P.F. ENTERTAINMENT

Carlos Castaño Moreno	carlos051247@uma.es
Nuria Rodríguez Tortosa	nurirt36@uma.es
Daniel García Rodríguez	dani.gr@uma.es
Javier Leiva Dueñas	javi__@uma.es
Javier Lanceta Salas	jlancetasalas@uma.es
Guillermo Tell González	guillermotellg@uma.es
Julián Castro Coloma	julianc2b@uma.es
Pablo Fernández Serrano	pablofs02@uma.es

Enlace a repositorio: GitHub
Enlace a página web: Trivial

ÍNDICE

1. INTRODUCCIÓN

Con nuestra aplicación lo que buscamos es entretener y permitir que los usuarios puedan jugar en grupo a un juego tradicional, de manera online y gratuita mientras aprenden.

Nuestro primer proyecto será una adaptación de Trivial con nuevos objetivos, e ideas como partidas rápidas. Iremos subiendo actualizaciones en nuevos parches.

2. ROLES

Para la asignación de roles, se ha procurado que todos los puestos del proyecto estén cubiertos por más de una persona, para minimizar el impacto que puede tener el abandono de un miembro, de tal manera que siempre haya alguien responsable de cada parte.

Tras una reunión de todo el equipo, y mediante previo consenso, se han asignado todos los roles, teniendo en cuenta las fortalezas y debilidades de cada miembro del equipo.

Planificación	Diseño	Implementación	Pruebas	Scrum Master	Analista
Tell, G.	Castaño, C.	Leiva, J.	Tell, G.	Castaño, C.	Tell, G.
Castaño, C.	Fernández, P.	Lanceta, J.	Leiva, J.	Castro, J.	Rodríguez, N.
Rodríguez, N.	Rodríguez, N.	Fernández, P.	Lanceta, J.		
	Castro, J.	García, D.	Castro, J.		
			García, D.		

Product Owner: D. Javier Troya Castilla.

3. GESTIÓN DEL RIESGO

En este apartado, realizamos un análisis de todos los posibles peligros a los que nos enfrentamos emprendiendo este proyecto. Comentaremos con detalle cada uno de ellos, así como la probabilidad de que ocurran y la estrategia empleada para resolverlos, y que no supongan un gran problema en cuanto a la planificación de desarrollo.

1. Fuga de personal (Tipo: Personal, Efectos: Tolerables): debido a baja por enfermedad, o la decisión de algún miembro del equipo que no sigue interesado en continuar. Es algo poco probable de que ocurra de forma voluntaria, ya que el equipo presenta gran interés en el desarrollo y finalización de esta aplicación.

   Estimamos que, si falta una persona, quedan suficientes desarrolladores para seguir según lo previsto, por lo que su impacto en el desarrollo sería bajo, y una manera de solucionarlo es buscar a personal que esté interesado y cualificado por este tipo de proyectos.

2. Complejidad del producto a desarrollar (Tipo: Tecnológico, Efectos: Serios): causado en ocasiones por la falta de "realismo" por parte de todo el equipo, al imaginar un producto que no es capaz de cumplir con la cronología exigida por parte del cliente, o incapacidad de desarrollo. Desde el comienzo de este proyecto estamos procurando en todo momento, que el producto se ciña a nuestras posibilidades tanto a nivel de desarrollo, como del tiempo disponible. Por lo tanto, hay que buscar un equilibrio entre cubrir las necesidades inicialmente planteadas, como la capacidad de desarrollo. En el caso de que se diera tendría un impacto elevado sobre el producto final.

3. Fallo de la base de datos (Tipo: Tecnológico, Efectos: Serios): dado que dependemos de un repertorio de preguntas, que sin duda alguna constituye el núcleo del proyecto, y sin la que se podría llevar a cabo el correcto funcionamiento de la aplicación. Es algo poco probable de que ocurra, debido a que tenemos pensado realizar copias de seguridad de estos datos, para así tenerlos siempre disponibles, o al menos una parte importante que garantice el correcto funcionamiento mientras se repara la conexión con la misma, así que su efecto sería bajo.

4. Falta de motivación (Tipo: Personal, Efectos: Catastróficos): que puede ser derivada de la propia complejidad que conlleva el desarrollo, o bien porque alguien crea que no se cumplen los objetivos que tenía pensados. Esta situación es de las menos probables que ocurran, pero una forma de evitar que ocurra es intentar que el producto final sea consensuado por todos los integrantes del proyecto. Al igual que la fuga de personal supondría un impacto bajo en el trabajo.

5. Cambio en el planteamiento inicial del proyecto (Tipo: Organización, Efectos: Tolerable/Catastrófico): que puede ser motivado por no haber entendido exactamente los requerimientos del cliente, por falta de personal, falta de recursos temporales, o de cualquier otra incidencia. Estimamos que puede presentar un gran impacto en el normal desarrollo del proyecto, en la medida de cómo se modifica todo lo inicialmente planteado. En esta situación lo mejor es la comunicación con el cliente, para poder encontrar una solución, así como la reunión de todo el equipo, para tomar una decisión de manera conjunta.

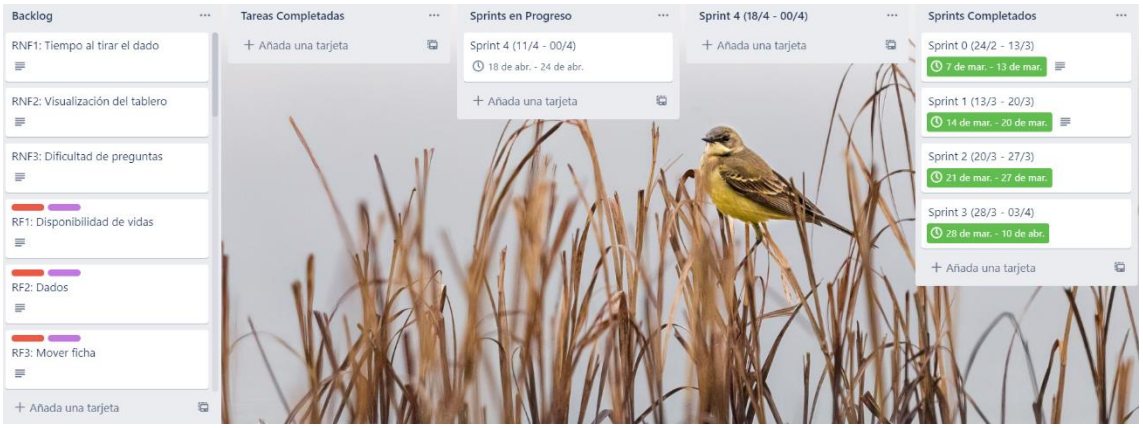
6. Pérdida de parte del proyecto (Tipo: Tecnológico, Efectos: Serios): debido a no almacenar debidamente alguna sección de código, o alguna parte de la memoria. Creemos que no es probable que ocurra, porque procuramos tener todas las copias de seguridad necesarias, y notificar los pequeños cambios que se vayan produciendo, en el caso de que se produzca una pérdida de código importante tendría un elevado impacto en el proyecto, ya que aumentaría el tiempo necesario para su finalización.

   Puede deberse a múltiples situaciones, y diversas partes del proyecto pueden verse involucradas, por lo que en cada caso habría que ver como asegurar que tenga el menor impacto posible, y conlleva una gran prevención, por parte de todo el equipo, para asegurarnos de que todo en lo que se vaya avanzando, lo tengamos correctamente guardado.

4. PLANIFICACIÓN

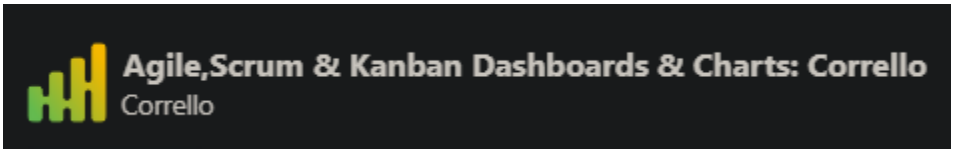
Para la organización de las subtareas del proyecto decidimos emplear la metodología Scrum, ya que nos permite tener más flexibilidad a los posibles cambios y adaptarnos a los contratiempos en el desarrollo del proyecto, además de permitir a todos los miembros del equipo a tomar parte de todas las partes del proyecto.

Por otro lado, gracias a su metodología incremental e iterativa, vamos comprobando la corrección de las partes anteriores al *sprint* actual, a través de reuniones semanales con todo el equipo.



Captura de las listas de tareas en Trello

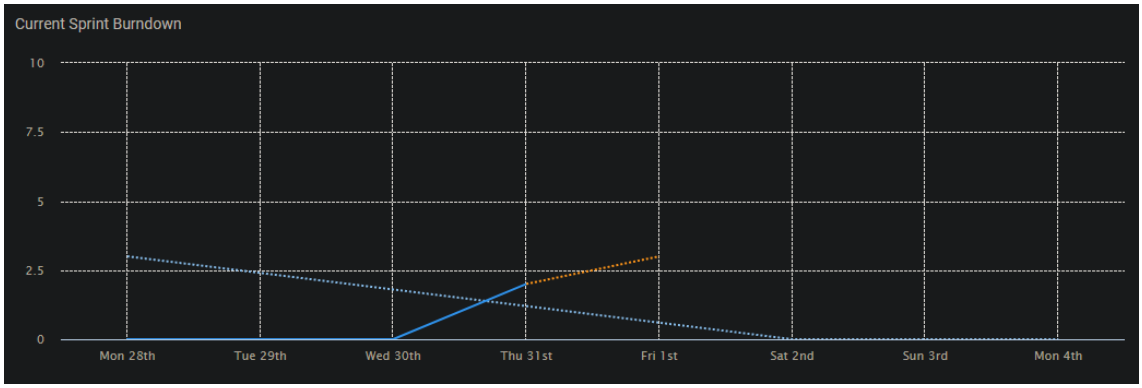
4.1 POWER-UPS



- **Corrello** para obtener gráficas Burndown del sprint actual.
- Automatización QoL con **Butler**:
    - Botón “Comenzar tarea”, que añade al usuario a la tarea y la marca como “En progreso”.
    - Botón “Terminar tarea”, que marca como cumplidas todas sus subtareas, y elimina la etiqueta “En progreso”.
    - Botón global “Mover completadas”, que mueve todas las tarjetas completadas a una lista auxiliar “Completed tasks”.
    - Botón global “Completar sprint”, que mueve de vuelta las tarjetas en “Completed tasks” a su sprint correspondiente.

4.2 GRÁFICA BURNDOWN

A través del Power-Up Corrello, se obtuvo la siguiente gráfica, no del todo exacta debido a los contratiempos de incorporación del Power-Up en el tablero Trello del proyecto.



5. REQUISITOS

En este apartado, presentamos los distintos requisitos surgidos de una entrevista con el cliente, así como una posterior reunión de todo el equipo. Para ello, tras la obtención de estos, los hemos especificado, analizado, una gestión de estos, para finalmente verificarlos y validarlos (Ingeniería de requisitos). Mostramos los requisitos explicados y el diagrama UML generado con el programa Magic Draw.

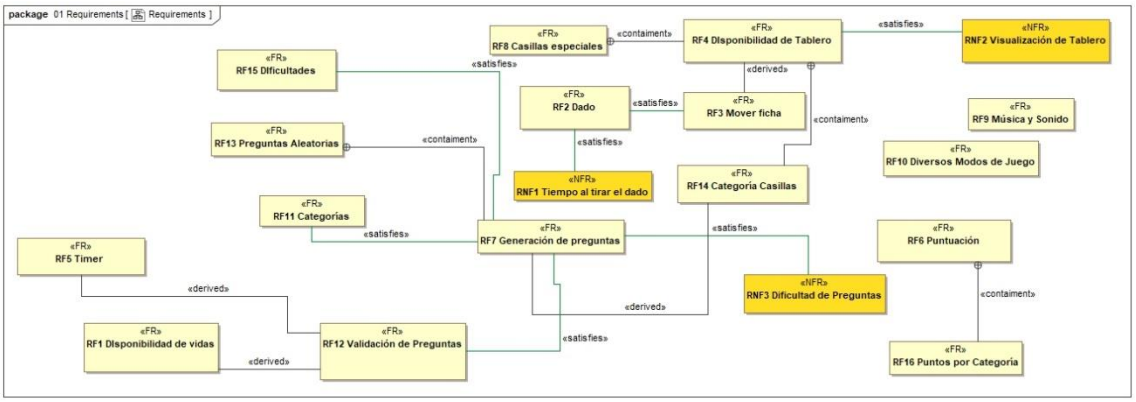


Diagrama de Requisitos

TARJETAS DE REQUISITOS

RF1: Disponibilidad de vidas

Como cliente,

Quiero tener disponibilidad de vidas.

Para tener un sistema que nos permita calcular el número de errores que comete el usuario y poder terminar la partida antes de llegar a la victoria si se excede el máximo de ellas.

Prueba de aceptación

- Al principio el usuario empezará con un número determinado de vidas.
- En el caso de que el usuario conteste una pregunta de maner errónea, o se quede sin tiempo se le restara una vida.
- El usuario tendrá la oportunidad de ganar una vida si cae en una de las casillas especiales del tablero.
- Si el usuario se queda sin vidas el juego terminará y habrá perdido.

RF2: Dados

Como usuario/jugador,

Quiero poder tirar dados.

Para poder avanzar sobre el tablero.

RF3: Mover ficha

Como sistema,

Quiero que se pueda mover la ficha del jugador cuando tire el dado.

Para que se garantice el correcto funcionamiento del juego, y además que la partida no entre en estado de bloqueo.

RF4: Disponibilidad de tablero

Como cliente,

Quiero tener siempre disponible al menos un tablero previamente generado.

Para que se garantice el que pueda siempre jugar a una partida, aun habiendo fallo en la generación de este.

Prueba de aceptación

- Al iniciar una nueva partida, se procurará generar un tablero aleatorio preferiblemente, con una distribución uniforme de casillas de categorías de preguntas.
- El tablero generado tendrá una forma lineal, con casillas de categorías de preguntas, y otras casillas especiales que pueden suponer por ejemplo la pérdida de vidas.
- En caso de error, existirá un tablero de repuesto, para que siempre se garantice su disponibilidad.

RF5: Timer

Como cliente,

Quiero que el juego cuente con un temporizador.

Para limitar cuánto tiempo el jugador tiene para responder la pregunta.

RF6: Puntuación

Como cliente,

Quiero que el juego tenga un sistema de puntuación cuando respondes preguntas.

Para que los jugadores puedan comparar puntuaciones entre ellos.

RF7: Generación de preguntas

Como sistema,

Quiero tener disponibles las preguntas en una base de datos.

Para poder generar preguntas de diferentes categorías y dificultades.

Prueba de aceptación

- Se recibe una petición de pregunta al caer en una casilla.
- Se accede al fichero que almacene las preguntas de la categoría escogida
- Se filtran las preguntas según la dificultad escogida.
- Se elige una pregunta aleatoria de entre las filtradas.
- Nos aseguramos de que no pueda volver a ser escogida la misma pregunta.

RF8: Casillas especiales
Como cliente, **Quiero** que el tablero disponga de casillas especiales que modifiquen el juego. **Para** mejorar la jugabilidad.
RF9: Música y Sonido
Como usuario, **Quiero** que se reproduzca música de fondo y ciertos sonidos como consecuencia a algunas acciones. **Para** mejorar la experiencia de manera indirecta.
RF10: Diversos Modos de Juego
Como usuario, **Quiero** se pueda seleccionar al inicio de la partida qué modo se quiere jugar. **Para** añadir más diversidad con la misma base.
RF11: Categorías
Como usuario, **Quiero** tener preguntas de distintas categorías. **Para** que el juego tenga más variedad.
RF12 Validación de preguntas
Como sistema, **Quiero** comprobar que la respuesta sea la correcta. **Para** contar correctamente el progreso de cada jugador.

Prueba de aceptación

- Al responder una pregunta, comprobar que su respuesta es la correcta.
- Comprobar que el usuario ha respondido la pregunta dentro del tiempo establecido.
- Sumar puntos de esa categoría a la estadística del usuario si ha acertado la pregunta.
- Indicar que el usuario ha perdido una vida si falla la pregunta.
- Indicar que el usuario ha perdido una vida si no ha contestado la pregunta antes de que el timer llegue a 0.

RF13: Preguntas Aleatorias

Como cliente,

Quiero que las preguntas se generen de forma aleatoria.

Para que no puedan salir preguntas repetidas.

RF14: Categoría Casillas

Como cliente,

Quiero que las casillas del tablero se dividan en las distintas categorías.

Para tener las preguntas divididas según su categoría.

RF15: Dificultades

Como cliente,

Quiero que haya distintos niveles de dificultad.

Para mejorar la jugabilidad según la habilidad del jugador.

RF16: Puntos por categoría

Como usuario,

Quiero obtener distintos puntos según la categoría de la pregunta.

Para medir mi conocimiento en las distintas categorías.

RNF1: Tiempo al tirar el dado
Como usuario, **Quiero** que no tarde mucho la animación y el funcionamiento del dado. **Para** agilizar el funcionamiento del juego.
RNF2: Visualización de tablero
Como usuario, **Quiero** que el tablero y la interfaz se vea correctamente y sea agradable a la vista.
RNF3: Dificultad de preguntas
Como usuario, **Quiero** que la dificultad de las preguntas mostradas sean acordes a la dificultad elegida. **Para** que la dificultad del juego sea coherente.

6. CASOS DE USO

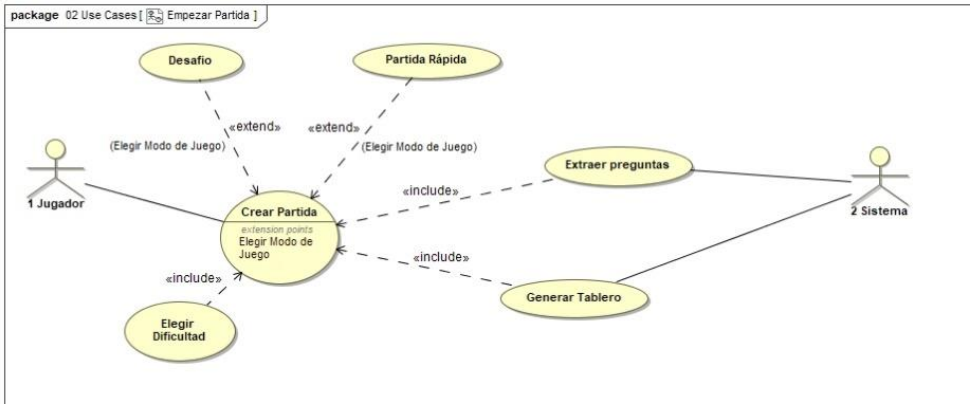
CU1. Crear partida

Contexto de uso: cuando el jugador lo requiera, podrá empezar una nueva partida en el modo de juego seleccionado
Precondiciones y activación: Cuando el jugador esté conectado al juego en el menú principal.
Garantías de éxito / Postcondición: se crea una nueva partida.
Escenario principal:

1. El jugador selecciona una nueva partida.
2. El jugador selecciona el modo de juego y la dificultad.
3. El sistema genera un nuevo tablero.
4. Se muestra al jugador la interfaz del juego con el tablero generado.

Escenarios alternativos:

- El sistema no consigue generar el tablero y se mostrará un error por pantalla.



Crear partida

CU2. Jugar

Contexto de uso: cuando la partida es creada, el jugador podrá jugar en ella.

Precondiciones y activación: el jugador ha seleccionado un modo de juego y el sistema ha creado la partida

Garantías de éxito / Postcondición: el jugador podrá jugar la partida seleccionada.

Escenario principal:

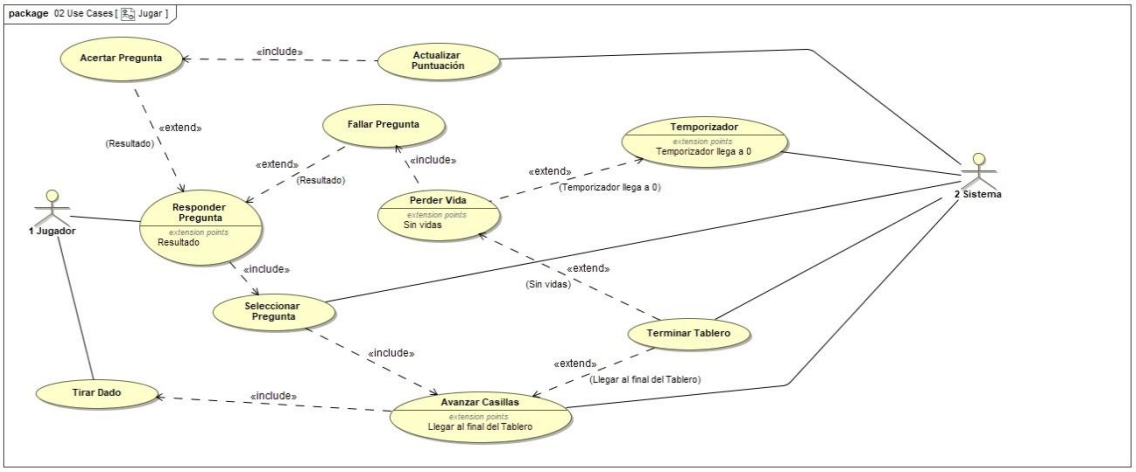
1. El jugador tirará un dado
2. El sistema moverá al jugador por el tablero en proporción al número que ha sacado el jugador en el dado
3. El sistema seleccionará una pregunta y se la mostrará al jugador.
4. El sistema activa el temporizador.
5. El jugador debe responder a esa pregunta antes de que el tiempo expire.
6. Si el jugador acierta, el sistema deberá actualizar la puntuación.
7. Si el jugador falla, el sistema le quitará una vida.
8. Si el jugador llega a la meta o el jugador se ha quedado sin vidas, el sistema termina el tablero.

Escenario alternativo:

- No se puede tirar el dado.
- El jugador no visualiza la pregunta correctamente.
- El sistema no reduce el número de vidas cuando el jugador falle una

pregunta o no responda antes.

- El sistema no reduce el número de vidas cuando el jugador falle una pregunta o no responda antes de que el tiempo se agote.
- El temporizador no funciona correctamente.
- El sistema no termina el tablero.



Jugar

CU3. Terminar partida

Contexto de uso: el jugador ha acabado la partida.

Precondiciones y activación: para llegar a este estado el jugador debe haber completado el tablero o haber perdido todas las vidas.

Garantías de éxito / Postcondición: se mostrará su puntuación y podrá volver a jugar.

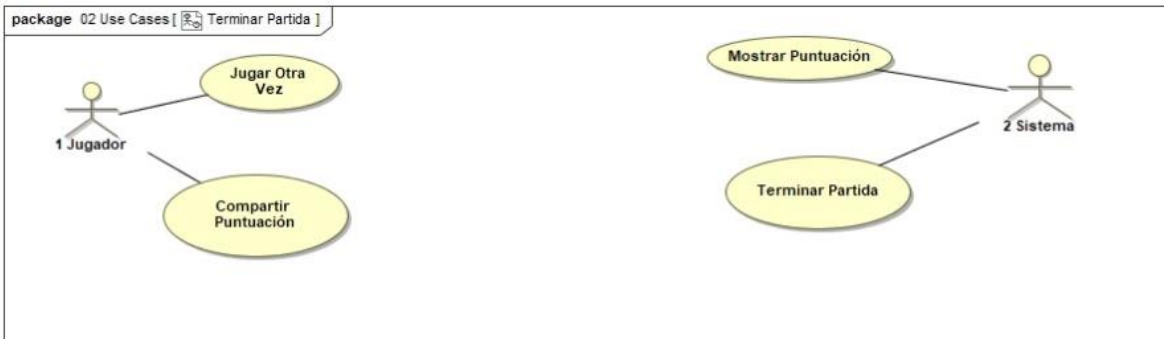
Escenario principal:

1. El sistema finaliza la partida.
2. El sistema muestra la puntuación obtenida al jugador.
3. El jugador puede compartir sus resultados en redes sociales.
4. El jugador puede volver a jugar otra partida.

Escenario alternativo:

- El sistema no finaliza la partida.
- El sistema no muestra la puntuación obtenida al jugador.

- El jugador no puede compartir su derrota en redes sociales.
- El jugador no puede volver a intentarlo.



Terminar partida

7. HERRAMIENTAS

Para la gestión y desarrollo del proyecto, así como para la constante comunicación de todo el equipo, destacamos el uso de las siguientes aplicaciones:

1. **Microsoft Word**: Redacción del presente documento (memoria del proyecto).
2. **Discord**: Comunicación y organización de las sesiones síncronas del proyecto.
3. **GitHub**: Almacenamiento y gestión del repositorio del proyecto online.
4. **Git**: Manejo del repositorio del proyecto de forma local.
5. **WhatsApp**: Comunicación entre los miembros.
6. **Trello**: Organización de las partes del proyecto.
7. **Magic Draw:** realización de diagramas.