10, 2021)

- [12] Artículo sin autor. “Board Game Arena Studio”. Board Game Arena. <https://studio.boardgamearena.com/> (accedido ago. 10, 2021)

- [13] Artículo sin autor. “How to join BGA developer team?”. Board Game Arena. <https://en.doc.boardgamearena.com/How_to_join_BGA_developer_team%3F> (accedido ago. 10, 2021)

- [14] Artículo sin autor. “Concepts”. Boardgame.io. <https://boardgame.io/documentation/#/> (accedido ago.10, 2021)

- [15] Sousa, David, A. (2014) Neuroeducación y Juego de Mesa. Alianza Editorial. Recuperado a partir de: <http://devir.es/wp-content/uploads/2020/03/Neuroeducaci%C3%B3n-y-juegos-de-mesa.pdf>

- [16] Rodríguez, Luz Nancy. Aplicación de los Juegos de Mesa como Estrategia Pedagógica para facilitar el Aprendizaje en la Educación Inicial (Trabajo de Grado II). Facultad Ciencias Humanas y Sociales, Licenciatura en Pedagogía Infantil. Bogotá D.C. (septiembre 2014). Recuperado a partir de: <https://repositorio.ibero.edu.co/bitstream/001/624/1/Aplicaci%C3%B3n%20de%20los%20juegos%20de%20mesa%20como%20estrategia%20pedag%C3%B3gica%20para%20facilitar%20el%20aprendizaje%20en%20la%20educaci%C3%B3n%20inicial.pdf>

- [17] Wikimedia Foundation. “Teoría de Juegos”. Wikipedia. <https://es.wikipedia.org/wiki/Teor%C3%ADa_de_juegos> (accedido ago.10, 2021)

- [18] Daniela Ovadia. (2019). Teoría de Juegos. *Investigación y Ciencia.* Mente & Cerebro nº99. Disponible en: <https://www.investigacionyciencia.es/revistas/mente-y-cerebro/el-inconsciente-sale-a-la-luz-783/teora-de-juegos-18013>

- [19] Jeroen Geuze. (2006). Intelligent Tutoring Agent for Settlers of Catan. Disponible en: <https://ris.utwente.nl/ws/portalfiles/portal/19225046/Geuze06intelligent.pdf>

- [20] Wikimedia Foundation. “Klaus Teuber”. Wikipedia. <https://es.wikipedia.org/wiki/Klaus_Teuber> (accedido jul. 14, 2021)

- [21] Wikimedia Foundation. “Los Colonos de Catán” .Wikipedia. <https://es.wikipedia.org/wiki/Los_Colonos_de_Cat%C3%A1n> (accedido jul. 14, 2021)

- [22] Alex. “6 Juegos de mesa parecidos al Catán”. Entre juegos de mesa. <https://www.entrejuegosdemesa.es/blog/seis-juegos-de-mesa-parecidos-a-catan/> (accedido jul. 15, 2021)

# 

# 13. FUENTES

A continuación, se muestra el enlace al repositorio donde se encuentra todo el código del juego:

Github personal (Ic3Style): <https://github.com/Ic3Style/TFG>

También, en este apartado se muestran algunos sitios web y origen de los recursos utilizados en el desarrollo del software del proyecto.

* Pautas TFG redacción UC3M: <https://uc3m.libguides.com/TFG/escribir>
* Curso JavaScript Junior completo (utilizado para aprender JavaScript): <https://www.youtube.com/watch?v=z95mZVUcJ-E&list=PLZPWESMLLabrO9nKnrYSOHB0y3uGuqt--&index=7>
* Logo Universidad Carlos III de Madrid (utilizado para la memoria): <https://es.wikipedia.org/wiki/Archivo:Logo_UC3M.svg>
* The Fisher-Yates algorithm (utilizado para barajar arrays): <https://dev.to/codebubb/how-to-shuffle-an-array-in-javascript-2ikj#:~:text=The%20first%20and%20simplest%20way,)%20%3D%3E%200.5%20%2D%20Math>
* Documentación oficial BoardGame.io (utilizado como referencia): <https://boardgame.io/documentation/#/>
* Proyecto Camelot BoardGame.io (utilizado como referencia): <https://github.com/blunket/camelot>
* Proyecto Elevation of Privilege BoardGame.io (utilizado como referencia): <https://github.com/dehydr8/elevation-of-privilege>
* Proyecto Briscola BoardGame.io (utilizado como referencia): <https://github.com/aflorj/briscola>
* Tutorial 4 en raya BoardGame.io (utilizado para aprender BoardGame.io): <https://www.lonesomecrowdedweb.com/blog/four-in-a-row-boardgameio/>
* Tutorial BoardGame.io (utilizado como base): <https://boardgame.io/documentation/#/tutorial>
* Documentación oficial React.js (utilizado para aprender y dudas de React): <https://es.reactjs.org/docs/getting-started.html>
* Sitio web w3schools (utilizado para dudas CSS): <https://www.w3schools.com/>
* Código de colores HTML (utilizado para los colores de la interfaz): <https://htmlcolorcodes.com/es/>
* Documentación oficial de desarrollo web Mozilla (utilizado para dudas JavaScript): <https://developer.mozilla.org/es/docs>
* Imágenes de poblados, carreteras, ciudades y cartas (utilizado en el juego): <https://github.com/BryantCabrera/Settlers-of-Catan>
* Todos los iconos, salvo construcciones, usados en el juego (utilizado para los botones e iconos del juego): <https://www.flaticon.com/>
* Imágenes de la memoria – Google Images (utilizado para la memoria): <https://www.google.com/imghp?hl=EN>

# ANEXO A: GLOSARIO

**JS**. JavaScript

**W3C**. World Wide Web Consortium

**DOM**. Document Object Model

**HTML**. HyperText MarkUp Language

**XML**. Extensible MarkUp Language

**JSX**. JavaScript XML

**ECMAScript**. European Computer Manufacturers Association Script

**APP**. Application Portability Profile

**PVP**. Person Versus Person

**3D**. Three-Dimensional

**BGA**. Board Game Arena

**CSS**. Cascading Style Sheets

**PHP**. Hypertext Preprocessor

**SQL**. Structured Query Language

**IA**. Inteligencia Artificial

**GM.** Gran Maestro (Título de Ajedrez)

**CLIPS.** C Language Integrated Production System

**CU**. Caso de Uso

**RQC**. Requisito de Capacidad

**RQR**. Requisito de Restricción

**RQF**. Requisito Funcional

**RQN**. Requisito No Funcional

**ID**. Identity Document

**NPM**. Node Package Manager

**IVA**. Impuesto de Valor Añadido

**TFG**. Trabajo de Fin de Grado

**MIT.** Massachusetts Institute of Technology

# ANEXO B: SUMMARY

## B.1. Introduction

### B.1.1. Scope of the Project

This project is framed in the field of creation or adaptation of board games to a digitalized environment.

Catan, or in its full name, The Settlers of Catan, is a board game for 2 to 4 players in its standard version, whose objective is to collect 10 victory points through building, trading and obtaining achievements.

This project has as its base objective the incorporation of this traditional game, as well as its mechanics and interface, to a virtual environment through the Boardgame.io platform, which allows the creation of games through an environment based on JavaScript and with a base structure for the development of the game.

### B.1.2. Motivation

Nowadays, we live in an increasingly digitalized world, where the vast majority of games or sources of daily entertainment are played on a screen. However, traditional board games still survive and remain as popular as ever.

Therefore, and within a purely pragmatic motivation, for situations where you want to enjoy this board game, but do not have the necessary conditions, we have raised the adaptation to a virtual environment where you can enjoy the game only needing internet access.

On the other hand, the main motivation of this work has been to use Catán as a research platform to obtain data that can be used in the future in other works.

## B.2. State of Art

### B.2.1. Document Object Model (DOM)

The DOM is a structure of objects that is generated by the browser when a web page is loaded. It is modifiable through languages such as JavaScript which allows a dynamic page with a customizable appearance.

This type of model works through a hierarchy established between the objects that form it, that is to say, some objects depend on others forming what are known as DOM trees.

### B.2.2. Computer Environment

The code is programmed using Javascript, although not the flat version, but the React oriented one, that is JSX or JavaScript Syntax Extension.

JavaScript is one of the interpreted programming languages, that is, its instructions are read and executed one by one in real time, in this case through the browser from where it is executed.

React was also used for the project, which is a relatively new JavaScript library (created in 2010), created by Facebook, that allows the improvement of web pages through the creation and assembly of interactive and reusable components.

One of the great advantages of React is its ability to develop not only the client side, but also the server side, allowing greater control and freedom over the management of the website.

The entire code of the project has been written using Visual Studio Code. This is a text editor developed by Microsoft for various operating systems (including Windows and Linux) in 2015.

It is currently the most used text editor for programming Javascript and Node.js because of the facilities it brings and the amount of useful libraries for the user that it has.

### B.2.3. Specific platforms for computer games develpoment

As time goes by, new web pages or websites appear that allow users to create games using resources from the web itself or with documentation and basic tutorials so that the creator can develop his game in a customized way.

Usually these games are programmed using JavaScript or TypeScript, with the help of libraries such as React or environments such as Node.js.

Among the most important are Vassal, Tabletop Simulator, Board Game Arena and Boardgame.io.

**Vassal** is an engine created for the programming and subsequent online use of different turn-based games designed to play PvP (Person vs Person), i.e., only between humans.

**Tabletop Simulator** is a game in itself. It is a game developed by Berserk Games in 2015 (currently available on Steam) that allows the creation and use of other board games through a 3D simulation environment, where players can move with a free camera and interact with the pieces.

**Board Game Arena (BGA)** is an online platform through which hundreds of games created by the community can be accessed and played for free from the browser itself.

**BoardGame.io** is an online platform that allows the creation of online games through the use of a structural base and its own libraries. Any user can create a game for free by writing simple functions and making use of the state variables provided by the library. We have finally chosen to use this platform due to the simplified structure of its games, being ordered in the state variables and the possibility of making events and phases within the game, easily modifying these states.

### B.2.4. Articles about board games

Three papers related to the area have been selected:

**- Neuroeducation and Board Game:** this text is an innovative educational proposal to achieve an improvement in students' competencies developed by the publishing group Devir, based in Brazil. Basically, neuroeducation proposes a revolution in terms of the educational system, seeking to leave aside the traditional and ineffective system currently in place, which produces a high rate of failures and "robotized" subjects. The proposal mentions 3 groups of advantages or skills that can be improved with board games applied to education. Each of them can be related to a type of intelligence, as defined by Howard Gardner.

**- Application of Board Games as a Pedagogical Strategy:** This final degree project was carried out by Luz Nancy Rodríguez at the Faculty of Human and Social Sciences of the Corporación Universitaria Iberoamericana, in 2014. This work delves into the topic addressed by the previous educational proposal already mentioned, since it proposes a way to change the traditional educational system for one based on brain stimulation through self-realization. It is presented as an argument that board games possess indispensable characteristics to stimulate learning in any age range (although it focuses on children), such as a didactic intention and objective, teamwork, an established regulation and a reward system.

**- Game Theory:** is an area of mathematics. It investigates optimal strategies based on the behavior and actions of the players [17]. On this topic there are numerous scientific articles and related or applied works, although this time we will comment on the article published on the website Investigación y Ciencia, in November 2019 by the author Daniela Ovadia. In this article they comment that game theory exists in a context where cooperation and competition are mixed, since the decisions that a user makes do not only affect him. This theory is interesting because the possible actions taken by a user can and, in fact, are influenced by the possible actions taken by others.

## B.3. Board game with interaction between players

### B.3.1. Usefulness and relevance

Whether at a competitive or cooperative level, this type of game forces players to make decisions that will affect the state of their game, in such a way that a wrong decision made by a player can cause him to lose the game later on. This forces the players to think of a strategy, which not only involves the player himself but also the rest of the players present.

This factor and many others that require interaction between players makes users improve and develop their decision-making capacity, with the objective of their own benefit, that is, the area covered by Game Theory, more specifically in Cooperative Game Theory.

### B.3.2. Settlers of Catan

Settlers of Catan or simply Catan is a whole saga of games created by the German board game inventor Klaus Teuber in 1995. It is the first German-style game that has become popular outside the European continent, as it currently has translations in up to 18 different languages.

Currently the game has sold more than 20 million copies (2018 figure), generating millions of euros and winning numerous awards. The base game has an average duration of 1 hour, but this time can be lengthened depending on the players' familiarity with the game and decision making time, as well as the number of players or if any expansions are being used.

The game is based on the construction of villages, cities and roads in order to achieve victory points. A total of 10 victory points are needed to win, once someone gets them the game is automatically over.

### B.3.3. Other games in the área

Strategy games are currently one of the most popular and claimed areas among board games, since they allow exercising the players' decision-making capacity, as well as achieving a family or competitive atmosphere, depending on the player's preference.

Within the field in which Catán delves, there are other similar games that are worth mentioning, since they have helped in the boom of this entertainment sector.

**Istambul** is a game for 2 to 5 players with a strategic financial theme. In this game, with games lasting approximately half an hour, players must lead their group of pieces, composed of a trader and four assistants.

**Agricola** is a board game that places players in Europe in the year 1670. It is an environment where the player takes the role of a farmer and his wife, who will have to make decisions about the crops and how it affects the economy.

**High Tension** is a board game that puts the players to experience firsthand the management of an energy supply network. The players will have to mark a route by which they will grant energy to the cities and compete for the monopoly of this one.

**Isle of Skye** is set on the island of the same name, which in history is considered one of the most beautiful on the planet. In the game, each participant becomes a clan chief whose main objective will be to build a kingdom that will allow them to obtain as many points as possible.

**Carcassonne** is a game based on the placement of tiles to form a board. Players are set in the time of ancient France and must develop a turn-based strategy to decide where to place each piece on the terrain and whether it is worth placing a piece on top.

**Adventurers on the Train** is a game based on the competitiveness of the North American railroad company. It is an easy to learn game based on obtaining resources through cars that will be used to expand the railroad routes.

## B.4. Targets

This section shows the detailed objectives and sub-objectives that are intended to be achieved with the realization of this project.

**- Creation of a detailed adaptation of the board game The Settlers of Catán in the BoardGame.io platform.** It is the main objective of this work and it is composed of:

o Creation of the Catán mechanics through the BoardGame.io framework using React and JavaScript.

o Creation of an interactive interface for the Catan through JavaScript, React and CSS.

o (Optional) Creation of a server to be able to play Catan online from different devices.

o (Optional) Creation of a bot or AI capable of playing Catan on BoardGame.io.

**- Creation** **of an information storage system (log).** The aim is to store the information of the players of each turn in a text file.

**- Test and validation of each of the functionalities of the game.** The adaptation is intended to be as close as possible to the original version of the game.

**- Incorporation of the requirements raised in the analysis.** The game must comply with all user and system requirements defined in the system analysis.

## B.5. System analysis

This section will specify in detail the information system used to satisfy the needs of the users when using the proposed software. For this purpose, the different requirements that the programmed game must have will be defined, as well as the use cases that have been necessary to corroborate its correct operation.

A total of 16 use cases, 25 capacity user requirements, 7 restriction user requirements, 53 functional system requirements and 6 non-functional system requirements have been defined.

All system requirements have been created from a user requirement, so that all user requirements have been covered by at least one system requirement, achieving a design that covers all customer needs.

## B.6. System design

### B.6.1 Class diagram

The project files are divided into .js (JavaScript) extension for those intended for the operation and mechanics of the software, as well as some data files; and the .css (CSS) extension for those describing the interface of the corresponding .js file.

### B.6.2. Development environment

As for the Operating System, it has been decided to work on Windows. Since to launch the project it is only necessary to install the libraries and have access to a console, Windows has been chosen for its simplicity and ease of installing programs (Visual Studio Code).

Visual Studio Code has been chosen as the code editor. This editor has intrinsic facilities for JavaScript and React development, as well as the ability to connect to GitHub to carry out a version control system.

JavaScript has been chosen as the programming language. This language allows dynamic development of web pages, as well as numerous functionalities designed for web development.

For the structure of the software it was decided to use the React.js library, which allows a hierarchical organization and a more advanced development of the website.

Finally, it was decided to use BoardGame.io as the video game development platform.

### B.6.3. Function design

For the development of the functions, a documentation system has been used within the code itself. Each function has 3 sections to document it:

**- Status**: Indicates in which part of its development is the function. It can be IN PROCESS, FINISHED WITHOUT REVIEWING AND FINISHED REVIEWED.

**- To-Do:** In case the function is in the IN PROCESS status, here you would put what remains to be done to finish the function.

**- Function:** Here you briefly define the objective and what the function does.

The code is composed of more than 5000 lines of code and a total of 75 functions are used in the files Game.js, Board.js and Casilla.js.

### B.6.4. Software development and interface

For ease of explanation, it has been decided to separate the software into different parts.

**The construction of the board** is divided into two parts, as is the rest of the code: the functional or mechanical part and the interface or graphic design. The board consists of 19 hexagons with a terrain and an associated number.

In the game there are 3 types of interacting squares: the vertex squares or intersections, which are where towns and cities can be built; the road squares or edges, which are where roads can be built; and the terrain squares, which are the ones that grant resources and where the thief can be placed.

The interface is linked to the game variables G and ctx, so when you modify the state by clicking on a square, the game state is automatically updated thanks to React's simultaneous rendering.

**The actions** in this game are linked to the stage of the game or phase of the turn in which the player is, since there are certain situations where it is not possible to use all the available actions.

According to the structure proposed by BoardGame.io, the actions are called "moves" and unless a separate interface is created, the moves are accessed from the BoardGame interface itself, which appears when the game is launched on the right. However, this interface is "for developers", the ultimate goal is that the possible actions are accessed through the interface created for the game.

The buttons on the action bar can be clicked at any time during the game, but will only take effect if you are in the corresponding phase or stage.

**The card bars** are used to display on the screen the resource and development cards a player has. They are constantly updated so that when using development cards or spending resources on an action, the used cards disappear from the bar.

It has been decided to separate the cards into two bars, resource and development, since the card count for the game status only takes into account the resource cards.

**The status table** is an interface created to reflect some of the variables and information stored in the general variable G. It is a table with one row per player and two extra rows for mission cards.

## B.7. Battery of tests

This section describes and numbers the tests performed for the correct operation of the system and its requirements. All tests have been performed using the software developed for the work.

The objective of the tests is to cover all the system requirements in a satisfactory way. In total 30 tests have been created, which fully cover the 53 functional system requirements raised in the system analysis.

## B.8. Project Management

The project was agreed and closed on October 26, 2020 with the tutors Javier Ignacio Carbo Rubiera and Alejandro Cervantes Rovira, although it was not until January 11, 2021 that the development of the project began.

The project was divided into two main areas to develop: code and documentation.

It was decided not to carry out a simultaneous development of the two areas because much of the memory depends on the work done and during the first months it was somewhat uncertain the way to go for its realization.

The development of the report started in mid-May 2021, once the project was largely developed and the objectives had been clarified.

## B.9. Conclusions

### B.9.1. Project conclusions

Taking into consideration the main objective of the project, the motivation of the software development and the rest of the objectives and subgoals stated in the objectives, it can be affirmed that:

- It has been possible to create a realistic adaptation of most of the functionalities of the original Settlers of Catan game.

- It has been possible to use the BoardGame.io platform and its features and libraries for the mechanics and functionality of the game.

- The project has been correctly developed using the structure offered by React.js, making use of its own features such as components or state.

- It has been possible to perform a comprehensive analysis that covers all or almost all the functionalities and needs of the system.

- A system and software design has been made that meets all the user and system requirements raised at the beginning in the system analysis section.

- A battery of tests has been performed to validate and verify all the functional requirements of the system.

### B.9.2. Personal conclusions

This project has been a great personal and professional challenge for me, as it has been my first big project done individually. It has meant a great number of hours of previous study since I had not done any work with JavaScript or React.js to date.

It has been a long way of self-education and search for information through the internet and friends and acquaintances, which has led to a result, in my opinion, very satisfactory.

I have needed to enter a lot of forums, most of them in other languages, and interact with users to solve very specific doubts and even hire a course to accelerate my learning of React and JavaScript.

Finally, BoardGame.io has turned out to be a very fun and interesting platform to program the game. It is true that it has a very small community and it is difficult to get advanced information about certain features of the platform, but that in turn makes it much more rewarding to finish a project of this magnitude, starting from scratch.

## B.10. Posible extensions of the project

This project has several possible expansions depending on the area to be enlarged:

- If you want to expand the mechanical and entertainment part of the game, you can program the expansions that the Settlers of Catan game has, as well as the 5-6 player versions for the base game or its expansions.

- It would also be interesting to work on the part of automatic agents and create a bot that would play automatically on the BoardGame.io platform. This bot could in the future play following a heuristic or by machine learning. Using the data obtained in different games, it could even program a neural network that uses the different elements of the state as inputs.

- As well said in the motivation, the research part is a great area to work on. With the data obtained in the log over several games, different studies can be carried out about human behavior in interactions with other players in different circumstances. This research could merge part of the Game Theory already mentioned in the paper together with Artificial Intelligence and automatic agents.