SCRUM

Grupo PMC

22 de mayo de 2022

LA METODOLOGÍA SCRUM

Durante este segundo cuatrimestre el grupo PMC nos hemos dedicado a adaptar a la era digital el tradicional juego de mesa *Rolit*, convirtiéndose en nuestro primer proyecto auto-organizado y de temática libre.

La magnitud del proyecto, el número de integrantes del equipo y los requerimientos de la asignatura de Ingeniería del Software II hacen que sea necesario establecer una metodología de trabajo eficaz para las características de nuestro equipo. En nuestro caso nos hemos organizado dentro del marco SCRUM.

En este documento expondremos cómo hemos materializado los principios de esta técnica de trabajo a lo largo del tiempo de desarrollo, mostrando nuestra evolución y aprendizaje.

Comencemos hablando sobre los fundamentos de SCRUM para poder contextualizar correctamente nuestro trabajo dentro de un marco teórico.

SCRUM se trata de una de las metodologías ágiles para desarrollo de proyectos más extendidas en el mundo laboral y se caracteriza por su adaptabilidad a los cambios, así como por la gran visibilidad que ofrece al cliente durante la etapa de desarrollo.

Así pues, esta técnica de trabajo ofrece un enfoque iterativo e incremental. Estas iteraciones suelen tener una duración media de dos semanas y cada una de ellas recibe el nombre de *sprint*. En cada *sprint*, el equipo de desarrollo se compromete a completar una lista de requisitos que conforman, junto a posibles tareas internas, el llamado *Sprint Backlog*.

Los requisitos pueden ir variando en el tiempo, de manera que surgan nuevas características, se modifiquen las ya existentes o se decida eliminar alguna. Estos requerimientos, que en la nomenclatura de SCRUM se llaman *Historias de Usuario*, se almacenan en el *Product Backlog*. A diferencia del *Product Backlog*, el *Sprint Backlog* debe permanecer inalterado durante la ejecución del *sprint*.

Además, existe una alta comunicación entre los miembros del equipo de desarrollo gracias al *Daily Scrum*, una reunión diaria de aproximadamente 15 minutos en la que se habla sobre la evolución del proyecto. En la misma línea se sitúan los Sprint Retrospectives y Sprint Reviews que reflexionan sobre el trabajo realizado en el sprint.

Para el correcto funcionamiento de esta metodología se precisa de personas que se dediquen expresamente a cerciorarse de ello, así como unos desarrolladores expertos, dando lugar, de forma natural, a una estructura no jerarquizada para los equipos SCRUM.

ESTRUCTURA Y FUNCIONAMIENTO

Como ya hemos adelantado en la sección anterior, de los principios básicos de SCRUM surge la necesidad intrínseca de establecer una estructura de equipo muy concreta.

Los grupos que se regulan según esta técnica son auto-organizados, de forma que los propios integrantes son quienes deciden como distribuir y realizar su trabajo, sin la participación de agentes externos. Esta característica, ligada a la multifuncionalidad exigida a los desarrolladores, obliga a que los miembros del equipo sean personas con experiencia.

Aunque en SCRUM no existen figuras que tengan una mayor autoridad que otras, sí que nos encontramos con distintos cargos que se encargan de unas responsabilidades específicas.

Product Owner

El *Product Owner* tiene la responsabilidad de maximizar el valor del producto resultante del trabajo del equipo de desarrollo. Las formas de conseguir este objetivo dependen cada organización, *Scrum Team* e individuos.

Además, el *Product Owner* es el encargado de la gestión eficiente del *Product Backlog*, que incluye:

- El desarrollo y comunicación explícita del producto final.
- Comunicar claramente los cambios y la creación de elementos del *Product Backlog*.
- Mantener el *Product Backlog* ordenado, limpio, visible y bien explicado.

En términos informales podría considerarse que el *Product Owner* es la "voz del cliente", pues es quien se comunica con ellos y transforma sus ideas en productos tangibles.

Siguiendo el espíritu democrático de SCRUM, al comienzo del cuatrimestre, PMC escogió a Virginia Chacón Pérez como *Product Owner*.

Scrum Master

El *Scrum Master* es quien se encarga de promocionar y mantener SCRUM. Esta tarea se realiza asegurándose que todas las personas entiende las reglas, valores y técnicas de SCRUM para mejorar el flujo de trabajo del equipo.

Así, las habilidades interpersonales y la capacidad de ayudar a los miembros del equipo a crecer y mejorar son esenciales para el *Scrum Master*. Entre las obligaciones de esta figura se pueden destacar:

- Organizar las reuniones de planificación de los sprints.
- Organizar las Daily Scrums.
- Eliminar cualquier obstáculo que dificulte el desarrollo del proyecto.

Una vez más, este puesto fue sometido a votación y los miembros de PMC decidieron que Leonardo Macías Sánchez ocupara el cargo.

Development Team

El equipo de desarrollo está formado por el conjunto de personas que se dedican a desarrollar cualquier aspecto de un incremento usable en cada *sprint*. El Development Team de este proyecto está formado por:

- Juan Diego Barrado Daganzo
- Virginia Chachón Pérez
- Sergio Miguel García Jiménez
- Daniel González Arbelo
- Leonardo Macías Sánchez
- María del Mar Ramiro Ortega

HISTORIAS DE USUARIO

PRODUCT BACKLOG

ÉPICA 1
"Como usuario quiero poder jugar a Rolit..."

Historia de usuario 1: ... siguiendo un conjunto mínimo de normas

Fase descriptiva: se establece un conjunto de normas que se deberán cumplir durante el juego.

Prioridad: alta, las normas del juego son la base para crear el mismo.

Tamaño estimado: 1 sprint para la base, aunque se irá mejorando a lo largo del proyecto.

Condiciones de aceptación: el juego es totalmente funcional con el conjunto de normas especificado.

Explicación detallada Esta historia de usuario fue concebida como tal y ejecutada en el primer sprint. Los *bullet points* a continuación enumerados forman parte de la historia de usuario homónima en el Sprint Backlog 1. Esta historia de usuario conlleva implementar las siguientes normas en el juego:

- Consta de un tablero cuadrado de 8x8 casillas
- Consta de hasta 4 jugadores con colores asignados: amarillo, rojo, verde y azul.
- Cada usuario puede elegir su nombre como nickname al inicio de la partida.
- Las piezas del juego son cubos redondos de colores que se introducen en las casillas del tablero.
- Solo se puede colocar un cubo redondo en una casilla vacía adyacente a una ya ocupada, excepto el primer cubo que se coloca en la posición que se desee.

- Un cubo se dice atrapado, cuando se encuentra en la línea que une un cubo recién puesto con otro del mismo color.

- Las direcciones válidas son las líneas rectas dadas por alguna de las casillas adyacentes.

- Cuando se coloca un cubo, se cambian de color todos los cubos atrapados en las direcciones válidas.

- En cada turno un jugador pone un único cubo y se pasa turno al siguiente jugador después de haber cambiado de color los posibles cubos atrapados.

- El juego acaba cuando el tablero se llena completamente.

- Gana el jugador que más bolas de su color tenga sobre el tablero cuando se termina el juego.

Historia de usuario 2: ... con una interfaz agradable

Frase descriptiva: tener una interfaz gráfica que permita al usuario interactuar con el juego de manera sencilla e intuitiva.

Prioridad: media, no es imprescindible para el funcionamiento del juego.

Tamaño estimado: 3 sprints.

Condiciones de aceptación: tener una interfaz gráfica funcional y sin bugs y que pueda ser utilizada por cualquier usuario externo al proyecto.

Explicación detallada Esta historia de usuario surge fruto de juntar las deprecadas historias de usuario:

- *Como usuario, me gustaría que se pudiesen personalizar los colores con los que jugamos cada jugador porque hace más visual el juego.*

  Dicha "historia de usuario ", que a la luz de esta refactorización consideramos "tarea " de esta nueva historia de usuario superior, fue implementada en el Sprint 2. Las tareas descritas en el Sprint Backlog 2 pasan a constituir las siguientes subtareas:

  - Los jugadores deben poder elegir entre los siguientes colores: rojo, amarillo, azul, rosa, verde, morado, negro, naranja, marrón y beige.

- *Como usuario, me gustaría que tenga una interfaz gráfica amable porque hace más fácil jugar.*

ÉPICA 2
"Como usuario quiero que Rolit introduzca características innovadoras..."

Historia de usuario 3: ... pensando en las posibilidades que brinda el multijugador

Frase descriptiva: tener varias modalidades de juego teniendo en cuenta el número de jugadores, así como diferentes tamaños, formas para el tablero, la posibilidad de jugar en red, o con inteligencias artificiales.

Prioridad: media.

Tamaño estimado: 5 sprints.

Condiciones de aceptación: se puede elegir entre varios tamaños y formas para el tablero, y además las modalidades de juego por equipos, jugador automático y clásica son completamente funcionales tanto al jugar con inteligencias artificiales como en red.

Explicación detallada Esta historia de usuario surge fruto de juntar las deprecadas historias de usuario:

- *Como usuario, me gustaría que el número de jugadores fuese variable porque así se adapta a diferentes grupos de personas.*

  Dicha "historia de usuario ", que a la luz de esta refactorización consideramos "tarea " de esta nueva historia de usuario superior, fue implementada en el Sprint 2. Las tareas descritas en el Sprint Backlog 2 pasan a constituir las siguientes subtareas:

  - Cuando el usuario decida crear una nueva partida, se le debe preguntar por el número de jugadores para dicha partida.
  - El número mínimo de jugadores es 2.
  - El número máximo de jugadores es 10.

- *Como usuario, me gustaría que el tamaño del tablero sea variable porque se adapta mejor a los grupos variables.*

  Dicha "historia de usuario", que a la luz de esta refactorización consideramos "tarea" de esta nueva historia de usuario superior, fue implementada en el Sprint 2. Las tareas descritas en el Sprint Backlog 2 pasan a constituir la siguiente subtarea:

  - se puede elegir tamaño entre: pequeño (9x9), mediano (13x13), grande (17x17).

- *Como usuario, me gustaría que la forma del tablero fuese variable para que cambiasen las estrategias.* Ahora hay diferentes formas del tablero, cada una de ellas con los tres tamaños mencionados previamente, estas son:

    - Círculo
    - Rombo
    - Cuadrado

- *Como usuario, me gustaría que se pudiera jugar en red.* Existe la posibilidad de jugar tanto en tu propio ordenador como de jugar en red contra otros usuarios.

- *Como usuario, me gustaría que se pudiera jugar contra una inteligencia artificial, así como que ellas jugaran solas.* Será posible jugar contra inteligencias artificiales, estas cuentan como jugadores, es decir, podrá el número de inteligencias sumado al número de personas debe sobrepasar 2 y no ser más de 10.

- *Como usuario, me gustaría que hubiese modalidades de juego sobre rolit:*

    - *Clásica*
    - *Por equipos*

Historia de usuario 4: ... siendo intuitivo y cómodo de jugar

Frase descriptiva: tener un tutorial al inicio del juego, poder cargar y guardar partida, poder ver la repetición de una partida paso a paso y poder salir del juego en cualquier momento durante la ejecución del mismo.

Prioridad: baja, son elementos adicionales a la funcionalidad del juego.

Tamaño estimado: 4 sprints

Condiciones de aceptación: existe la posibilidad de ejecutar un tutorial al inicio del juego para entender las normas, las opciones de cargar y guardar partida se ejecutan correctamente en cualquier momento del juego, y al salir del juego en cualquier momento no se lancen excepciones y funcione correctamente.

Explicación detallada Esta historia de usuario surge fruto de juntar las deprecadas historias de usuario:

- *Como usuario, me gustaría que hubiese un tutorial para entender bien las normas y excepciones del juego.*

- *Como usuario, me gustaría que se pudiera guardar repeticiones de partida para poder revisarlas más tarde.*

- *Como usuario, me gustaría que se pudiese guardar y cargar partida para continuar más tarde porque permite poner en pausa el juego.* Ya deprecado antes de esta refactorización, pero fue implementada en el Sprint Backlog 1 bajo las siguientes características:

    - Se debe poder guardar la partida y salir del juego en cualquier momento durante la ejecución del mismo.

    - Al iniciar el juego, debe poderse elegir entre cargar una partida guardada o iniciar una nueva.
    - Se pueden tener varias partidas guardadas
    - Para cargar una partida, se muestra una lista para poder elegir la partida a continuar.
    - Para guardar partida, se debe elegir un nombre identificativo para denotar a dicha partida.

- *Como usuario, me gustaría poder guardar y cargar distintas partidas, eligiendo el nombre del fichero donde se cargan/guardan.* Se trata de un refinamiento del punto anterior que se introdujo sin dificultades en el Sprint 2.

- *Como usuario, me gustaría poder salir del juego en cualquier momento (introducida el 17 de febrero de 2022)*

  Dicha "historia de usuario ", que a la luz de esta refactorización consideramos "tarea " de esta nueva historia de usuario superior, fue implementada en el Sprint 2. Las tareas descritas en el Sprint Backlog 2 pasan a constituir las siguientes subtareas:

    - La partida debe finalizar cuando cualquier jugador así lo indique.
    - Se usará un comando *exit* para llevar a cabo esta funcionalidad.

REGISTRO DE CAMBIOS

Historias iniciales

- Como usuario, me gustaría que el número de jugadores fuese variable porque así se adapta a diferentes grupos de personas.

- Como usuario, me gustaría que se puediesen personalizar los colores con los que jugamos cada jugador porque hace más visual el juego.

- Como usuario, me gustaría que hubiese tableros distintos para poder tener distintas estrategias a la hora de jugar.

    - Como usuario, me gustaría que el tamaño del tablero sea variable porque se adapta mejor a los grupos variables.
    - Como usuario, me gustaría que la forma del tablero fuese variable para que cambiasen las estrategias.

- Como usuario, me gustaría que hubiese varias modalidades de juego sobre rolit:

    - Clásica
    - Modalidades por tiempo
    - Por equipos

    - Trampa del Ordenador (El ordenador me cambia las cosas)
    - Trampa de cada jugador (Cada jugador tiene un número fijo de oportunidades de trampa)
    - Bolas especiales.

- Como usuario, me gustaría que se pudiese guardar y cargar partida para continuar más tarde porque permite poner en pausa el juego.
- Como usuario, me gustaría poder poner mi nombre como nickname durante la partida.
- Como usuario, quiero que al final se muestre un ranking de puntos de la partida.
- Como usuario, me gustaría que tenga una interfaz gráfica amable porque hace más fácil jugar.
- Como usuario, me gustaría que se pudiera guardar repeticiones de partida para poder revisarlas más tarde.
- Como usuario, me gustaría que hubiese un tutorial para entender bien las normas y excepciones del juego.
- Como usuario, me gustaría jugar al rolit siguiendo un conjunto mínimo de normas (mirar normas rolit).

Cambios Sprint 1

Historias añadidas

- Como usuario, me gustaría poder salir del juego en cualquier momento.

Cambios Sprint 2

Historias modificadas

- Como usuario, me gustaría que se pudiese guardar y cargar partida para continuar más tarde porque permite poner en pausa el juego.
- Como usuario, me gustaría poder guardar y cargar distintas partidas, eligiendo el nombre del fichero donde se cargan/guardan.

Cambios Sprint 3

Historias modificadas

- Como usuario, me gustaría que hubiese varias modalidades de juego sobre rolit:

- Clásica
- Modalidades por tiempo
- Por equipos
- Trampa del Ordenador (El ordenador me cambia las cosas)
- Trampa de cada jugador (Cada jugador tiene un número fijo de oportunidades de trampa)
- Bolas especiales.

Historias añadidas

- Como usuario, me gustaría que hubiese varias modalidades de juego sobre rolit:
  - Clásica
  - Por equipos
  - Contra la CPU
  - Online

Cambios Sprint 4

Historias modificadas

- Como usuario, me gustaría que hubiese varias modalidades de juego sobre rolit:
  - Clásica
  - Por equipos
  - Contra la CPU
  - Online

Historias añadidas

- Como usuario quiero que Rolit introduzca características innovadoras pensando en las posibilidades que brinda el multijugador:
  - Como usuario, me gustaría que se pudiera jugar en red.
  - Como usuario, me gustaría que se pudiera jugar contra una inteligencia artificial, así como que ellas jugaran solas.

Cambios Sprint 5

En el sprint 5 fueron refactorizadas completamente todas las historias de usuario, manteniendo solo 2 épicas con 2 historias de usuario cada una. Nos adaptamos al formato de SCRUM, incluyendo frase descriptiva, prioridades, tamaños, condiciones de aceptación y explicación detallada. El resultado puede verse en:

SPRINTS

SPRINT 1

Revisión del Sprint

Respecto a los objetivos esperados del Sprint Backlog, se ha implementado adecuadamente la funcionalidad de "*Como usuario, me gustaría jugar al rolit siguiendo un conjunto mínimo de normas*". El usuario es capaz de desarrollar con normalidad una partida del juego rolit sin bugs conocidos.

En cuanto a "*Como usuario, me gustaría poder guardar y cargar partida.*", la funcionalidad se ha implementado correctamente, aunque el método es rudimentario y está obligado a futuras modificaciones para aportarle robustez.

El usuario puede elegir con libertad los nicknames que prefiera para jugar a rolit, luego la funcionalidad "*Como usuario, me gustaría poder poner mi nombre como nickname durante la partida*" está implementada satisfactoriamente.

Al finalizar la partida, se muestra un ranking que permite ver los jugadores y sus puntuaciones, lo que da por satisfecha la funcionalidad "*Como usuario, quiero que al final se muestre un ranking de puntos de la partida*".

Retrospectiva

More Of

- Es necesario que los miembros del equipo adquieran más experiencia con el manejo de Git y GitHub, así como los flujos de trabajo para Sistemas de Control de Versiones Descentralizados.

- Aumentar el grado de comunicación entre los miembros del equipo y la periodicidad de la misma.

Keep Doing

- Trabajar de forma paralela y asignar una tarea (no siempre la misma para tener una visión global del proyecto) a grupos de entre los miembros del equipo ha agilizado la producción de software.
- Mantener a todos los miembros al tanto de los cambios, modificaciones o direcciones que toma el proyecto a lo largo del proyecto.

Start Doing

- Fijar dos días a la semana en los que podamos reservar un aula de trabajo en grupo para realizar avances del proyecto de forma paralela a los días de clase.
- Construir un flujo de trabajo definido que se adapte bien al formato distribuido de Git y que permita trabajar de forma paralela entre miembros con distintas funcionalidades.

Stop Doing

- Hacer una planificación deficiente del Sprint en cuanto a definición clara sobre que se va hacer durante el mismo.

Less of

- Es necesario dedicarle más tiempo al diseño de la práctica que a la programación de la misma porque ahorra tiempo y mejora la calidad de trabajo y del software.

Planificación del siguiente Sprint

Durante el siguiente Sprint, el equipo de desarrollo llevará a cabo las siguientes historias de usuario del Product Backlog:

- *Como usuario, me gustaría que el número de jugadores fuese variable para adaptarse mejor a diferentes grupos de personas.*
- *Como usuario, me gustaría que se pudiesen personalizar los colores con los que jugamos cada uno para hacer más visual el juego.*
- *Como usuario, me gustaría poder salir del juego en cualquier momento.*
- *Como usuario, me gustaría que hubiese distintos tamaños de tablero seleccionables.*

Además, durante este Sprint el trabajo se va a distribuir de la siguiente manera:

- En primer lugar, se dedicará una gran parte del esfuerzo a formar un diseño robusto de la aplicación, para mejorar la calidad del código existente.
- Se solventarán las deudas de código existentes de la versión anterior.
- Una vez hecho el diseño correspondiente, una parte del equipo se dedicará a la modificación del código para adaptarse al nuevo diseño y la otra a generar el código para las nuevas funcionalidades.

SPRINT 2

Revisión del Sprint

Respecto a los objetivos esperados del Sprint Backlog, se ha implementado adecuadamente la funcionalidad de "*Como usuario, me gustaría poder guardar y cargar distintas partidas.*". El usuario es capaz de guardar un instante de una partida en cualquier momento y darle un nombre. Para cargar la partida, hay que introducir el nombre del archivo al seleccionar "cargar partida".

En cuanto a "*Como usuario, me gustaría poder salir del juego en cualquier momento.*", se ha introducido al juego un comando exit que permite llevar a cabo esta funcionalidad.

El usuario puede elegir con libertad los nicknames que prefiera para jugar a rolit y su color, así como el número de jugadores, luego las funcionalidades "*Como usuario, me gustaría que el número de jugadores fuese variable para adaptarse mejor a diferentes grupos de personas.*" y "*Como usuario, me gustaría que se pudiesen personalizar los colores con los que jugamos cada uno para hacer más visual el juego.*" están implementadas satisfactoriamente.

Además, al comenzar una partida nueva se le da al usuario la opción de elegir el tamaño del tablero, por lo que la historia de usuario "*Como usuario, me gustaría que hubiese distintos tamaños de tablero seleccionables.*" también ha sido ejecutada.

Paralelamente a la implementación de nuevas historias de usuario, hemos refactorizado gran parte del código para facilitar su desarrollo y manejo de cara a futuros sprints. Los cambios se pueden consultar en el documento *RefactorizacionSprint1* dentro de la carpeta *DocumentosDesarrollo*.

Retrospectiva

More Of

- Es necesario que los miembros del equipo adquieran más experiencia con el manejo de Git y GitHub, así como los flujos de trabajo para Sistemas de Control de Versiones Descentralizados. Se observa una mejora notable con respecto al sprint anterior.

- Reservar más tiempo para la depuración del código.

- Mejorar el flujo de trabajo definido para que se adapte bien al formato distribuido de Git y que permita trabajar de forma paralela entre miembros con distintas funcionalidades.

Keep Doing

- Trabajar de forma paralela y asignar una tarea (no siempre la misma, para tener una visión global del proyecto) a grupos de entre los miembros del equipo ha agilizado la producción de software.
- Mantener a todos los miembros al tanto de los cambios, modificaciones o direcciones que toma el proyecto a lo largo del proyecto.
- Mantener el grado de comunicación entre los miembros del equipo y la periodicidad de la misma.
- Continuar con las reuniones dos días a la semana en los que reservamos un aula de trabajo en grupo para realizar avances del proyecto de forma paralela a los días de clase.

Start Doing

- Fijar con antelación una reunión para llevar a cabo el Sprint Review.

Stop Doing

Less of

- Deberíamos dedicarle algo menos de tiempo al código en favor de la documentación.

Planificación del siguiente Sprint

Durante el siguiente Sprint, el equipo de desarrollo llevará a cabo las siguientes historias de usuario del Product Backlog:

- *Como usuario, me gustaría que se pudiera guardar repeticiones de partida para poder revisarlas más tarde.*
- *Como usuario, me gustaría que se pudiese jugar a la versión por equipos de Rolit.*
- *Como usuario, me gustaría que se hubiese distintas formas de tableros seleccionables.*

Además, durante este Sprint el trabajo se va a distribuir de la siguiente manera:

- En primer lugar, se discutirá la necesidad de una refactorización de Controller, y se ejecutará en caso de que así se acuerde.
- Se solventarán algunas deudas de código existentes de la versión anterior.

- El equipo se dedicará generar el código para las nuevas funcionalidades. En el tiempo restante se tratará de mejorar algunas características del sprint anterior.

SPRINT 3

Revisión del Sprint

En primer lugar, la funcionalidad "*Como usuario, me gustaría que se pudiera guardar repeticiones de partida para poder revisarlas más tarde.*"ha sido implementada exitosamente. Resulta cómodo revisar repeticiones, tal y como se deseaba.

Aparte, el cargado de partidas al inicio de la ejecución es ahora más cómodo y amigable con el usuario. Al seleccionar la opción de cargar partida se muestra una lista con las partidas disponibles para ser cargadas, de forma que el usuario no tiene que conocer el nombre de los ficheros en los que han sido guardadas previamente.

Hemos podido dedicar también una buena parte del esfuerzo de este Sprint a la refactorización de código. Esto ha sido muy útil y lo seguirá siendo a medida que sigamos avanzando con el proyecto, pues nos ha facilitado en gran medida el desarrollo del código, lo ha hecho más entendible y hemos conseguido una mayor modularidad, de manera que implementar cambios ahora es mucho más sencillo.

Por otro lado, la funcionalidad "*Como usuario, me gustaría que hubiese distintas formas de tableros seleccionables.*"también se ha logrado como esperábamos. Ahora hay tres formas para elegir: cuadrado, círculo y rombo. Aparte, para cada una de estas formas hay tres tamaños de tableros: pequeño, mediano y grande.

Finalmente, hay una funcionalidad que no hemos podido implementar en este Sprint, que es la funcionalidad de "*Como usuario, me gustaría que se pudiese jugar a la versión por equipos de Rolit.*". Una causa de esto ha sido que hemos dedicado bastante tiempo de este Sprint a desarrollar UML, por lo que, a pesar de que habría sido posible implementar esta funcionalidad, habría sido a coste de hacerlo con un código menos modular, perdiéndose en parte lo logrado con la refactorización. Por tanto, hemos considerado más conveniente dejar esta funcionalidad de cara al Sprint siguiente, puesto que lo podremos hacer mejor.

Retrospectiva

More Of

- Es necesario que los miembros del equipo adquieran más experiencia con el manejo de Git y GitHub, así como los flujos de trabajo para Sistemas de Control de Versiones Descentralizados. Se observa una mejora notable con respecto al sprint anterior.

- Mejorar el flujo de trabajo definido para que se adapte bien al formato distribuido de Git y que permita trabajar de forma paralela entre miembros con distintas funcionalidades.

Keep Doing

- Trabajar de forma paralela y asignar una tarea (no siempre la misma, para tener una visión global del proyecto) a grupos de entre los miembros del equipo ha agilizado la producción de software.
- Mantener a todos los miembros al tanto de los cambios, modificaciones o direcciones que toma el proyecto a lo largo del proyecto.
- Mantener el grado de comunicación entre los miembros del equipo y la periodicidad de la misma.
- Continuar con las reuniones dos días a la semana en los que reservamos un aula de trabajo en grupo para realizar avances del proyecto de forma paralela a los días de clase.
- Seguir asignando suficiente tiempo para la depuración del código.

Start Doing

- Fijar con antelación una reunión para llevar a cabo el Sprint Review.
- Mantener más reuniones a lo largo de la semana, independientemente del número de integrantes que puedan asistir.

Stop Doing

Less of

- Deberíamos dedicarle algo menos de tiempo al código en favor de la documentación.

Planificación del siguiente Sprint

Durante el siguiente Sprint, el equipo de desarrollo llevará a cabo las siguientes historias de usuario del Product Backlog:

- *Como usuario, me gustaría que se pudiese jugar a la versión por equipos de Rolit..*
- *Como usuario, quiero poder jugar a Rolit con una interfaz agradable.*

Además, durante este Sprint el trabajo se va a distribuir de la siguiente manera:

- En primer lugar, se crearán grupos a los que asignarles estas historias de usuario, y otro grupo para desarrollar los tests de JUnit.
- Se trabajará en saldar la deuda de documentación pendiente desde los Sprints anteriores.

SPRINT 4

Revisión del Sprint

En primer lugar, la funcionalidad de "*Como usuario, me gustaría poder jugar a la versión de Rolit por equipos*" ha sido completada con éxito. La refactorización completa de la parte de la lógica del juego ha permitido que incluir el resto de modos de juego no modifique en absoluto la distribución de responsabilidades que se ha hecho hasta ahora. La inclusión de los Builders ha posibilitado que la creación de los Game quede completamente al margen del resto de clases, que simplemente manejan una abstracción del tipo Game.

La funcionalidad de "*Como usuario, me gustaría poder jugar a Rolit con una interfaz agradable*" ha sido completada con éxito en su fase Beta. Nos permite depurar y experimentar con esta distribución de componentes hasta tener un diseño sólido para futuras mejoras.

Retrospectiva

More Of

- Es necesario que los miembros del equipo adquieran más experiencia con el manejo de Git y GitHub, así como los flujos de trabajo para Sistemas de Control de Versiones Descentralizados. Se observa una mejora notable con respecto al sprint anterior.
- Es necesario poner al día los diagramas UML del proyecto.

Keep Doing

- Trabajar de forma paralela y asignar una tarea (no siempre la misma, para tener una visión global del proyecto) a grupos de entre los miembros del equipo ha agilizado la producción de software.
- Mantener a todos los miembros al tanto de los cambios, modificaciones o direcciones que toma el proyecto a lo largo del proyecto.
- Mantener el grado de comunicación entre los miembros del equipo y la periodicidad de la misma.
- Continuar con las reuniones dos días a la semana en los que reservamos un aula de trabajo en grupo para realizar avances del proyecto de forma paralela a los días de clase.
- Seguir asignando suficiente tiempo para la depuración del código.
- Fijar con antelación una reunión para llevar a cabo el Sprint Review.

- Seguir haciendo reuniones de debate sobre cuál es el mejor diseño antes de empezar a programar.

Start Doing

Stop Doing

- Deberíamos dejar de hacer la documentación lo último porque tener archivos de desarrollo facilita que todo el equipo esté al tanto del estado de cierta funcionalidad y lo pueda tener en cuenta en sus diseños.

Less of

- Deberíamos dedicarle algo menos de tiempo al código en favor de la documentación.

Planificación del siguiente Sprint

Durante el siguiente Sprint, el equipo de desarrollo llevará a cabo las siguientes tareas de la historia de usuario del Product Backlog *Como usuario quiero que Rolit introduzca características innovadoras pensando en las posibilidades que brinda el multijugador*:

- *Como usuario, me gustaría que se pudiera jugar en red..*
- *Como usuario, me gustaría que se pudiera jugar contra una inteligencia artificial, así como que ellas jugaran solas.*

SPRINT 5

Revisión del Sprint

En primer lugar, la funcionalidad de *"Como usuario, me gustaría que se pudiera jugar en red."* ha sido completada con éxito para el modo GameClassic a falta de determinar si debería implementarse para GameTeams.

La funcionalidad de *"Como usuario, me gustaría que se pudiera jugar contra una inteligencia artificial, así como que ellas jugaran solas."* ha sido completada en gran medida en su fase Beta. No obstante, precisa refactorización y por ello hemos optado por no incluirla en el código principal en este sprint, ya que dificultaba la funcionalidad del juego en red.

Retrospectiva

More Of

- Mantener a todos los miembros al tanto de los cambios, modificaciones o direcciones que toma el proyecto a lo largo del proyecto.
- Es necesario poner al día los diagramas UML del proyecto.

Keep Doing

- Seguir manejando de forma planificada y eficiente el uso de git y gitHub, aprovechándonos de por ejemplo, la posibilidad de crear ramas.
- Trabajar de forma paralela y asignar una tarea (no siempre la misma para tener una visión global del proyecto) a grupos de entre los miembros del equipo ha agilizado la producción de software.
- Mantener el grado de comunicación entre los miembros del equipo y la periodicidad de la misma.
- Continuar con las reuniones dos días a la semana en los que reservamos un aula de trabajo en grupo para realizar avances del proyecto de forma paralela a los días de clase.
- Seguir asignando suficiente tiempo para la depuración del código.
- Fijar con antelación una reunión para llevar a cabo el Sprint Review.
- Seguir haciendo reuniones de debate sobre cuál es el mejor diseño antes de empezar a programar.
- Seguir haciendo la documentación con un tiempo prudencial antes del final del sprint.

Start Doing

Stop Doing

Less of

- Deberíamos dedicarle algo menos de tiempo al código en favor de la documentación.

Planificación del siguiente Sprint

Durante el siguiente Sprint, el equipo de desarrollo concluirá las siguientes tareas de la historia de usuario "*Como usuario quiero que Rolit introduzca características innovadoras pensando en las posibilidades que brinda el multijugador*" del Product Backlog:

- "*Como usuario, me gustaría que se pudiera jugar contra una inteligencia artificial, así como que ellas jugaran solas.*"

Además también realizaremos las tareas de la historia de usuario "*Como usuario, me gustaría que Rolit fuera intuitivo y cómodo de jugar*" del Product Backlog:

- "*Como usuario, me gustaría que hubiese un tutorial para entender bien las normas y excepciones del juego.*"

De cara al código deberíamos terminar todos los TODO, FIXME e "issues", así como la refactorización del mismo y el tratamiento de las excepciones.

Deberíamos continuar actualizando y mejorando la documentación. Siendo de vital importancia los UML de sprints anteriores y del actual.

SPRINT 6

Revisión del Sprint

La funcionalidad de "*Como usuario, me gustaría que se pudiera jugar contra una inteligencia artificial, así como que ellas jugaran solas*" ha sido completada con éxito. Existen 3 niveles de dificultad seleccionables para IA que pueden jugar entre ellas o contra otros jugadores físicos.

La característica "*Como usuario, me gustaría que hubiese un tutorial para entender bien las normas y excepciones del juego*" ha sido pospuesta por retrasos en la depuración y funcionamiento correcto del juego general, lo que imposibilitaba que se pudiese hacer una simulación a modo de tutorial.

La Historia de Usuario "*Como usuario, me gustaría poder jugar a Rolit con una interfaz agradable*" ha sido por fin completada en su totalidad al adaptar la consola para seguir jugando a Rolit, además se ha hecho un lavado de cara a la GUI que hace más intuitivo y agradable el juego.

La funcionalidad "*Como usuario, me gustaría que se pudiera jugar en red*" ha sido ampliada para que se puedan jugar los distintos modos de juego también en red. Además, se han introducido nuevas ventanas en la vista para aumentar el feedback que recibe el usuario cuando se conecta a un servidor, mejorando también la Historia de Usuario "*Como usuario, me gustaría poder jugar a Rolit con una interfaz agradable*".

Retrospectiva

More Of

- Es necesario poner al día los diagramas UML del proyecto.
- Es necesario poner al día los documentos de desarrollo que se han venido haciendo.
- Deberíamos ir dejando cerrados los *issues* marcados en GitHub.

Keep Doing

- Trabajar de forma paralela y asignar una tarea (no siempre la misma, para tener una visión global del proyecto) a grupos de entre los miembros del equipo ha agilizado la producción de software.
- Mantener a todos los miembros al tanto de los cambios, modificaciones o direcciones que toma el proyecto a lo largo del proyecto.
- Mantener el grado de comunicación entre los miembros del equipo y la periodicidad de la misma.

- Continuar con las reuniones dos días a la semana en los que reservamos un aula de trabajo en grupo para realizar avances del proyecto de forma paralela a los días de clase.
- Seguir asignando suficiente tiempo para la depuración del código.
- Fijar con antelación una reunión para llevar a cabo el Sprint Review.
- Seguir haciendo reuniones de debate sobre cuál es el mejor diseño antes de empezar a programar.

Start Doing

- Hacer las planificaciones de los sprint un poco más exhaustivas, dejando bien marcadas las responsabilidades y las necesidades a priori de cada Historia de Usuario.

Stop Doing

Less of

- Deberíamos dedicarle algo menos de tiempo al código en favor de la documentación.

Planificación del siguiente Sprint

Durante el siguiente Sprint, el equipo de desarrollo llevará a cabo las siguientes tareas de la historia de usuario "*Como usuario quiero que Rolit introduzca características innovadoras siendo intuitivo y cómodo de jugar*":

- *Como usuario, me gustaría que hubiese un tutorial para entender bien las normas y excepciones del juego.*
- Nos vamos a dedicar a solucionar posibles bugs y poner tensión a las situaciones de juego para comprobar que las soluciones son consistentes.

DESCRIPCIÓN DE TRABAJO

Como trabajo conjunto, es importante destacar la discusión sobre el diseño previa al desarrollo del código en la que todos los integrantes se han puesto de acuerdo en el reparto de tareas, especificación de resposabilidades, acotación de conceptos... Se ha dedicado mucho tiempo a esto y, aunque no se ve a simple vista en el código, claramente se ve reflejado en el resultado final y en la capacidad de cambio del proyecto.

Sergio Miguel García Jiménez

Código

- Crear parte del menú principal en modo consola.
- Colaborar en el diseño de la clase Player.
- Desarrollar gran parte de la lógica de cargar, guardar y borrar partida (SaveLoadManager).
- Contribuir a que modelo trabaje exclusivamente con JSON. Esto incluía la concepción del diseño de reports así como su implementación en las distintas clases, tarea inicialmente desarrollada por Leo, Virginia y yo, y continuada por el resto de compañeros.
- Desarrollé junto con Leo toda la funcionalidad de red, incluyendo elementos de la GUI que precisa, su extensión a todos los modos de juego y la lógica detrás de las conexiones.
- Me encargué junto a otros del desarrollo de la estructura de la GUI; concretamente en lo referente al tablero, diálogos de cargar, crear, borrar partida y la interacción de la GUI con el modelo, así como una participación activa en su pulido y debug por las posteriores características introducidas por el modelo.
- Realicé diversas labores de optimización para solventar un pobre rendimiento de la GUI a la hora de mostrar el tablero, actualizándose ahora en tiempo casi instantáneo y constante.

- El debug de las características que estaba desarrollando conllevó a debugear otras clases referentes al modelo para perfeccionarlo.
- Encargado de realizar algunos merge y etiquetado.
- Proponer y solventar algunas de las issues.

Documentación

- He realizado los primeros diagramas de clases y UML para ofrecer unos primeros prototipos para poder determinar por qué estilo de diseño queríamos adoptar.
- Concebí el diseño de la refactorización de las historias de usuario para adaptarlas al formato de la asignatura, tarea después completada por compañeros.
- Creación de los primeros artículos y la estructura preliminar de la Wiki en los primeros sprints.
- Trabajo UML preliminar con Modelio.
- Pugnar por pasar de trabajar en el software Modelio a PlantUML para trabajar con mayor comodidad.

Leonardo Macías Sánchez

Código

- Concepción, diseño y desarrollo de la clase SaveLoadManager, que en sprints futuros sería refactorizada por Sergio para adaptarla a las nuevas necesidades.
- Desarrollo de las clases Player, Cube y Color, sobre todo durante la etapa inicial del proyecto.
- Refactorización de la clase Board para que contuviera la lógica del juego.
- Desarrollo y adaptación del juego a la red, incluyendo una extensión de la interfaz gráfica que facilitase el funcionamiento al usuario. Esta tarea fue desarrollada conjuntamente con Sergio.
- Diseño y desarrollo del paquete replay, que abarca la interfaz Replayable y las clases Replay y GameState. Además de implementar toda la lógica, también me he dedicado a su correcta visualización, tanto en consola como en GUI.
- Principal impulsor y defensor del uso de JSONObjects en el proyecto. Ideé la interfaz Reportable y me encargué del diseño de los JSON y de su posterior implementación junto con Sergio y Virginia.
- Diseño e implementación de los métodos toString() del modelo.
- Colaboración en el diseño de los GameBuilders junto con Juan Diego y Daniel.
- Refactorización, junto a Juan Diego, de la clase Game para poder introducir las modalidades de GameTeams y GameClassic.
- Ayudar a Virginia y Mar con problemas que surgieron al crear JUnits.
- Creación del ranking junto con Mar.
- Refactorización completa de todas las clases de la GUI, rediseñandola al completo (a excepción del tablero) y creando nuestros propios componentes visuales, los RolitComponents.
- Propuse la creación de la clase TurnManager.
- Labores de debug.

Documentación

- Diseño de los reports.
- Redactar el funcionamiento de los Builders.
- Desarrollo de diagramas de secuencia.
- Redactar algunos de los sprint reviews y retrospectives.

- Colaboración en el desarrollo de la wiki.
- Búsqueda de una alternativa a Modelio para el desarrollo de diagramas de clases.

Daniel González Arbelo

Código

- Desarrollo en las etapas más tempranas del proyecto de las clases Game y Board, implementando los métodos encargados de la ejecución y actualización del juego en los turnos.
- Retoques y debug en las primeras etapas de la clase SaveLoadManager para el funcionamiento de cargado y guardado de partida.
- Creación de las clases herederas de Command, junto con Mar, aplicándose el patrón comando.
- Desarrollo de la lectura de datos de entrada para la creación del juego por consola junto con Leo y Mar.
- Creación de los ficheros para las formas de los tableros y las imágenes de las celdas en GUI.
- Creación de las primeras versiones funcionales de la GUI (ventanas de creación de juego, tablero, ranking, etc.), desarrolladas todas conjuntamente con Sergio.
- Creación, desarrollo y debug de las inteligencias artificiales y sus estrategias (y desarrollo de algunas alternativas para su funcionamiento con Juan Diego, que fueron finalmente deprecadas).
- Resolución de algún fallo en el funcionamiento de la red.
- Refactorización, junto con Juan Diego, del modelo para que trabajase en su propia hebra, y de la vista por consola.
- Refactorización del manejo de turnos (con la creación de TurnManager, idea elaborada con Leo) y de la clase Player para adaptarlos al funcionamiento con la hebra del modelo y las inteligencias artificiales.
- Creación de la lógica encargada del funcionamiento del tutorial.
- Tareas de debug, manejo de excepciones y resolución de issues.

Documentación

- Creación de los primeros diagramas de secuencia para fijar el formato.
- Establecimiento de un procedimiento para el desarrollo de los diagramas de secuencia usando PlantUML y IntelliJ.
- Documentación acerca del funcionamiento de la GUI.
- Participación en los sprint reviews y retrospectives.

María del Mar Ramiro Ortega

Código

- Desarrollo de un primer modelo de ranking, mostrado al final de la partida y ordenado por puntos.
- Creación de las clases herederas de Command, junto con Daniel, aplicándose el patrón comando.
- Diseño, junto con Daniel, y desarrollo de los distintos tamaños y formas del tablero, tanto a nivel de lógica del juego como a nivel de su carga y guardado por medio de la clase SaveLoadManager.
- Realización de una versión incial de los JUnits con mi compañera Virginia.
- Actualización de los JUnits en todas las refactorizaciones siguientes junto con Sergio.
- Añadido de nuevos tests a los JUnits de las distintas historias de usuario que íbamos implementando.
- Creación del ranking de GUI junto con Leo y mejora del ranking de console.

Documentación

- Creación de diagramas de secuencia.
- Desarrollo de una refactorización de las Historias de Usuario junto con Virginia, que fueron actualizadas y subidas a la wiki.
- Documento acerca de los diversos cambios que hicimos durante los distintos sprints junto con Leo, así como su actualización en la wiki.
- Redactar algún sprint review y retrospective, así como la actualización de la wiki respecto a varios sprints reviews y retrospectives que realizamos en PDF.
- Participación en los sprint reviews y retrospectives, junto con el resto de mis compañeros.
- Javadoc del paquete Builder y de la vista de consola.
- Documentación y UML sobre la HU *Como usuario quiero poder jugar a Rolit siguiendo un conjunto mínimo de normas* .

Juan Diego Barrado Daganzo

Código

- Desarrollo de la lógica interna de Game en las fases iniciales del proyecto, creación de la clase Player e integración de la misma sobre el modelo.
- Desarrollo del mecanismo de cambio de puntuaciones a través de los Cube, que permitía encapsular mucho mejor la puntuación de cada jugador.
- Diseño de las responsabilidades conceptuales de los integrantes del modelo, es decir, asignar las capacidades y dependencias del Board, Game, Player... tanto a nivel de código como a nivel de concepto, lo que unificó lo que se esperaba de cada clase e hizo más fácil la integración de funcionalidades posteriores.
- Creación y diseño de la clase builder y de la refactorización del juego a través de factorías.
- Encapsulación del modelo en la clase abstracta de Game, definición de las responsabilidades de cada una de las clases hijas (GameClassic y GameTeams) e implementación de las mismas.
- Adaptación del Modelo-Vista-Controlador a través de la generación de un hilo propio para el modelo que permitía una mejor integración de los jugadores automáticos. Creación de la cola de prioridad de Cube para poder utilizar el modo red y los jugadores automáticos de forma simultánea.
- Creación de la vista de consola y generación de la interfaz *ConsoleWindow* para mantener un criterio uniforme a modo de "componentes" tal y como se hace en Swing, lo que simplificó la independencia del modelo y la vista por su similitud con las ventanas de Swing.
- Depurado de la interfaz de RolitObserver y de los métodos de notificación a observadores.
- Depurado de la red para integrar los jugadores automáticos.

Documentación

- Rediseñado de los reports iniciales.
- Creación del formato de los documentos de desarrollo (los utilizados durante los sprint por el equipo de desarrollo, no los que se incluyen en la entrega) para informar a todos los participantes de la implementación final que se dió al diseño planeado de forma general.

Virginia Chacón Pérez

Código

- Diseño, desarrollo e implementación de los JSONObjects y de la interfaz Reportable junto con Leo y Sergio.
- Diseño y desarrollo de los tests de JUnit junto con Mar.
- Ayuda con la refactorización de la GUI a Leo.

Documentación

- Primera en empezar a entender y a trabajar con Modelio, aunque luego descubriésemos otros programas más útiles para nuestro proyecto en concreto.
- Diseño de los primeros diagramas de clases con Modelio.
- Diseño de algunos diagramas de secuencia, tanto en IntelliJ como en PlantUML.
- Creación de la parte de Historias de Usuario en la Wiki y actualización de la misma con Mar.
- Colaboración en otras partes de la Wiki.
- Participación en los Sprint Reviews y Retrospectives junto con el resto de mis compañeros.
- Creación y edición del vídeo tutorial para entender las normas del juego.