



**UNIVERSIDAD DE CASTILLA-LA MANCHA**  
**ESCUELA SUPERIOR DE INFORMÁTICA**

**GRADO EN INGENIERÍA INFORMÁTICA**  
**TECNOLOGÍA ESPECÍFICA DE COMPUTACIÓN**

**TRABAJO FIN DE GRADO**

GLOBAL-MANAGER: Entrenando mediante un Juego Serio a  
los Jefes de Proyecto en los Desafíos del Desarrollo Global del  
Software

Rubén Márquez Villalta

julio, 2020





**UNIVERSIDAD DE CASTILLA-LA MANCHA**  
**ESCUELA SUPERIOR DE INFORMÁTICA**

**Departamento de Tecnologías y Sistemas de la Información**

**TECNOLOGÍA ESPECÍFICA DE COMPUTACIÓN**

**TRABAJO FIN DE GRADO**

**GLOBAL-MANAGER: Entrenando mediante un Juego  
Serio a los Jefes de Proyecto en los Desafíos del  
Desarrollo Global del Software**

Autor: Rubén Márquez Villalta

Tutor: Francisco Pascual Romero Chicharro

Co-Tutor: Aurora Vizcaíno Barceló

julio, 2020

TRIBUNAL:

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

FECHA DE DEFENSA: \_\_\_\_\_

CALIFICACIÓN: \_\_\_\_\_

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

*Dedicatoria...*



## **Resumen**

En la actualidad, cada vez más empresas están introduciendo un nuevo modelo de desarrollo, el cual resulta ser más des-localizado que el modelo convencional, donde los miembros del proyecto pueden estar en distintos países. Esta tendencia, llamada Desarrollo Global de Software (DGS), está creciendo rápidamente debido a la globalización, sin embargo, conlleva que aparezcan nuevos riesgos en su gestión, los cuales pueden ser agrupados en tres bloques: comunicación, coordinación y control. Es por ello, que se necesitan a jefes de proyecto preparados para afrontar y solventar los problemas que puedan ocurrir, lo que requiere de ciertas habilidades técnicas y no técnicas (soft skills) para gestionar con éxito este tipo de proyectos.

Últimamente, ha cobrado gran importancia el uso de juegos serios para la enseñanza y el entrenamiento de un conjunto de conocimientos y habilidades específicas. Por lo tanto, en este proyecto se llevará a cabo el desarrollo de un juego serio para el entrenamiento de jefes de proyecto en habilidades no técnicas necesarias para afrontar con éxito la gestión de un proyecto de software global. Este juego contará con un módulo de Inteligencia Artificial con el fin de monitorizar las acciones del jugador y ajustar dinámicamente el desarrollo del mismo.





## **Abstract**



# AGRADECIMIENTOS

---

*Rubén Márquez Villalta*  
Ciudad Real, 2020



# ÍNDICE GENERAL

---

<b>Resumen</b>	<b>VII</b>
<b>Agradecimientos</b>	<b>XI</b>
<b>Índice de figuras</b>	<b>XV</b>
<b>Índice de tablas</b>	<b>XVII</b>
<b>Índice de listados</b>	<b>XIX</b>
<b>Índice de algoritmos</b>	<b>XXI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Propuesta . . . . .	2
1.3. Estructura del documento . . . . .	4
<b>2. Objetivos</b>	<b>7</b>
2.1. Objetivo principal . . . . .	7
2.2. Objetivos específicos funcionales . . . . .	7
2.3. Objetivos específicos técnicos . . . . .	8
<b>3. Estado del arte</b>	<b>11</b>
3.1. Desarrollo Global del Software . . . . .	11
3.1.1. Beneficios del Desarrollo Global del Software . . . . .	11
3.1.2. Desafíos del Desarrollo Global del Software . . . . .	12
3.1.3. Rol del jefe de proyecto . . . . .	14
3.2. Habilidades necesarias en Desarrollo Global del Software . . . . .	14
3.2.1. Habilidades en el equipo de trabajo de Desarrollo Global del Software . . . . .	14
3.2.2. Habilidades en jefes de proyecto de Desarrollo Global del Software . . . . .	14
3.3. Gamificación . . . . .	14
3.3.1. Juegos Serios . . . . .	14
3.4. Trabajos relacionados con el tema . . . . .	14
3.4.1. Juegos Serios para Desarrollo Global del Software . . . . .	14
3.4.2. Juegos Serios para Jefes de Proyecto . . . . .	14
3.4.3. Juegos Serios para Jefes de Proyecto en Desarrollo Global del Software . . . . .	14
<b>4. Método de Trabajo</b>	<b>15</b>
4.1. Scrum . . . . .	15
4.1.1. Roles . . . . .	15
4.1.2. Componentes de Scrum . . . . .	15
4.2. Desarrollo basado en Prototipos . . . . .	15

4.2.1.	Etapas del modelo de prototipos . . . . .	15
4.3.	Marco Tecnológico . . . . .	15
4.3.1.	Herramientas Software . . . . .	15
4.3.2.	Herramientas Hardware . . . . .	15
<b>5.</b>	<b>Resultados</b>	<b>17</b>
5.1.	Sprint 0 . . . . .	17
5.1.1.	Equipo Scrum . . . . .	17
5.1.2.	Alcance del Proyecto . . . . .	17
5.1.3.	Pila del Producto . . . . .	17
5.1.4.	Planificación del Proyecto . . . . .	17
5.2.	Sprint 1 . . . . .	17
5.3.	Sprint 2 . . . . .	17
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>19</b>
6.1.	Conclusión . . . . .	19
6.2.	Lecciones Aprendidas . . . . .	19
6.3.	Trabajo Futuro . . . . .	19
6.4.	Publicaciones . . . . .	19
6.5.	Valoración Personal . . . . .	19
	<b>Bibliografía</b>	<b>21</b>

# ÍNDICE DE FIGURAS

---

1.1.	Colaboración mundial en el DGS . . . . .	1
1.2.	Desafíos en los proyectos DGS . . . . .	2





# ÍNDICE DE TABLAS

---

1.1.	Resumen del proyecto G3SOFT . . . . .	4
2.1.	Resumen de los objetivos del TFG . . . . .	9
3.1.	Resumen de las dificultades más importantes en proyectos DGS . . . . .	14



# ÍNDICE DE LISTADOS

---



# ÍNDICE DE ALGORITMOS

---



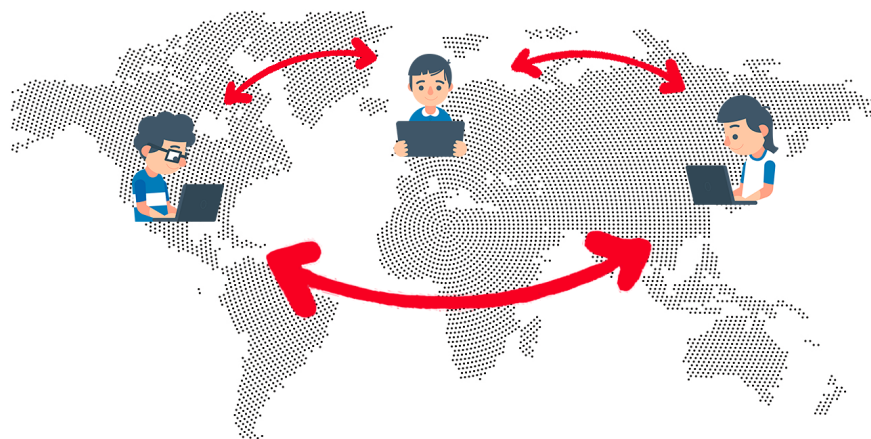
# INTRODUCCIÓN

---

## 1.1. MOTIVACIÓN

En los últimos años, la Ingeniería del Software ha observado notables cambios a la hora de desarrollar proyectos software. Tradicionalmente, el modelo de desarrollo de software utilizado consistía en la coordinación de diferentes equipos de trabajo en un mismo edificio (*Desarrollo Colocalizado*), posteriormente, estos equipos de trabajo pasaron a organizarse entre diferentes edificios de una o varias ciudades, pero siempre centralizados en un mismo país (*Desarrollo Distribuido*). Sin embargo, en la actualidad y debido a la globalización, cada vez más compañías separadas geográficamente colaboran para desarrollar software hasta traspasar fronteras llegando a un nivel mundial, por lo que se ha evolucionado hacia un modelo de desarrollo mucho más globalizado y deslocalizado, conocido como *Desarrollo Global de Software* (DGS), o en inglés *Global Software Development* (GSD) [17].

El DGS está teniendo cada vez más aceptación entre los profesionales. Esta tendencia consiste en la colaboración entre diferentes equipos de desarrollo (fig. 1.1), los cuales se encuentran ubicados alrededor del mundo en diferentes ciudades, países y continentes. Estos grupos de trabajo pueden pertenecer a distintas organizaciones, pero trabajan conjuntamente en un mismo proyecto software. En el proyecto podrá existir tanto una comunicación *asíncrona* como *síncrona* entre los equipos de trabajo, lo cual dependerá de una serie de características del proyecto [18].



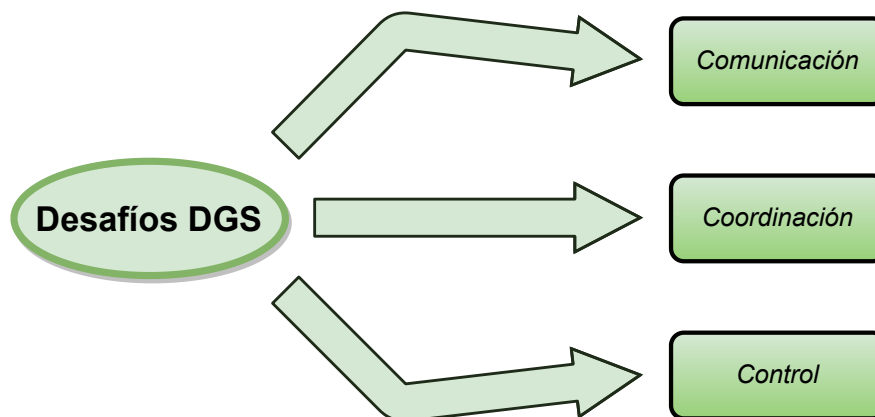
**Figura 1.1:** En el DGS diferentes equipos de desarrollo colaboran a nivel mundial en un mismo proyecto

Gradualmente, esta tendencia esta cogiendo cada vez más fuerza en el campo de la ingeniería del software, considerándose una norma en el desarrollo de software [5]. Esto es debido a que las

organizaciones pueden conseguir grandes beneficios utilizando este nuevo modelo de desarrollo, ya que la principal ganancia que se consigue con su uso es la reducción en el coste económico de los proyectos, debido a que se suelen buscar territorios donde la mano de obra cualificada es barata y fácilmente disponible [14]. Además, se pueden encontrar otros beneficios notables como pueden ser el acercamiento del desarrollo del software al cliente y al mercado local, la reducción del período necesario para el desarrollo del software al maximizar la productividad y la expansión hacia la inclusión de trabajadores mayormente cualificados en sus actividades de desarrollo [1].

Sin embargo, acompañando a las anteriores ventajas que se pueden conseguir con los proyectos DGS, existen una serie de inconvenientes, los cuales son causados, principalmente, a las diferencias existentes en este tipo de proyectos las cuales podemos dividir en cuatro clases: las *diferencias lingüísticas*, la *distancia geográfica*, la *diferencia cultural* y la coexistencia de diferentes *zonas horarias*; haciendo mucho más difícil el consenso y entendimiento común [14]. Estas diferencias acentúan la problemática de administrar y gestionar un proyecto software, apareciendo así los tres principales desafíos en la gestión de proyectos DGS (fig. 1.2), también llamado en [17] como las tres ces:

- Desafíos en la comunicación. Los equipos de trabajo deben mantener una comunicación adecuada y activa, con el fin de llevar a cabo un intercambio constante de información y conocimientos.
- Desafíos en la coordinación. Las tareas deben estar sincronizadas, para no sufrir retrasos y alcanzar objetivos e intereses comunes.
- Desafíos en el control. El proyecto debe ser gestionado constantemente y confirmar que se cumplen fechas de entrega, estándares, presupuestos, etc.; además de solventar posibles contratiempos que puedan ocurrir durante el ciclo de vida del proyecto.



**Figura 1.2:** Los tres principales desafíos en la gestión de proyectos DGS

Estos inconvenientes y desafíos complican la gestión de este tipo de proyectos, lo que puede implicar en retrasos de tareas o incluso en la cancelación del mismo, ya que según la literatura, la mayoría de proyectos DGS terminan fracasando. Según [12], la principal causa del elevado fracaso de estos proyectos es debido a la imperfecta y dificultosa gestión de los mismos. Es por esto, que para conseguir los beneficios que nos ofrece el DGS es necesario que los jefes de proyecto posean grandes conocimientos y experiencia en la gestión de estos proyectos, además de contar con una serie de habilidades (no solo técnicas, sino también no técnicas), para hacer frente a los posibles contratiempos que puedan ocurrir en el ciclo de vida del proyecto y conseguir la finalización exitosa del mismo.

## 1.2. PROPUESTA

Como se ha indicado en la sección anterior, existe una gran problemática con la nueva tendencia de desarrollar software mediante un entorno global, debido a que un elevado número de proyectos



que utilizan este tipo de modelo de desarrollo terminan fracasando, y son escasos aquellos que consiguen finalizar exitosamente, obteniéndose así los beneficios que se consiguen frente al modelo de desarrollo tradicional. Esta situación se debe, en especial, a que la educación en actividades para enseñar conocimientos sobre DGS no se está teniendo en cuenta, lo que implica que futuros ingenieros de software no posean ciertas habilidades y capacidades necesarias para afrontar los desafíos que conllevan los proyectos DGS [14]. Es evidente que este modelo de desarrollo se termine convirtiendo en un estándar, por lo que es necesario entrenar a nuestros estudiantes de ingeniería de software para afrontar estas dificultades, ya que se terminarán convirtiendo en los futuros ingenieros de DGS [3].

La gestión y administración es el pilar principal sobre el que gira un proyecto, y en especial un proyecto DGS, ya que es necesaria la organización de un gran número de trabajadores y equipos de desarrollo, a los que se le añade la problemática de gestionar diferentes factores a tener en cuenta como la separación geográfica, la cultura de los diferentes países involucrados en el proyecto o el horario de trabajo en cada país, es por esto que gestionar este tipo de proyectos eficientemente, es un autentico reto. Por lo tanto, es evidente la necesidad de que existan jefes de proyecto altamente cualificados en la gestión de proyectos DGS, para que puedan afrontar la administración del mismo con éxito, solventando todos los impedimentos que puedan ocasionarse. Sin embargo, en la actualidad es complicado encontrar a jefes de proyectos DGS altamente cualificados, con una gran experiencia y con los conocimientos y habilidades necesarias para afrontar correctamente su trabajo. Como consecuencia, son muchas las organizaciones y artículos que han demandado la carencia de habilidades y experiencia en los jefes de proyecto DGS, como la principal causa del elevado nivel de fracaso en los mismos [12].

Por consiguiente, es notoria la necesidad de que existan programas de educación que enseñen a nuestro futuros ingenieros de software conocimientos sobre DGS en general, y habilidades (tanto técnicas como no técnicas) necesarias para afrontar con éxito la gestión de este tipo de proyectos en particular. En contraposición, llevar a cabo el entrenamiento y enseñanza de estas habilidades y conocimientos no es una tarea sencilla, puesto que se precisaría la necesidad de introducir a ingenieros de software inexpertos en proyectos reales (para que adquieran esa experiencia necesaria) y por consiguiente las compañías no estén dispuestas a invertir sus recursos en este tipo de programas de entrenamiento. Esta posición de las compañías es debido a que se pueden poner en riesgo proyectos en curso, además de resultar complejo reproducir un escenario real en un entorno de educación [14].

En cualquier caso, hay diferentes formas de llevar a cabo la educación de diferentes conocimientos prácticos y el entrenamiento de ciertas habilidades, sin que esto pueda afectar, en nuestro caso, a un proyecto real. En el campo de la *Educación en la Ingeniería de Software* se han realizado avances, buscando la manera más efectiva de educar ciertos conocimientos a estudiantes de ingeniería de software, apareciendo métodos tradicionales como proyectos finales, combinación de diferentes técnicas de enseñanza como aprendizaje basado en proyectos [2], innovadoras estrategias como las clases volteadas [8], o darle un enfoque relacionado con el uso de juegos, apareciendo el termino de *Gamificación* [11].

Dentro de la gamificación podemos encontrar diferentes tendencias como pueden ser: los cursos académicos [15], los entornos de aprendizaje [6] o las aplicaciones que presentan escenarios reales, conocidos como *Juegos Serios* (JSs) [13]. En especial, los JSs (también llamados juegos educativos), según [7], consisten en juegos que van más allá del puro entretenimiento y constituyen una potente herramienta que permite a sus jugadores experimentar y aprender de sus errores, adquiriendo así experiencia y conocimientos". Los JSs ayudarán en el proceso de aprendizaje mediante la simulación de entornos virtuales, sin el riesgo que conllevaría tener al estudiante en un entorno real [3, 7, 12], además de que la tendencia del desarrollo de JSs ha tenido una gran aceptación en la última década.

Como resultado de lo cual, este *Trabajo Fin de Carrera* (TFG) se centrará en el desarrollo de un JS, llamado *GLOBAL-MANAGER*. El objetivo de *GLOBAL-MANAGER* será el de ayudar a estudiante en ingeniería de software a adquirir y entrenar ciertas habilidades (en especial aquellas no técnicas

también llamadas *soft skills*) necesarias cuando se aborda el papel de jefe de proyecto DGS. El jugador tendrá que abordar la gestión de un proyecto DGS ficticio desde el comienzo hasta la entrega del producto software al cliente, tratando de resolver los diferentes impedimentos que se puedan ocasionar en el ciclo de vida. De esta manera, los jugadores podrán adquirir experiencia de una manera sencilla, barata e independiente, permitiéndoles afrontar con éxito un futuro trabajo en la gestión de un proyecto DGS.

Este TFG se enmarca dentro de un contrato I+D con el grupo de investigación Alarcos<sup>1</sup> de la Universidad de Castilla La-Mancha (UCLM)<sup>2</sup>, en concreto en el proyecto "G3SOFT: Ingeniería de Modelos para el Gobierno y Gestión del Desarrollo Global de Software"<sup>3</sup>, el cual se centra en la mejora del gobierno y la gestión de proyectos de DGS. En la tab. 1.1 se muestra un resumen de dicho proyecto.

<b>Nombre:</b>	G3SOFT: Ingeniería de Modelos para el Gobierno y Gestión del Desarrollo Global de Software
<b>Financiación:</b>	JJCM Consejería de Educación y Cultura y Deportes, y Fondos FEDER
<b>Referencia:</b>	SBPLY/17/180501/000150
<b>Dirección WEB:</b>	<a href="https://alarcos.esi.uclm.es/proyectos/G3SOFT/index.php">https://alarcos.esi.uclm.es/proyectos/G3SOFT/index.php</a>
<b>Grupo de investigación:</b>	Grupo Alarcos
<b>Universidad colaboradora:</b>	Universidad de Castilla-La Mancha (UCLM)
<b>Investigadores principales:</b>	Francisco Ruiz González Aurora Vizcaíno Barceló

**Tabla 1.1:** Resumen del proyecto G3SOFT

### 1.3. ESTRUCTURA DEL DOCUMENTO

A continuación, se define la estructura del documento, la cual hace referencia a la memoria del TFG y esta dividida en los siguientes capítulos:

- **Capítulo 1. Introducción:** breve indicativo sobre la motivación, contexto y problemática que engloba este TFG, al igual que la solución que se propone.
- **Capítulo 2. Objetivos:** listado tanto del objetivo principal como los objetivos secundarios que se persiguen con la realización de dicho TFG, además de las tareas necesarias para llevarlo a cabo.
- **Capítulo 3. Estado del arte:** información referente sobre el DGS, la gestión y administración de proyectos DGS, de igual modo se especifican un conjunto de habilidades, las cuales son necesarias para llevar a cabo el trabajo de jefe de proyecto software en un entorno distribuido. Por último, se define e indica la importancia de la gamificación en general, y de los JS en particular, a su vez de una serie de ejemplos sobre JS relacionados con el tema de este TFG.
- **Capítulo 4. Método de trabajo:** especificación de la metodología de trabajo que se seguirá para el desarrollo de este TFG, al igual que las herramientas y tecnologías (tanto software como hardware) que se utilizarán en dicho periodo.
- **Capítulo 5. Resultados:** informe de los resultados obtenidos tras llevar a cabo cada uno de los objetivos y tareas definidas en el capítulo 2, utilizando el método de trabajo definido en el capítulo 4, al igual que todos los posibles problemas e impedimentos que puedan haber surgido en la realización de este TFG.

<sup>1</sup><https://alarcos.esi.uclm.es>

<sup>2</sup><https://www.uclm.es/>

<sup>3</sup><https://alarcos.esi.uclm.es/proyectos/G3SOFT/index.php>

- **Capítulo 6. Conclusiones:** exposición de la conclusión y trabajos futuros tras haber realizado el presente TFG, del mismo modo que las lecciones aprendidas y una valoración personal.
- **Bibliografía:** sinopsis de las fuentes de información consultadas para la realización de este TFG.



# OBJETIVOS

---

Este capítulo se centrará en presentar y explicar de manera detallada cual es el objetivo principal que se persigue con la realización del presente TFG, al igual que los objetivos específicos, donde se especificarán aquellos objetivos funcionales y técnicos necesarios para la elaboración del proyecto. A modo de resumen, al final del presente capítulo en la Tabla 2.1, se mostrarán todos los objetivos que se han definido en el inicio del proyecto.

## 2.1. OBJETIVO PRINCIPAL

El principal objetivo (OP) del presente TFG consiste en diseñar y desarrollar una aplicación software de escritorio, la cual consistirá en un JS en 2.5D, o lo que también se conoce como *pseudo-3D*<sup>1</sup>, donde la tridimensionalidad de un video juego en 3D se limita a un plano de dos dimensiones. Este JS, al cual titularemos como *GLOBAL-MANAGER*, ayudará a estudiantes en ingeniería de software e ingenieros de software inexpertos del sector a adquirir ciertas *soft skills*, las cuales son necesarias para llevar a cabo una correcta gestión de un proyecto DGS. En las partidas del juego, los jugadores adquirirán el rol del jefe de proyecto en un entorno DGS, donde tendrá que gestionar dicho proyecto desde su creación hasta la entrega del producto a su correspondiente cliente. Por lo que, se le permitirá al jugador jugar diferentes partidas para que se pueda entrenar en dicha labor y poder afrontar con éxito en el futuro un posible trabajo de jefe de un proyecto DGS.

## 2.2. OBJETIVOS ESPECÍFICOS FUNCIONALES

El actual proyecto está compuesto por una serie de objetivos específicos funcionales (OEFs), los cuales han de tenerse en cuenta en el desarrollo del JS *Global-Manager*. Al final del TFG y en especial en el Capítulo 6, se llevará a cabo un análisis para comprobar si dichos objetivos han sido debidamente cumplimentados. Estos OEFs son los siguientes:

- **OEF1.** El JS diferenciará un total de tres niveles distintos de jugador. Estos niveles se calcularán utilizando técnicas de *inteligencia artificial* (IA), en función de los conocimientos del jugador en la gestión de proyectos y en el modelo de desarrollo de software DGS. Los niveles de usuarios que dispondrá el juego serán:
  - NIVEL BAJO: Se corresponderá a aquellos jugadores que posean bajos o nulos conocimientos en la gestión de proyectos o en el DGS, además de no tener una gran experiencia en el desarrollo de software. Son aquellas personas que necesitarán un aprendizaje mucho más lento y prolongado para adquirir todos los conocimientos necesarios para la gestión de proyectos DGS.
  - NIVEL MEDIO: Se corresponderá a aquellos jugadores que posean ciertos conocimientos en la gestión de proyectos o en el DGS, o incluso poseer cierta experiencia en dichos

---

<sup>1</sup><https://es.wikipedia.org/wiki/2.5D>

- campos. Estas personas necesitarán de un aprendizaje mucho más rápido, debido a que ya conocerán ciertos aspectos de la materia y les costará menos acostumbrarse al juego.
- NIVEL ALTO: Se corresponderá a aquellos jugadores que posean grandes conocimientos en la gestión de proyectos o en el DGS, además de haber trabajado en numerosas ocasiones en alguno de estos campos. En este último nivel, los jugadores no necesitarán tanto adquirir conocimientos, sino reforzarlos y entrenarse para mejorar su capacidad de gestionar proyectos DGS satisfactoriamente.
  - **OEF2.** El JS dispondrá de un *modelo de estudiante* dinámico, es decir, se creará a cada jugador un proceso de aprendizaje en función del nivel de jugador que se le haya definido al principio. Por lo tanto, las partidas que juegue dicho jugador se crearán automáticamente personalizadas a sus conocimientos.
  - **OEF3.** Las partidas del juego deberán estar divididas en dos fases. La primera fase consistirá en la configuración inicial del proyecto DGS ficticio, la cual estará compuesta por una interfaz gráfica donde se mostrarán un conjunto de parámetros, los cuales son necesarios conocerlos y configurarlos en un proyecto de estas características. El jugador tendrá que fijar cada uno de los parámetros. Además, en tiempo real, se calcularán un conjunto de factores de éxito del proyecto con la configuración actual y un nivel de dificultad del proyecto (el cual podrá tener los siguientes valores *muy fácil, fácil, normal, difícil, muy difícil*), que ayudará a saber si la configuración definida es correcta.
  - **OEF4.** La segunda fase consistirá en una simulación de un proyecto DGS, en donde el jugador adquirirá el rol de jefe de proyecto y deberá administrar dicho proyecto. El jugador tendrá que hacer frente y en algunas ocasiones solucionar diferentes situaciones, a las cuales llamaremos eventos, que puedan ocurrir en el ciclo de vida de un proyecto DGS real. Estos eventos estarán relacionados con las tres ces (los tres grandes desafíos de un proyecto DGS): comunicación, coordinación y control; y podrán ser tanto eventos positivos (repercusión favorable hacia el proyecto), como eventos negativos (repercusión desfavorable hacia el proyecto).

## 2.3. OBJETIVOS ESPECÍFICOS TÉCNICOS

Una vez definidos cuales serán los OEFs del proyecto en la Sección 2.2, es necesario definir otros objetivos específicos técnicos (OETs). Estos OETs son los siguientes:

- **OET1.** En el cálculo del nivel del jugador (anteriormente descrito) se deberán utilizar técnicas de IA, en especial utilizando *Lógica Borrosa*<sup>2</sup>. Al principio del juego y antes de que el jugador juegue una partida, deberá rellenar un formulario con diferentes preguntas sobre sus conocimientos y experiencia en gestión de proyectos y DGS. A través de estas preguntas se obtendrá un nivel para el nuevo jugador. Para implementar esta encuesta y el cálculo automático del nivel se llevará a cabo una entrevista a un experto en la materia, para adquirir aquellos conocimientos que nos hagan saber cuales son los aspectos importantes para conocer las nociones sobre gestión de proyectos y DGS de una persona.
- **OET2.** Desarrollar el JS utilizando el marco de trabajo de Microsoft *.NET*<sup>3</sup>. Además, se utilizará el lenguaje de programación *C#*<sup>4</sup>, el cual se encuentra más enfocado en la programación de videojuegos.
- **OET3.** La gestión de los datos asociados al proyecto (características de los eventos, modelo del estudiante y dominio) se llevará a cabo utilizando el sistema de gestión de bases de datos

<sup>2</sup>La lógica borrosa, también conocida como lógica difusa, consiste en un enfoque computacional basado en grupos de pertenencia (parcialmente verdadero o parcialmente falso), en vez de en la tradicional lógica booleana de verdadero o falso.  
<https://www.wikiversus.com/informatica/logica-difusa/>

<sup>3</sup><https://dotnet.microsoft.com/>

<sup>4</sup>[https://es.wikipedia.org/wiki/C\\_Sharp](https://es.wikipedia.org/wiki/C_Sharp)

relacional *SQLite*<sup>5</sup>, junto con el componente *LINQ*<sup>6</sup> (Language Integrated Query) de Microsoft .NET para las consultas a datos de manera nativa a los lenguajes .NET, como C#.

<i>Código Objetivo</i>	<i>Descripción</i>
<b>OP</b>	Diseño y desarrollo de un JS en 2.5D, que ayude a sus jugadores a adquirir ciertas soft skills y a entrenarse en la correcta gestión de proyectos DGS.
<b>OEF1</b>	Diferenciación de tres niveles de jugador (bajo, medio y alto) para crear así diferentes procesos de aprendizaje, en función de los conocimientos del jugador en la mataria.
<b>OEF2</b>	Creación de un modelo de estudiante dinámico para la creación de partidas personalizadas.
<b>OEF3</b>	Primera fase del juego: Implementación de una interfaz grafica, en donde el jugador deberá configurar un conjunto de parametros iniciales del proyecto DGS. Además del calculo, en tiempo real, de un conjunto de fáctores de éxito y el nivel de dificultad del proyecto.
<b>OEF4</b>	Segunda fase del juego: Implementación de la simulación del proyecto en función de la configuración inicial definida, el jugador deberá gestionar dicho proyecto DGS haciendo frente a diferentes eventos (positivos y negativos) relacionados con la comunicación, coordinación y control del proyecto.
<b>OET1</b>	Cálculo del nivel del jugador mediante un formulario, el cual se implementará utilizandológica borrosa, tras realizar una adquisición de conocimientos mediante una entrevista a un experto sobre cuales son los aspectos más importantes para conocer las nociones sobre gestión de proyectos y DGS de una persona.
<b>OET2</b>	Desarrollo del JS utilizando Microsoft .NET, junto con el lenguaje de programación C#.
<b>OET3</b>	Gestión de los datos asociados al proyecto mediante el sistema de gestión de bases de datos relacionales SQLite, junto con el componente LINQ de Microsoft .NET.

**Tabla 2.1:** Resumen de los objetivos del TFG

<sup>5</sup><https://sqlite.org/index.html>

<sup>6</sup><https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>





## ESTADO DEL ARTE

---

El objetivo de este capítulo consistirá en la definición y explicación detallada del argumento del presente TFG. A continuación, se presenta el estado del arte del método de desarrollo DGS, donde se detallará su definición, beneficios, problemáticas. Además, se indicará la importancia que conlleva la gestión en un proyecto de estas características. Por otra parte, se harán referencia a las habilidades que son necesarias para llevar a cabo el trabajo de jefe de proyecto en un entorno DGS. A continuación, se tratará la situación actual de la gamificación para la enseñanza, junto con los JS y el uso de la IA para la personalización de juegos. Para finalizar, se listarán una serie de ejemplos relacionados con el objetivo de este TFG.

### 3.1. DESARROLLO GLOBAL DEL SOFTWARE

Debido a la globalización, en el campo de la ingeniería de software aparece un nuevo modelo de desarrollo denominado como Desarrollo Global de Software. A diferencia del modelo tradicional, donde el equipo de trabajo estaba centralizado en un solo edificio o en varios edificios, pero siempre en el mismo país, el DGS consiste en el desarrollo de un producto o servicio software, en donde diferentes equipos de desarrollo, pertenecientes a organizaciones diferentes y ubicadas en países dispares, colaboran en el mismo proyecto software y son coordinados y gestionados en tiempo real basándose en los desafíos del DGS, las 3Cs: comunicación, coordinación y control [17].

A lo largo de la última década, el DGS se ha afianzado como una de las vertientes más relevantes en la investigación y práctica dentro del campo de la ingeniería del software. Las primeras prácticas de este nuevo tipo de proceso de desarrollo de software surgen hace más de 30 años, con los primeros usos de *outsourcing* [4]. Sin embargo, no sería hasta 2006 cuando se extendiera su uso con la celebración de la primera conferencia internacional sobre DGS, el *ICGSE (IEEE International Conference on Global Software Engineering)* [17, 19].

#### 3.1.1. Beneficios del Desarrollo Global del Software

Cuando hablamos de DGS es notable resaltar la cantidad de beneficios que pueden alcanzar las empresas con su correcto uso. Sin embargo, no todos sus beneficios son conocidos en el sector, ya que podemos encontrar beneficios que se puedan obtener de forma directa o indirecta, y son muchas las investigaciones, como [1, 9, 10, 19], que están estudiado tanto aquellos beneficios notables como aquellos no vistos a simple vista con el uso de DGS como proceso de desarrollo de software. Por lo tanto, a continuación se formulará un listado de todos los beneficios que podemos obtener con esta nueva tendencia:

- **Reducción del coste de producción.** Uno de los beneficios más importantes y una de las razones por las cuales las organizaciones y compañías están optando en afrontar un modelo de desarrollo de software global, consiste en la reducción de los costes del proyecto en el proceso de producción. Este beneficio se debe en gran medida a la globalización, ya que ha hecho posible

que actividades en el proceso de desarrollo puedan realizarlas empleados que se encuentran en países que cuentan con salarios más reducidos [1]. Un ejemplo de esta situación sería la India, en donde el salario base anual de un desarrollador de software es de *US\$15,000*, mientras que en Irlanda un mismo trabajador puede ganar cuatro veces esa cantidad, y a su vez esa cantidad consistiría en la mitad del salario de un desarrollador en Estados Unidos [9, 10]. Alcanzar esta situación también ha sido posible gracias al despliegue de enlaces de comunicación de alta velocidad, los cuales ayudan a la transferencia de información y conocimientos entre los empleados separados geográficamente [1]. Este beneficio conlleva que las compañías tengan que tener en cuenta la gestión del coste en cuanto a los viajes de empleados entre grupos de trabajo, ya que en este modelo de desarrollo los empleados no se conocen y es necesario que exista un poco de comunicación *face-to-face* para consolidar la relación entre empleados, creando así un ambiente de confianza [9, 10].

- **Aprovechamiento de la zona horaria.** En un entorno global, como es un proyecto que utiliza DGS como proceso de desarrollo, hace posible que las organizaciones puedan aprovecharse de la diferencia en las zonas horarias de sus empleados, con el fin de incrementar las horas de trabajo del día y reducir así el tiempo de desarrollo del servicio o producto software [9, 10]. De esta manera se lograrán jornadas de trabajo más extensas en el proyecto, que en uno con un modelo tradicional y por lo tanto una mayor productividad para finalizar dicho proyecto en un menor tiempo [19]. Esta situación es conocida como desarrollo *follow-the-sun*<sup>1</sup> y es considerada un potente beneficio en el DGS. Sin embargo, lograr este escenario en la realidad es complicado, además de poder producirse retrasos en las respuestas en la comunicación de los empleados, debido a horarios de comida o días festivos, lo que puede ocasionar retrasos en el proyecto. Por lo tanto, es importante que exista cierto solapamiento en las jornadas laborales de los empleados, con el fin de que exista cierta comunicación sincrónica [9, 10].
- **Modularización del proceso de desarrollo.** Una manera de afrontar un proyecto DGS es separando las tareas del proyecto en módulos independientes bien definidos, para que de esta manera cada equipo de desarrollo posea ciertos módulos de trabajo. Permitiendo así que las decisiones referentes a cada módulo se tomen de forma aislada entre los miembros del equipo de trabajo, además de reducir costes en la coordinación. De esta manera, podemos diferenciar dos tipos de estrategias: *basada en módulos*<sup>2</sup> y *basada en fases*<sup>3</sup> [1, 9, 10].
- **Acceso a plantillas de trabajadores altamente cualificados.** El DGS hace posible acceder fácilmente a grupos de trabajadores altamente cualificados repartidos por todo el mundo. De esta manera, las empresas se pueden beneficiar de los conocimientos, diversidad de experiencias, destrezas y habilidades de trabajadores repartidos por todo el mundo, para que lleven a cabo el desarrollo de diferentes actividades software del proyecto DGS [1, 9, 10, 19].
- **Proximity to market and customer.** Gracias al DGS, las compañías pueden establecer fácilmente filiales en aquellos países donde se localizan los clientes, y conseguir así un acercamiento y un conocimiento más profundo del mercado local. Además, con esta práctica las compañías consiguen expandirse hacia nuevos mercados, sin la necesidad de trasladar a sus equipos de desarrolladores [1, 9, 10, 19].

### 3.1.2. Desafíos del Desarrollo Global del Software

La distancia existente entre los equipos de desarrolladores en un proyecto DGS hace posible el acercamiento hacia grandes beneficios por parte de las organizaciones, sin embargo conseguirlos no

<sup>1</sup>"La estrategia *follow-the-sun* se caracteriza en que cuando un equipo de trabajo finaliza su jornada laboral, la jornada de otro equipo comienza en otra parte del mundo, de esta manera se consigue un desarrollo del proyecto las 24 horas del día"[17]

<sup>2</sup>"La estrategia basada en módulos consiste en dividir el proyecto en diferentes módulos, los cuales se pueden considerar un artefacto completo del proyecto, y repartirlos entre los sites"[17]

<sup>3</sup>"La estrategia basada en fases consiste en asignar a cada equipo de trabajo una fase del proceso de desarrollo software"[17]

es tarea sencilla, ya que dicha distancia conlleva también la introducción de un conjunto de desafíos, a los cuales, las organizaciones deben hacer frente para alcanzar los beneficios anteriormente citados [9]. Los desafíos que puedan aparecer en este tipo de proyectos se pueden agrupar en relación con los tres grandes procesos en el desarrollo de software: comunicación, coordinación y control. Estos grupos de desafíos también se les conoce como las 3 Cs [17, 19], y hacen referencia a las siguientes situaciones:

- **Desafíos de comunicación.** Hace referencia a aquellas situaciones en donde se lleva a cabo una comunicación entre trabajadores, es decir, un intercambio de información y conocimientos con el fin de que no se produzcan malentendidos y el proyecto pueda avanzar.
- **Desafíos de coordinación.** Hace referencia al mantenimiento de los trabajadores en la realización de las diferentes tareas de un proyecto, con el fin de alcanzar objetivos e intereses comunes y la evolución del proyecto progrese adecuadamente.
- **Desafíos de control.** Hace referencia a la administración y gestión del proyecto en general, teniendo en cuenta día a día diferentes aspectos del proyecto como pueden ser los calendarios de entregas, presupuestos, calidad, estándares, etc.

Por otro lado, los proyectos DGS se caracterizan por la existencia de diferentes nacionalidades, organizaciones e incluso culturas que pueden agravar esta situación. Esto hace que puedan aparecer diferentes tipos de distancias entre los miembros del proyecto, provocando una acentuación en los desafíos anteriormente citados [19]. De esta manera, aparece otra clasificación de los desafíos de un proyecto DGS en función de lo que se conoce en la literatura como las tres distancias [9, 10, 19]:

- **Distancia geográfica.** Se puede definir como la medida de esfuerzo necesario en un individuo para visitar un punto alejado de su ubicación. Un ejemplo sería el de dos ubicaciones separadas geográficamente por una gran distancia pero con un enlace aéreo directo frente a dos ubicaciones cercanas geográficamente pero con poca infraestructura de transporte; de esta manera el primer caso poseerá poca distancia geográfica, mientras que en el segundo será elevada [19]. Además, la distancia geográfica dificulta la posibilidad de existir una comunicación informal o cara a cara entre los miembros que ayude a afianzar las relaciones entre ellos, el trabajo en equipo, consolidación de la confianza y la comunicación fluida de información importante del proyecto [9].
- **Distancia temporal.** Ligada a la anterior, se puede definir como la medida de la diferencia en el tiempo existente en la comunicación entre dos individuos [19]. Como se dijo en la sección anterior, la distancia temporal en un proyecto DGS puede conllevar ciertos desafíos como es la estrategia follow-the-sun, con el objetivo de reducir tiempo y costes, sin embargo, surgen también ciertas problemáticas como es el hecho de que aparezcan retrasos en las respuestas en los intercambios de información entre los trabajadores, debido a la reducción del solapamiento de horas en las jornadas laborales de los equipos de desarrolladores. Esto implica que se tengan que usar herramientas de comunicación asíncronas, las cuales pueden afectar negativamente al manejo de ambigüedades y al aumento de los malentendidos, produciéndose así retrasos en el proyecto; frente a una comunicación síncrona mucho más directa y segura [9].
- **Distancia socio-cultural.** Se puede definir como la medida en que un individuo conoce y comprende las costumbres, cultura e idioma de otro individuo, con el objetivo de llevar a cabo una correcta comunicación. Esta situación es debida a que como en un proyecto coexisten diferentes nacionalidades, es frecuente que existan diferentes culturas entre sus miembros [19]. Dicha distancia puede conllevar diferentes interpretaciones en una comunicación o situación, lo que puede obstaculizar la comunicación y coordinación del proyecto [9]. Además, puede provocar que aparezcan conflictos y malentendidos entre los miembros del proyecto, retrasando así la evolución del mismo.

Por lo tanto, es importante tener en cuenta los posibles desafíos que puedan surgir en un proyecto DGS, es por ello que son numerosas las investigaciones sobre estas problemáticas, como es el caso de

[16], en donde se lleva a cabo una Revisión Sistemática de la Literatura (RSL) de 101 estudios sobre las dificultades más importantes en un proyecto de estas características, junto con un cuestionario a 41 organizaciones sobre sus prácticas en el mundo real. A continuación, en la Tabla 3.1, se mostrará un resumen con las dificultades más importantes y comunes en proyectos DGS.

**Tabla 3.1:** Resumen de las dificultades más importantes en proyectos DGS

Desafío DGS	Frecuencia de aparición en 101 estudios (%)	Frecuencia de acuerdo con 41 organizaciones (%)	Media (%)
Falta de entendimiento cultural	62	70	66
Ausencia de comunicación	54	76	65
Falta de la gestión del conocimiento	38	78	58
Falta de gestión del tiempo	42	71	56.5
Ausencia de coordinación	35	69	52
Ausencia de control	27	75	51
Actividades de ingeniería de requisitos	28	71	49.5
Asignación de tareas	18	80	49
Ausencia de la verdad	34	59	46.5
Actividades de gestión del cambio	22	68	45
Falta en la concienciación de equipo	23	66	44.5
Gestión de conflictos	17	71	44
Estimación del coste y del esfuerzo	15	73	44
Actividades de integración	14	73	43.5
Gestión del riesgo	15	71	43
Distancia geográfica	28	56	42
Ausencia de un proceso uniforme entre los diferentes sites	19	63	41
Falta de una infraestructura informática adecuada	11	68	39.5
Protección de la propiedad intelectual	9	64	36.5

### 3.1.3. Rol del jefe de proyecto

## 3.2. HABILIDADES NECESARIAS EN DESARROLLO GLOBAL DEL SOFTWARE

### 3.2.1. Habilidades en el equipo de trabajo de Desarrollo Global del Software

### 3.2.2. Habilidades en jefes de proyecto de Desarrollo Global del Software

## 3.3. GAMIFICACIÓN

### 3.3.1. Juegos Serios

## 3.4. TRABAJOS RELACIONADOS CON EL TEMA

### 3.4.1. Juegos Serios para Desarrollo Global del Software

### 3.4.2. Juegos Serios para Jefes de Proyecto

### 3.4.3. Juegos Serios para Jefes de Proyecto en Desarrollo Global del Software

# MÉTODO DE TRABAJO

---

## **4.1. SCRUM**

### **4.1.1. Roles**

### **4.1.2. Componentes de Scrum**

## **4.2. DESARROLLO BASADO EN PROTOTIPOS**

### **4.2.1. Etapas del modelo de prototipos**

## **4.3. MARCO TECNOLÓGICO**

### **4.3.1. Herramientas Software**

### **4.3.2. Herramientas Hardware**



# **RESULTADOS**

---

## **5.1. SPRINT 0**

### **5.1.1. Equipo Scrum**

### **5.1.2. Alcance del Proyecto**

#### **5.1.2.1. Requisitos Funcionales**

#### **5.1.2.2. Requisitos No Funcionales**

### **5.1.3. Pila del Producto**

### **5.1.4. Planificación del Proyecto**

#### **5.1.4.1. Historias de usuario**

#### **5.1.4.2. Casos de uso**

##### **5.1.4.2.1. Actores**

##### **5.1.4.2.2. Casos de uso**

## **5.2. SPRINT 1**

## **5.3. SPRINT 2**





# CONCLUSIONES Y TRABAJO FUTURO

---

- 6.1. CONCLUSIÓN
- 6.2. LECCIONES APRENDIDAS
- 6.3. TRABAJO FUTURO
- 6.4. PUBLICACIONES
- 6.5. VALORACIÓN PERSONAL



# BIBLIOGRAFÍA

---

- [1] Pär J Ågerfalk y col. «Benefits of global software development: the known and unknown». En: *International Conference on Software Process*. Springer. 2008, págs. 1-9.
- [2] Afra A Alabbadi y Rizwan J Qureshi. «The proposed methods to improve teaching of software engineering». En: *International Journal of Modern Education and Computer Science* 8.7 (2016), pág. 13.
- [3] Sarah Beecham y col. «How best to teach global software engineering? Educators are divided». En: *IEEE Software* 1 (2017), págs. 16-19.
- [4] Barry Boehm. «A view of 20th and 21st century software engineering». En: *Proceedings of the 28th international conference on Software engineering*. 2006, págs. 12-29.
- [5] Ivana Bosnić, Igor Čavrak y Mario Žagar. «Assessing the impact of the distributed software development course on the careers of young software engineers». En: *ACM Transactions on Computing Education (TOCE)* 19.2 (2019), págs. 1-27.
- [6] Lisa J Burnell, John W Priest y JB Durrett. «Teaching distributed multidisciplinary software development». En: *IEEE software* 19.5 (2002), págs. 86-93.
- [7] Alejandro Calderón, Mercedes Ruiz y Rory V O'Connor. «A multivocal literature review on serious games for software process standards education». En: *Computer Standards & Interfaces* 57 (2018), págs. 36-48.
- [8] Eun Man Choi. «Applying inverted classroom to software engineering education». En: *International Journal of e-Education, e-Business, e-Management and e-Learning* 3.2 (2013), pág. 121.
- [9] Eoin Ó Conchúir y col. «Exploring the assumed benefits of global software development». En: *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*. IEEE. 2006, págs. 159-168.
- [10] Eoin Ó Conchúir y col. «Global software development: where are the benefits?». En: *Communications of the ACM* 52.8 (2009), págs. 127-131.
- [11] Thomas M Connolly, Mark Stansfield y Thomas Hainey. «An application of games-based learning within software engineering». En: *British Journal of Educational Technology* 38.3 (2007), págs. 416-428.
- [12] Jose Eduardo Nunes Lino y col. «Project management game 2D (PMG-2D): A serious game to assist software project managers training». En: *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2015, págs. 1-8.
- [13] Andrew Meneely y Laurie Williams. «On preparing students for distributed software development with a synchronous, collaborative development platform». En: *Proceedings of the 40th ACM technical symposium on Computer science education*. 2009, págs. 529-533.
- [14] Miguel J Monasor y col. «Preparing students and engineers for global software development: a systematic review». En: *2010 5th IEEE International Conference on Global Software Engineering*. IEEE. 2010, págs. 177-186.

- [15] Christian Murphy, Dan Phung y Gail Kaiser. «A distance learning approach to teaching eXtreme programming». En: *Proceedings of the 13th annual conference on Innovation and technology in computer science education*. 2008, págs. 199-203.
- [16] Mahmood Niazi y col. «Challenges of project management in global software development: A client-vendor analysis». En: *Information and Software Technology* 80 (2016), págs. 1-19.
- [17] Mario Piattini Velthuis, Aurora Vizcaíno Barceló y Félix García Rubio. «Desarrollo global de software». En: *RAMA* (2014).
- [18] Rafael Prikladnicki y Leonardo Pilatti. «Improving contextual skills in global software engineering: A corporate training experience». En: *IEEE International Conference on Global Software Engineering*. IEEE. 2008, págs. 239-243.
- [19] Aurora Vizcaíno, Félix García y Mario Piattini. «Visión general del desarrollo global de software». En: *International Journal of Information Systems and Software Engineering for Big Companies (IJISEBC)* 1.1 (2015), págs. 8-22.