



ARQUITECTURA Y DESARROLLO DE UN SISTEMA BACKEND BASADO EN  
MICROSERVICIOS PARA LA ADMINISTRACIÓN DEL FLUJO DE INFORMACIÓN QUE  
MANEJAN LOS VIDEO JUEGOS

PROYECTO DE TESIS PRESENTADO PARA EL CUMPLIMIENTO DE LOS REQUISITOS  
DE GRADO DE PARA LA CARRERA DE INGENIERIA DE SISTEMAS EN LA  
INSTITUCIÓN UNIVERSITARIA POLITECNICO GRANCOLOMBIANO

CARLOS MARIO SANZ VELASCO

DICIEMBRE 2021



ARQUITECTURA Y DESARROLLO DE UN SISTEMA BACKEND BASADO EN  
MICROSERVICIOS PARA LA ADMINISTRACIÓN DEL FLUJO DE INFORMACIÓN QUE  
MANEJAN LOS VIDEO JUEGOS

ESTUDIANTES:

CARLOS MARIO SANZ VELASCO

DIRECTOR: ISABEL MAHECHA

INSTITUCION UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS

INGENIERÍA DE SISTEMAS

BOGOTÁ

2021

Certifico que he leído este proyecto de grado y que, en mi sentir, es completamente adecuado en alcance y calidad como proyecto para el grado de ingeniero de sistemas.

---

Certifico que he leído este proyecto de grado y que, en mi sentir, es completamente adecuado en alcance y calidad como proyecto para el grado de ingeniero de sistemas.

---

## **DEDICATORIA**

*Dedico este proyecto de grado a mi Dios, mis padres, mi hermano y mi pareja porque en ellos siempre encuentro el apoyo necesario e incondicional para sobreponerme a las dificultades y poder salir adelante.*

*Por otro parte, también se lo dedico a los profesores y estudiantes de la universidad que tuvieron que ver con mi análisis e investigación, en especial a la profesora Isabel Mahecha, al profesor Gabriel Buitrago, y al estudiante de Maestría en Matemáticas del Politécnico Grancolombiano, Juan David Sánchez Bravo.*

*- Carlos Mario Sanz Velasco.*

## **AGRADECIMIENTOS**

Primero que todo, quiero agradecerle a Dios, porque, a pesar de las dificultades, me ha dado la fuerza mental y física para poder siempre persistir en el avance y finalización de este proyecto y de mi carrera.

También quiero agradecerle a mi madre y a mi padre, porque siempre me han apoyado a pesar de las circunstancias, me han aconsejado de la forma correcta, y me han hecho sentir mejor en los momentos difíciles. Sus consejos no solo Han sido para fortalecerme mentalmente, sino que también, para hacerme una mejor persona cada día.

A mi pareja que, también, me ha apoyado y ha sacrificado tiempos de nuestra relación, para que yo pueda crecer como persona y profesional.

**Carlos Mario Sanz Velasco**

## **RESUMEN**

En el siguiente proyecto se implementa un sistema de capa intermedia mediante el cual se desarrollarán microservicios para la administración y el flujo de información de un video juego educativo del area de matemáticas, específicamente, de la asignatura de álgebra para los estudiantes de bachillerato.

Para la codificación del sistema se utilizó Java como lenguaje de programación, Springboot como framework para los microservicios, MySql como motor de base de datos, y Scrum como metodologíade trabajo y desarrollo.

Finalmente, el resultado de la investigación fue un sistema de microservicios, los cuales funcionan bajo estándares de calidad, y darán cumplimiento a una serie de tareas las cuales van a ser necesarias para el funcionamiento integral del video juego.

## TABLA DE CONTENIDO

INTRODUCCIÓN.....	31
GENERALIDADES .....	32
2.1 Antecedentes .....	32
2.2 Planteamiento del problema. ....	34
2.3 Objetivos. ....	37
2.3.1 Objetivo General. ....	37
2.3.2 Objetivos Específicos.....	37
2.4 Justificación.....	37
2.5 Delimitación .....	38
2.5.1 Alcance.....	38
2.6 Metodología.....	40
MARCO TEORICO .....	46
DESARROLLO DEL PROYECTO.....	54
4.1 Fase de levantamiento de información y diseño Funcional .....	54
RESULTADOS .....	67
5.1 Resultados del Desarrollo e implementación .....	67
CONCLUSION.....	72
6.1 Objetivos .....	72

6.2 Pregunta de Investigación.....	74
6.3 Conclusiones Generales.....	75
TRABAJO FUTURO.....	76
REFERENCIAS.....	76
ANEXOS.....	80
9.1 Manual Técnico.....	80
9.2 Manual de Instalación y Configuración.....	80
9.3 Manual de Despliegue.....	81

## INDICE DE FIGURAS

<b>Figura 1</b> .....	61
<b>Figura 2</b> .....	61
<b>Figura 3</b> .....	62
<b>Figura 4</b> .....	63
<b>Figura 5</b> .....	63
<b>Figura 6</b> .....	64
<b>Figura 7</b> .....	65
<b>Figura 8</b> .....	67
<b>Figura 9</b> .....	68
<b>Figura 10</b> .....	69
<b>Figura 11</b> .....	70
<b>Figura 12</b> .....	71
<b>Figura 13</b> .....	72
<b>Figura 14</b> .....	73
<b>Figura 15</b> .....	74
<b>Figura 16</b> .....	75
<b>Figura 17</b> .....	76
<b>Figura 18</b> .....	76
<b>Figura 19</b> .....	77
<b>Figura 20</b> .....	77
<b>Figura 21</b> .....	77
<b>Figura 22</b> .....	78
<b>Figura 23</b> .....	79
<b>Figura 24</b> .....	79
<b>Figura 25</b> .....	81

<b>Figura 26</b> .....	82
<b>Figura 27</b> .....	83
<b>Figura 28</b> .....	84
<b>Figura 29</b> .....	85
<b>Figura 30</b> .....	88
<b>Figura 31</b> .....	89
<b>Figura 32</b> .....	90
<b>Figura 33</b> .....	91
<b>Figura 34</b> .....	91
<b>Figura 35</b> .....	92
<b>Figura 36</b> .....	92

## **LISTA DE ANEXOS**

Anexo 4 Manual Técnico.....	82
Anexo 4 Manual de Despliegue .....	82
Anexo 4 Manual de instalación 56.....	82
Anexo 4 Plan de Trabajo .....	83
Anexo 4 ScriptMysql .....	83
Anexo 4 Fuentes.....	83

## GLOSARIO

**Ambientación:** Proceso por el cual se realizan las diferentes instalaciones y configuraciones de archivos necesarios para emular un mismo sistema que generalmente está en un servidor de producción.

**Apache Tomcat:** Es un contenedor o servidor de archivos que generalmente se usa para hacer despliegues de componentes java.

**Api:** Un api es una interfaz que ofrece funcionalidades para ser reusadas por otros sistemas.

**Capa de desarrollo:** O también llamado backend, es un nivel en el desarrollo de aplicaciones, que se encarga de implementar la lógica para el manejo de la información que entra y sale de una base de datos.

**Base de datos:** Bancos de información que generalmente se utilizan para guardar los datos de un sistema.

**Componente de software:** Es un fragmento de código de programación que esta creado para una función específica.

**Ciclo de vida del software:** Es el tiempo en el que un software va a funcionar.

**IDE:** Es el programa que se utiliza para compilar y desarrollar un software.

**Java:** Lenguaje de programación que se ejecuta del lado del servidor y sirve para crear aplicaciones ya sea de escritorio o de ejecución en la web.

**Eclipse:** Programa o IDE, con múltiples herramientas de programación, mediante el cual se compilan y desarrollan, generalmente, aplicaciones del lenguaje de programación java.

**Errores de software:** Son fallos que tiene algún componente de software de la aplicación.

**Framework:** En desarrollo de software, es una estructura de trabajo mediante la cual el desarrollador se soporta para realizar ciertos procesos de una forma más fácil y práctica.

**Frontend:** Nivel de desarrollo donde generalmente se manipulan los elementos de las vistas de un programa.

**HTTP:** Es el protocolo por donde viajan los mensajes de hipertexto.

**Incidencias:** Serie de errores que presenta un sistema de información o software.

**Lenguaje de programación:** Es el lenguaje que entiende y manipula desde una computadora el ser humano para construir aplicaciones de software y que posteriormente, son compiladas para que la maquina pueda entenderlas por medio de código binario.

**Manual Técnico:** Documento para hacer el registro de la descripción de los componentes técnicos de un sistema que se construyó.

**Manual de instalación:** Documento para describir los archivos y programas necesarios para realizar un desarrollo de software.

**Manual de Despliegue:** Documento que establece los diferentes archivos para realizar el proceso de despliegue y funcionamiento de un software.

**Manual de Componentes;** Documento que describe los componentes de software que intervienen en su lógica y flujo de información.

**Metodologías ágiles:** Son paradigmas de trabajo que ofrecen resultados, de un desarrollo, periódicamente sin la premisa de que todo el sistema haya sido terminado.

**Mockups:** Diseños de vistas de un sistema que generalmente se hacen con un previo estudio por profesionales de las áreas de UX – UI.

**PHP:** Lenguaje de programación que se ejecuta del lado del servidor y sirve para el desarrollo de sistemas de información web.

**Protocolo:** Regla que tiene una comunicación de datos entre sistemas, en algunos casos, estas reglas están diseñadas por diferentes capas, con el fin de que haya mayor seguridad en la información que está viajando.

**Reglas negocio:** Son los requerimientos que hacen únicos a los procesos que maneja una empresa.

**Rest:** Estilo arquitectónico que se utiliza para construir servicios o microservicios web. Su transferencia de información se basa en los métodos del protocolo HTTP, como lo son: Post, Put, Patch, Get, Option y Delete.

**Sistema:** Conjunto de elementos relacionados con un fin en común.

**Sistema de información:** Conjunto de elementos que están correlacionados para realización el envío y recepción de datos.

**Soap:** Arquitectura que ofrece protocolos de información para que sistemas de diferente naturaleza se puedan comunicar.

**Usuarios de un sistema:** Son aquellos actores que van a manipular desde la parte final de un sistema, todas sus funcionalidades.

**Workspace:** Es el espacio de trabajo que se utiliza para guardar los archivos y avances del desarrollo de un software.

## ACRÓNIMOS

**API:** (Application Programming Interfaz) Interfaz de Programación de Aplicaciones.

**DCL:** (Data Control Language) Lenguaje de control de datos.

**DDL:** (Data Definition Language) Lenguaje de definición de datos.

**DML:** (Data Manipulation Language) Lenguaje de manipulación de datos.

**HTTP:** (Hypertext Transfer Protocol) Transferencia de protocolo de hipertexto.

**JDBC:** (Java Database Connectivity) Conectividad de bases de datos con java.

**JPA:** Java Persistence Api.

**JSP:** (Java Server Pages) Paginas de mensajes Java.

**ORM:** (Object Relational Mapping) Objeto de mapeo relacional.

**SQL:** (Structured Query Language) Lenguajes de sentencias estructuras.

**REST:** (Representational State Transfer) Representación de estado de Transferencia.

## INTRODUCCIÓN

En el siguiente proyecto se realizó el estudio, la investigación y el desarrollo de un sistema de microservicios para mejorar el desempeño de un video juego creado para el aprendizaje de la asignatura de Álgebra.

Antes de empezar, es importante tener en cuenta la evolución que han tenido los video juegos y sistemas informáticos, quienes casi como hermanos, han logrado evolucionar cada uno en materia de hardware y software desde el año 1950, pasando por consolas, para los videojuegos, de un solo jugador, y de gráficas de muy baja calidad hasta consolas con procesadores de 8 núcleos, RAM de 8 Gigas y tarjetas gráficas y unidades GPU que ayudan a un alto estándar de visualización, y por el lado de los sistemas, se ha visto la evolución desde el software que se hicieron en combinación con elementos electrónicos como por ejemplo relojes, dispositivos de medición electrónico, partes de computadoras, entre otros, hasta las aplicaciones que conocemos hoy en día que reciben y manejan un alto número de peticiones, procesos y comunicación con otros componentes de software

Por ende, en este proyecto, se realiza el desarrollo de un sistema de información el cual, se va a construir en base a un paradigma modular llamado microservicios, que va a servir como backend de un video juego, el cual cada día cobra más fuerza, y es implementado por diferentes compañías

por su gran capacidad de adaptarse a diferentes ecosistemas de software y cumplir con estándares de calidad que puedan garantizar el buen desempeño de dicho sistema.

Es a causa de lo anterior, que por medio de este proyecto se pretende realizar un aporte el cual puede servir para que diferentes profesionales del gremio que quieran desarrollar proyectos que giren alrededor del mismo tema, tengan una guía, y un punto de vista el cual les sirva como soporte para otras implementaciones tecnológicas.

## GENERALIDADES

### 2.1 Antecedentes

En los últimos años, tanto a nivel nacional como a nivel mundial se han realizado diferentes investigaciones acerca de las arquitecturas adecuadas para desarrollar diferentes tipos de software como videojuegos, aplicaciones móviles, portales web, sistemas de información entre otras.

Una de estas investigaciones fue hecha por la Universidad Católica de Colombia, llamada *“Desarrollo de un Sistema de Información para la Gestión de Proyectos de Responsabilidad Social del Programa de Ingeniería de Sistemas de la Universidad Católica de Colombia ”* (Pazcagaza Gutierrez, págs. 1,2) el cual tenía como propósito implementar un sistema (con el lenguaje php y su framework Laravel) que le ayudará, mediante una arquitectura modular, ya que dicha gestión se realizaba físicamente guardando alrededor de 100 proyectora a optimizar el manejo de la información ya que todo lo estaban llevando en discos compactos, y además, llevaban un Excel para salvar el registro de las ubicaciones. Al final, se entregó el sistema desarrollado y probado el cual evidenció el mejoramiento que hubo en cada uno de los procesos y documentaciones de la universidad.

También, tenemos, *“la arquitectura de software basada en microservicios para el desarrollo de aplicaciones web de la asamblea nacional”*, (Lopez Hinojosa, Arquitectura de Software Basada en Microservicios Para el desarrollo de Aplicaciones Web de La Asamblea Nacional, 2017, págs. 1,2) el cual fue un proyecto de grado del programa de Maestría que se realizó en la universidad Técnica del Norte que pretendía mostrar los beneficios de una arquitectura de microservicios, para reemplazar la arquitectura monolítica.

Por otro lado, En la Universidad de San Buenaventura de Cali, se desarrolló un trabajo llamado *“Diseño y construcción de un componente para el generador de código ZaturaCode que genere proyectos con arquitectura basada en microservicios en java”* (García Victoria & Daza Gonzalez, 2017, págs. 1,2) el cual tiene como finalidad desarrollar un componente de software que esté siguiendo una arquitectura de microservicios el cual refleja los beneficios a nivel de mantenibilidad y una comunicación segura entre aplicaciones.

Finalmente, tenemos un proyecto, llamado *“Investigación y desarrollo de un Videojuego,* (Villardón García, 2017) realizado por parte de la facultat de Belles Arts de Sant Carles de la Universitat Politècnica de Valencia. En este documento se describía el desarrollo de una aplicación móvil para IOS, el cual iba a ser un prototipo de la parte visual de un videojuego y se iba manejar por medio de una arquitectura monolítica para el desarrollo de sus componentes demostrando en ciertas partes del proceso que tuvo retrasos y varios tipos de vulnerabilidades.

## 2.2 Planteamiento del Problema.

Se ha realizado un video juego educativo que consiste en ayudar a los estudiantes en aprender a solucionar problemas y ejercicios para la asignatura de álgebra. Estos problemas están enfocados, inicialmente, en algunos temas específicos como operaciones entre ecuaciones de primer grado, operaciones entre ecuaciones de segundo grado, y solución de ecuaciones algebraicas por medio de los casos de factorización.

¿Bajo qué concepto se escogió el video juego como medio de aprendizaje?

Se escogió bajo el concepto o técnica llamada “gamificación”.

¿Por qué se escogió el algebra como tema de estudio para el video juego?

Se escogió porque el álgebra es la raíz del cálculo, y es la raíz de la trigonometría, por lo tanto, se considera como la parte más importante de las matemáticas escolares. Es el tema más difícil de asimilar para un niño, porque es donde comienza a ver el termino de manejo de variables.

Para poder hacer la implementación del video juego solo se tenía planeado trabajar sobre una capa donde se pudiera realizar la programación de su componente lógico y su componente gráfico, es

decir, su desarrollo iba a ejecutarse a través de una arquitectura monolítica que manejara todos sus componentes por medio de unos únicos archivos de configuración, sin embargo, al avanzar en el desarrollo, y tener más claro toda su estructura, se pudo evidenciar la falta de un componente que fuera capaz de manejar aquellos datos que pudieran influir en el video juego de forma significativa como por ejemplo: los puntajes, las partidas, los avances de los jugadores, los ejercicios resueltos, datos de los usuarios y roles entre otros. Estos datos que serán necesarios para el funcionamiento y la medición del video juego, tenían proyectado seguir creciendo en la medida que los estudiantes de las diferentes instituciones educativas lo fueran utilizando, por lo tanto, era necesario desarrollarlo, no en base a una arquitectura monolítica, si no a través de una arquitectura modular como la arquitectura de microservicios que permitiera desarrollar el sistema de información de tal forma que pudiéramos garantizar el bajo acoplamiento, el mantenimiento, la seguridad, la escalabilidad y la eficiencia del video juego.

**Bajo acoplamiento:** Al ser una arquitectura monolítica, no permite trabajar de forma independiente lo que va a ser el desarrollo de la lógica y gráficos del video juego de forma independiente al desarrollo del manejo y control de la información que se va a manejar para el funcionamiento de este.

**Mantenimiento:** El video juego como cualquier otro tipo de software, durante su ciclo de vida, puede manifestar diferentes clases de errores o fallos en sus componentes, estos fallos, en primer lugar, pueden generar incidencias globales; en segundo lugar, para poder solucionar estas incidencias y analizar los errores que se estén presentando, sería necesario, hacer todo un proceso

despliegue agregando el paso de la detención de todos los componentes como los archivos del video juego, el servidor, los archivos de configuración de la base de datos entre otros.

Escalabilidad: Para la evolución y el crecimiento del video juego, se irán haciendo nuevas implementaciones, como por ejemplo crear o cambiar funcionalidades, adaptar ciertas reglas de negocio o incluso insertar nuevas fuentes de información que el juego necesite para poder seguir funcionando. Aquí se vuelve a la misma problemática que se manifiesta en el mantenimiento de un software y es que todo el sistema va a tener que ser detenido para poder hacer algún cambio o para poder incluir algún tipo de desarrollo nuevo.

Seguridad: Es muy importante tomar precauciones respecto a la seguridad informática y más para aplicaciones donde se tomará información personal pues en caso de que se encuentren vulnerabilidades y estas se lleguen a explotar o a utilizar pues se pueden exponer datos de suma importancia que pueden llegar a afectar a los clientes.

Además de lo planteado anteriormente, se debe tener en cuenta que la implementación de un videojuego así sea uno que se ejecute en línea, corre también por ciertos riesgos que no son dados precisamente por vulnerabilidades dentro del desarrollo de aplicaciones si no debido a factores externos, tales como vulnerabilidades por las redes que se usan o las inyecciones SQL.

## **2.3 Objetivos.**

### **2.3.1 Objetivo General.**

Proponer un sistema de información que esté implementado bajo una arquitectura de software que permita la administración, relación y el flujo de la información de un video juego para el aprendizaje de la asignatura de Álgebra.

### **2.3.2 Objetivos Específicos.**

- 1.** Identificar una arquitectura de software que pueda garantizar los estándares de calidad de la escalabilidad, mantenibilidad, seguridad y bajo acoplamiento del video juego.
- 2.** Estructurar una base de datos que pueda servirle al video juego como banco de información, para que este pueda almacenarla y consultarla de una manera más segura y eficiente.
- 3.** Desarrollar un sistema de información para manejar la información relevante del video juego que va a ayudar a los estudiantes de bachillerato a mejorar el aprendizaje de la asignatura de Álgebra.

## **2.4 Justificación.**

Actualmente, el desarrollo de video juegos se hace principalmente por medio de una herramienta llamada “Unity” la cual ofrece una serie de funcionalidades como animaciones, motores gráficos y físicos, sonidos etc.

Dentro de las tantas funcionalidades que se prestan, existe una en particular que se utiliza para construir desde “Unity” el backend de los video juegos que se generen, el cual es un módulo que ofrece el lenguaje de programación C# con el cual no solo se puede desarrollar lógica que necesite el video juego para el manejo de la información, sino que además maneja unas librerías para poder conectarse a componentes externos como bases de datos o servicios web. Este tipo de arquitectura donde todos los componentes de software como los gráficos, animaciones, backend o bases de datos se manejan a través de un solo entorno de desarrollo, y bajo unos mismos archivos de configuraciones y despliegues, se le llama arquitectura monolítica.

Sin embargo, la construcción de un componente backend por medio de una arquitectura monolítica genera diferentes inconvenientes para el equipo de desarrollo del video juego y para los usuarios de este; desde problemas de seguridad como inyecciones de ataques SQL hasta problemas de performance de la aplicación por los procesos en los despliegues a causa de los mantenimientos o mejoras.

Por ende, el desarrollo de este proyecto se enfocará en estudiar y desarrollar un sistema de información, proponiendo una arquitectura de software modular basada en una arquitectura de

microservicios, que actuará como capa intermedia e independiente, permitiéndole, al desarrollador del video juego enfocarse en la concepción, diseño y ejecución de este, y a su vez, este tipo de software se adaptará a trabajar dinámicamente, es decir permitirá trabajar con metodologías ágiles, lo cual será importante para la evolución del video juego.

Por otro lado, al utilizar este tipo de arquitectura de software, generará que el sistema del video juego pueda ser versátil en la construcción de su capa backend, es decir, podrá utilizar diferentes lenguajes y tecnologías de programación, para desarrollarlo, como por ejemplo java, javascript, node, .net entre otros ; también, ayudará a realizar los escalamientos e integraciones con otro tipo de aplicaciones, sistemas, plugins, base de datos o servicios web ya que las peticiones y respuestas de información no viajarán a través de las funciones o métodos de las clases o los componentes del lenguaje, si no que su comunicación se regirá a partir de los métodos proporcionados por el protocolo HTTP entre los cuales encontramos el PUT, POST, GET y DELETE; adicional, podrá apoyarse en herramientas que ofrecen los frameworks para poder alimentar el sistema con diferentes servicios que soportarán su seguridad, y la posibilidad de generar un proceso de abstracción de cada uno de sus componentes de software para poder generar futuros mantenimientos e implantaciones sin ningún tipo de dependencia.

## **2.5 Delimitación.**

### **2.5.1 Alcance.**

El alcance de este proyecto estuvo definido por medio de los requerimientos solicitados por parte del desarrollador del video juego, y desarrollado en las cuatro etapas de desarrollo que fueron establecidas para construir el sistema (análisis, diseño, desarrollo y pruebas), dentro de las iteraciones de ejecución. Además, se contempla también todo el proceso de investigaciones, entrevistas y entrega de documentos como lo son: el documento de entrega(tesis), manual técnico, manual de instalación y manual de despliegue. Dentro del alcance no se contará con otro tipo de manuales ni con otro tipo de actividades como los son las capacitaciones, ni el soporte de la aplicación. Además, de lo anterior se deben tener en cuenta las siguientes restricciones de aquellos puntos que se incluirán y, aquellos, que no se incluirán dentro del desarrollo del proyecto.

### **¿Qué incluirá?**

- El desarrollo de los microservicios acordados y establecidos en el documento de requerimientos e historias de usuario.
- El diseño y desarrollo de la base de datos por medio del motor MySQL, y esta se construirá de acuerdo con los requerimientos del sistema.
- La instalación y ambientación del desarrollo en el computador o servidor donde el juego esté funcionando.
- El desarrollo que se hará por medio del lenguaje de programación Java, usando la versión 1.8.
- El servidor, en donde se ejecutarán los microservicios, el cual será Apache Tomcat, versión 7.
- La documentación técnica del sistema como el manual técnico, manual de manual de despliegue y manual de instalación.

### **¿Qué no incluirá?**

- El desarrollo no comprenderá vistas o mockups, o cualquier otro tipo de desarrollo que esté relacionado con el Frontend del video juego.
- No se harán implementaciones dentro del video juego como el análisis y trabajo en su lógica, y el diseño de sus gráficos.

- Para este sistema, no se realizarán desarrollos con otro tipo de lenguaje de programación.
- No incluirá otro tipo de motor de base de datos.
- Esta versión no incluirá un prototipo final, si no un prototipo funcional, ya que al ser un proyecto de grado que se va a trabajar de forma individual, el desarrollo final se puede ver afectado por la falta de recurso humano.

## 2.6 Metodología

De acuerdo con el trabajo de “*Metodologías de Desarrollo de Software*” (2015) de la Pontificia Universidad Católica de Colombia define la metodología como “*Un conjunto de propuestas tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente, los artefactos que se deben producir, y las herramientas y notaciones que se usarán*” lo que significa que nosotros en el momento de utilizar una metodología para desarrollar un proyecto, lo que hacemos es establecer una serie de pasos que están debidamente estructurados y que sirvan como un paradigma para que quienes utilicen la metodología puedan tener control y resultados eficientes en sus procesos.

Por ende, se ha seleccionado Scrum como metodología de desarrollo ya que esta ofrece un marco de trabajo de agilidad para la ejecución de las actividades que se implementarán durante ella, esto quiere decir, que, durante la ejecución del proyecto, se van a estar generando resultados cada vez que finalice una iteración de trabajo y no hasta que finalice todo el proyecto.

A continuación, se hace una breve explicación de las partes y el proceso de la metodología:

## **Metodología Scrum**

Es una metodología de trabajo que sirve para organizar y controlar los procesos de trabajo, en cualquier tipo de industria. Scrum refleja un trabajo como metodología ágil, por lo cual es muy usado hoy en día para los procesos de desarrollo de software ya que también es compatible con la integración continua, devops, la cual, también, se basa en procesos ágiles de desarrollo y despliegue.

En este marco de trabajo se pueden encontrar los siguientes roles, actividades y fases:

### **Roles**

**Product Owner:** Es el representante del cliente, el cual generalmente transmite los requerimientos al equipo, y a su vez informa los avances del desarrollo a los stakeholders.

**Scrummaster:** Este rol es quien organiza las actividades de seguimiento del equipo de desarrollo y sus avances. Dentro de las actividades que se realizan están las reuniones como los dailys o los planning.

**Equipo:** Es el equipo de trabajo que generalmente está conformado por los arquitectos, desarrolladores y testers que participarán en el desarrollo del sistema, o en la solución de incidencias de este.

**Stakeholders:** Son todas aquellas entidades, organizaciones y personas que se encuentran interesadas e involucradas en el desarrollo.

## **Actividades:**

**Sprint:** Es el espacio y tiempo donde se solucionan las historias de usuario que fueron previamente planificadas y distribuidas en la reunión del Planning.

**Backlog:** Es el listado de las tareas e historias de usuario que se establecen al inicio de las iteraciones del spring.

**Historias de usuario:** Documentos mediante los cuales se establecen y distribuyen los requerimientos que previamente se establecieron en el backlog, y que van a servir para desarrollar durante cada sprint.

**Planning:** Se realiza la planeación y distribución de tareas durante el sprint para cada uno de los miembros del equipo de trabajo.

**Dailys:** Reunión diaria, que puede ser versátil, en donde se reúnen el equipo de trabajo (En algunos casos participa el product owner) con el fin de validar los avances e impedimentos que se han tenido durante el sprint entorno al desarrollo.

**Review:** Esta es la reunión que se realiza al final del sprint para revisar el cumplimiento de la solución de las historias de usuario.

**Retrospective:** El equipo de trabajo identifica los puntos que ayudaron al desarrollo de los requerimientos, pero también, se identifican aquellos puntos que pudieron haber retrasado o entorpecido el desarrollo de este.

**Refinamiento:** Reunión donde se verifican y se afinan las historias de usuario que quedaron pendientes durante el sprint.

Durante las iteraciones de este proyecto se podrán ver las siguientes fases:

**Análisis:** Se establecerán los requerimientos a resolver en cada iteración por medio de las historias de usuario, y se planteará de acuerdo con los requerimientos el cómo se puede implementar la arquitectura de microservicios, y la comunicación entre los componentes de dicha arquitectura.

**Diseño:** En esta fase se harán dos tipos de diseño en cada iteración, una que va a estar orientada a diseñar las tablas, campos o data de la base de datos, y la segunda que va a estar orientada a los diseños de los componentes que intervengan en la arquitectura de microservicios como va a ser el caso del diagrama de secuencia, clases y paquetes.

**Desarrollo:** De acuerdo con la arquitectura estipulada, historias de usuario, y el diagrama lógico de secuencia se comenzarán a desarrollar cada uno de los componentes de los microservicios.

**Pruebas Unitarias:** Una vez terminado cada desarrollo se practicarán las pruebas unitarias usando como herramienta de tecnología, JUnit, y Mockito.

**Pruebas funcionales:** Estas se realizarán por parte del desarrollador del video juego de la siguiente manera:

Por parte del desarrollador backend:

- Al terminar cada desarrollo, se realizará un despliegue local del servidor y allí se ejecutarán las funcionalidades correspondientes.
- Se usará la herramienta Soap UI para probar las peticiones y respuestas de los microservicios.

Desarrollador del video juego:

- Se implementarán los endpoints de los microservicios en el video juego y desde allí se probará el flujo completo desde el video juego hasta la base de datos.

**Despliegue:** Cuando se terminen las pruebas unitarias se procederá a desplegar el archivo war, en primera instancia, para hacer las pruebas funcionales, y en segunda, para hacer el paso a producción.

**Paso a producción:** Es el proceso donde el archivo war se desplegará en el servidor de producción, es decir el servidor donde estará alojado el sistema en su versión probada, y el cual estará de cara al usuario final en este caso el video juego.

## **MARCO TEORICO**

### **5.1 Arquitectura en informática**

Es la visión global que se tiene de un sistema informático, en donde se conocen los diferentes componentes de este y la comunicación entre dichos componentes para poder, a partir de esto, realizar un análisis y diseño del mantenimiento, evolución o creación de otro sistema o de alguno existente.

Dentro de la arquitectura en informática se puede encontrar diversas arquitecturas como, por ejemplo: De software como las monolíticas y las arquitecturas de microservicios, o, por otro lado, podemos encontrar arquitecturas de computadores como las MIPS, Harvard, o Von Neumann.

A continuación, se explicarán algunas de ellas:

#### **5.1.1 Arquitectura Harvard**

Es utilizada en los procesadores y tiene como características, diferenciar los datos de las instrucciones ya que ambos son separados en distintas pistas de almacenamiento y son transmitidos a partir de rutas de señal.

#### **5.1.2 Arquitectura Von Neumann**

Esta es una arquitectura en donde las unidades de CPU se comunican con la unidad de memoria por medio de un sistema de buses y allí guardan tanto los datos como las instrucciones.

### **5.1.3 Arquitectura Monolítica**

Esta arquitectura hace referencia a un sistema de información que puede contener uno o varios componentes de software los cuales están supremamente acoplados y todos dependen del funcionamiento de los demás. Si un solo componente llegara a fallar, todo el sistema podría caerse, y así mismo, para poder hacer algún soporte, cambio o implementación en algún componente, este influirá de manera global.

Esto pasa por la estructura del desarrollo en donde todos los componentes de software que trabajan en base a un archivo de configuración que maneja las dependencias de cada uno de ellos y por otro

lado debido al tipo de comunicación de sus componentes por medio de instancias e implementación de métodos, se produce un fuerte acoplamiento, generando un estilo de proyecto que en la medida del tiempo pueda generar problemas en la eficiencia del funcionamiento y en los costos de mantenerlo.

#### **5.1.4 Arquitectura de microservicios**

Es una arquitectura por medio de la cual se desarrollan componentes de software, completamente separados de la capa front que trabajan de manera modular e independiente, cumpliendo con una función específica, y comunicándose con otros componentes, en la mayoría de las veces, por medio de peticiones a través de los protocolos http. Esto genera que su envío de mensajes sea adaptable a cualquier tipo tecnología, sin importar el framework que se utilice, su lenguaje de programación, o el sistema operativo en el que se ejecute.

#### **5.2 Estándares de Calidad de Software de la ISO 9126**

Estos son estándares, que ayudan a desarrollar sistemas de información con buenas prácticas de desarrollo, que puedan minimizar el porcentaje de errores que se puedan presentar en el tiempo de vida de un software.

### 5.2.1 Funcionalidad

Se habla de funcionalidad cuando hay una correcta estructura, entre los atributos de los componentes de software y sus métodos o funciones, la cual se basa en las siguientes características: Adecuación, Exactitud, Interoperabilidad, seguridad y cumplimiento funcional.

- **Adecuación:** En la adecuación, se verifica que los atributos de software estén correctamente relacionados con las diferentes funciones.
- **Exactitud:** Significa que el resultado que muestra la función es el esperado de acuerdo con la relación entre los atributos y las funciones.
- **Interoperabilidad:** Aquí se analizan los atributos de un software y su efectiva interacción con otros componentes.
- **Seguridad:** Se analizan los atributos de un software y su pericia para evitar que otros programas y datos puedan acceder.

### 5.2.2 Fiabilidad

Es la capacidad de un componente de software de funcionar correctamente, bajo ciertas condiciones estipuladas, y durante un tiempo determinado.

Al igual que en la Funcionalidad, para que un programa sea fiable debe cumplir con las siguientes características: Madurez, Recuperabilidad y Tolerancia a fallos.

- **Madurez:** Se analizan los atributos de un software, y su capacidad de responder a fallas.
- **Recuperabilidad:** Se analizan los atributos de un componente de software, y su capacidad para poder recuperar los datos que hayan intervenido en su funcionamiento.
- **Tolerancia a fallos:** Es la posibilidad que tiene un sistema o componente de software para poder seguir con su buen desempeño a pesar de que se presenten fallas.

### 5.2.3 Eficiencia

Se puede describir la eficiencia como el correcto funcionamiento de un sistema cuando se dispongan de cierta cantidad de recursos y su desempeño sea el adecuado.

**Sus características son:** Comportamiento en el tiempo y comportamiento de recursos.

- **Comportamiento en el tiempo:** Se verifica que el sistema responda y procese sus recursos en poco tiempo.
- **Comportamiento de recursos:** Se debe utilizar los recursos necesarios para que el sistema trabaje bien, y en condiciones específicas.

### 5.2.3 Mantenibilidad

Cuando se habla de mantenibilidad, se habla de la capacidad de un componente de software o sistema para poder implementar nuevas funcionalidades, corregir las actuales y poder corregir errores.

**Las características de la mantenibilidad son:** Estabilidad, Facilidad de Análisis, facilidad de cambios y facilidad de pruebas.

- **Estabilidad:** Capacidad del sistema para funcionar y responder en caso de que se cambie algunos de sus componentes.
- **Facilidad de Análisis:** Posibilidad de poder leer y diagnosticar un sistema para poder verificar fallos, corregir errores, o hacer nuevas implementaciones.
- **Facilidad de Cambio:** Capacidad de un sistema o software para permitir hacer cambios o implementaciones.
- **Facilidad de Pruebas:** Capacidad de un software para poder ser probado luego de implementaciones y cambios.

#### 5.2.4 Portabilidad

Es la capacidad que tiene un software para poder ser implementado y probado desde cualquier plataforma.

**Características:** Capacidad de instalación.

- **Capacidad de instalación:** Es aquel sistema o software que cuenta con los atributos necesarios para poder ser instalado en un ambiente específico.

#### 5.3 Bases de datos

Las bases de datos es uno de los elementos más importantes de todos los sistemas que existen el mundo. Hoy en día es muy poco común que se construyan elementos tecnológicos sin pensar en bases de datos de información. Por ejemplo, podemos ver cómo desde los dispositivos electrónicos como los hornos microondas, decodificadores, xbox, hasta aquellos softwares como los sistemas de información web, sistemas con inteligencia artificial, sistemas IOT entre otros, han necesitado para

su funcionamiento de un banco de registros o una base de datos para poder trabajar de forma correcta.

En el caso de este documento se utilizarán bases de datos que están diseñadas para el almacenamiento de datos provenientes de un software, y además para que cierto personal calificado y capacitado pueda ser capaz de instalarlas, configurarlas y administrarlas adecuadamente.

Dentro de las bases de datos, hoy en día, existen dos tipos: SQL Y No SQL.

### **5.3.1 Bases de datos SQL**

Las bases de datos sql son aquellas que utilizan archivos .sql, y son manipuladas y ejecutadas con lenguajes DDL, DML Y DCL.

**Lenguajes DDL:** Por medio de estos lenguajes se definen las estructuras de los datos, y se utilizan sentencias como: Alter, Create y Drop.

**Lenguajes DML:** Con estas sentencias se manipulan directamente los datos, como lo son: SELECT, INSERT, UPDATE y DELETE.

**Lenguajes DCL:** Se usan para manipular las funciones de accesos a los componentes a las bases de datos. Podemos encontrar GRANT y REVOKE.

## **5.4 Gamificación**

La gamificación es una técnica de aprendizaje para los estudiantes por medio de la cual se propone el juego como un medio de aprendizaje, utilizando la motivación, la recompensa y la experiencia para que el estudiante tenga como resultado un resultado significativo y funcional. Esto quiere decir que quién va a trabajar bajo esta técnica puede tomar cualquier tema y aplicarla por medio de un juego, a esto se le puede llamar gamificación o juego serio. (Juego serio es cuando en un concepto académico o laboral se aplica la gamificación para tener algún tipo de resultado).

La gamificación apoya el concepto de la motivación y promueve la competencia, esto lo hace a partir de retos, diversión, y aprovechamiento del tiempo libre. El aprendizaje está ligado a la motivación, entonces la idea es utilizar la gamificación como potenciador para que se mejore el aprendizaje.

## DESARROLLO DEL PROYECTO

### Fase de levantamiento de información y diseño Funcional

En la etapa inicial del desarrollo del proyecto, se realiza una fase del Análisis y Diseño funcional en donde se entrevista al desarrollador del video juego para conocer los requerimientos y alcance que tendrá el video juego.

Por medio de estos requerimientos se estableció con el estudiante backlog con el que se iban a construir las historias de usuario, y las cuales se iban a desarrollar en cada sprint. Se realizaron dos reuniones no solo para definir de forma correcta los requerimientos, sino también los alcances.

**Tabla 1**

Por medio de esta tabla establece el backlog del proyecto

id	Descripción de los requerimientos	Product Owner	Equipo de desarrollo	Entidad
1	Arquitectura y definición de la base de datos	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
2	Instalación y configuración de ambientes, y programas, y creación del proyecto	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
3	Creación del microservicio para el registro de los estudiantes	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
4	Creación del microservicio para la validación de credenciales	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
5	creación de microservicio para generación del token	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
6	Creación del microservicio para la consulta de estudiantes por correo electrónico	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
7	Creación de microservicio para la consulta de estudiantes por puntaje	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano
8	Creación de microservicio para la encriptación de contraseñas y desencriptación de hash	Estudiante de Maestría	Carlos Sanz	Politécnico Grancolombiano

## **Proceso de Preparación de ambientes**

Antes de comenzar el desarrollo es muy importante que nuestro proyecto esté debidamente instalado y configurado con las siguientes herramientas y tecnologías adecuadas:

**Java 8:** Se va a utilizar la versión 1.8 de JDK ya que es una de las versiones más completas y sencillas, y además, con implementaciones muy funcionales como es el caso de las expresiones lambdas las cuales nos ayudan a reducir código y nos ayudan a pasar parámetros de una expresión normal a una de orden superior, o, por ejemplo, están las fechas que tienen un tratamiento diferente, a las versiones anteriores de java, el cual trae un nuevo formato y métodos que hacen del desarrollo de las fechas un tema más sencillo.

**Springtool Suite:** Es un conjunto de herramientas basadas en el framework Spring, las cuales son empaquetadas para poder realizar desarrollos orientados a la web. En este caso se ha escogido como framework de desarrollo, ya que cuenta con un conjunto de librerías muy útiles, no solo para el desarrollo de los microservicios, sino que además cuenta con librerías para otro tipo de procesos que se atravesarán durante el camino, como la conexión y la persistencia a la base de datos, acceso a herramientas para la seguridad de la información entre otras.

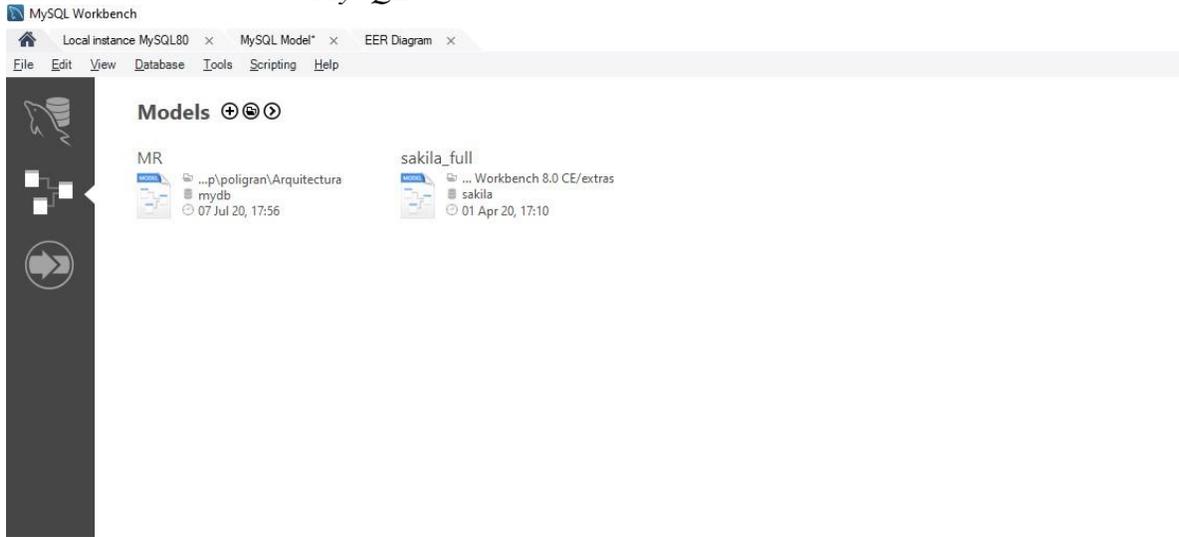
**Soap ui y Postman:** Estas son dos aplicaciones que se escogieron porque dan la posibilidad de realizar pruebas de entradas y salidas de los servicios que vayamos a construir.

**MySQL:** Se ha seleccionado MySQL como se ve en la **Figura 1**, como motor de base de datos de este sistema ya que es muy rápida y sencilla de manejar, cuenta con diferentes capas de seguridad, cumple con una condición de interoperabilidad ya que se puede usar con Windows, Linux u otros sistemas operativos, sin nombrar que es gratuita es que es gratuita.

Antes de comenzar cualquier tipo de desarrollo, es muy importante definir cómo se va a trabajar la base de datos. En este caso se va a trabajar con MySQL, donde inicialmente, de acuerdo con los requerimientos no funcionales por parte del video juego, necesitaremos una base de datos fácil de usar, segura y de buen rendimiento. Esto nos ofrece MySQL la cual nos permite no solo crear una base de datos, sino que, además, podemos realizar el diseño de esta. Este diseño tiene como nombre “Modelo Relacional”.

**Figura 1**

*Motor de base de datos MySQL.*



Nota: Imagen de captura al motor de base de datos MySQL con sus instancias. (Elaboración Propia)

**Figura 2**

*Historia 1*

**1 DEFINICIÓN DE HISTORIAS DE USUARIO**

**Identificador de funcionalidad:** H1 – Arquitectura y definición de base de datos



<b>Sprint:</b> # 1	<b>Puntos de historia:</b>
--------------------	----------------------------

**Funcionalidad:**

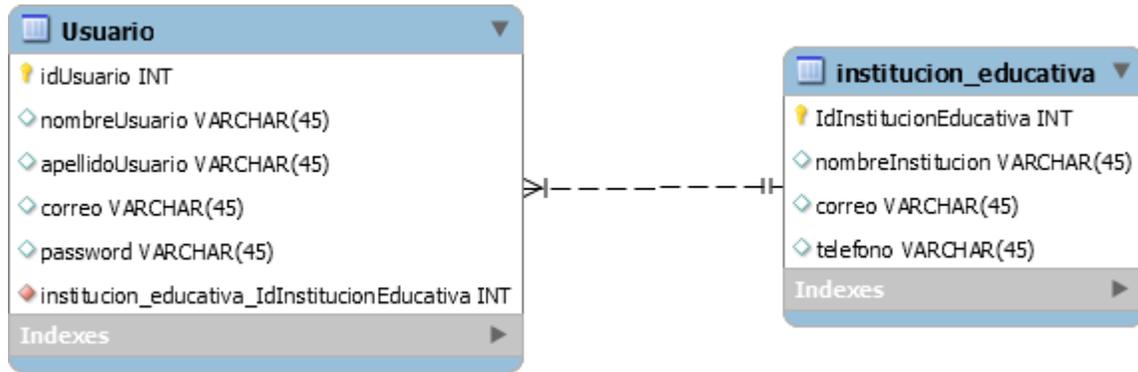
<b>Cómo</b>	Creación de base de datos
<b>Quiero</b>	Que exista una base de datos donde guardar y consultar la información del video juego
<b>Para</b>	guardar y consultar la información del video juego

**Criterios de Aceptación**

Contexto
<p>Título: Criterios de aceptación</p> <ul style="list-style-type: none"> <li>• La base de datos debe contemplar las tablas de usuario, score, institución educativa, tipo documento, ciudad y país.</li> <li>• Debe ser una base de datos normalizada</li> <li>• Debe tener un modelo relacional.</li> </ul>

**Figura 3**

*Modelo relacional inicial*



Nota: Modelo relacional inicial con el cual se suplían los requerimientos iniciales del sistema.

(Elaboración Propia)

De acuerdo con la primera historia del sprint **Figura 2**, se realizó el diseño de la tabla Usuario, donde se van a guardar los datos de los usuarios (“nombreUsuario”, “apellidoUsuario”, “la institución educativa”) incluyendo las credenciales (“correo”, y “password”) como se puede ver en **Figura 3**.

De esta tabla se van a administrar los datos de los usuarios, por lo tanto, el sistema no solo deberá registrar los datos, sino que también deberá permitir la consulta de las credenciales de cada usuario para hacer la validación respectiva del acceso al sistema. Además, también se diseñó la

tabla “Institución Educativa” que estará relacionada con la tabla usuario, la cual va a servir para almacenar y consultar los datos de las instituciones educativas que estén interesadas en contratar el video juego.

Figura 4

Historia 2 (Elaboración Propia)

1 DEFINICIÓN DE HISTORIAS DE USUARIO

Identificador de funcionalidad: H2 – Instalación, creación de ambientes y proyecto

Sprint: # 2	Puntos de historia:
-------------	---------------------

Funcionalidad:

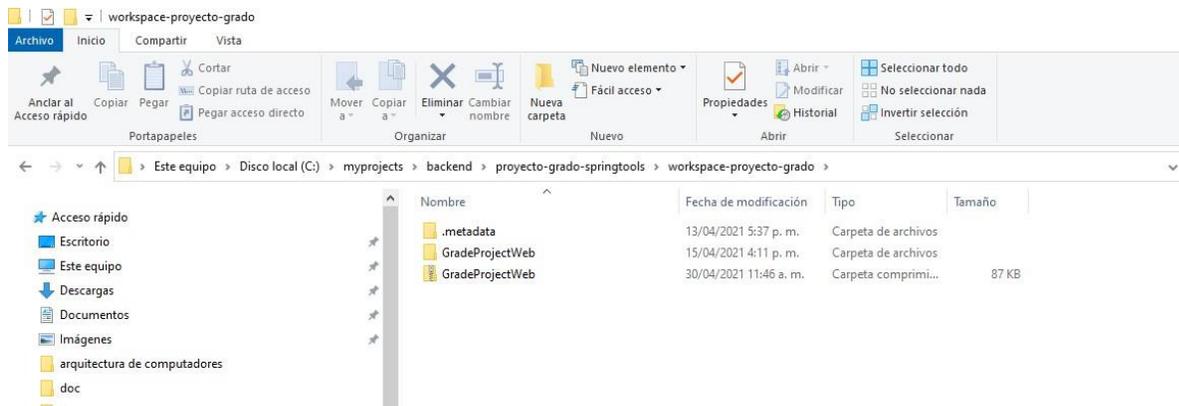
Cómo	Instalación de herramientas, dependencias y programas para la construcción del <u>backend</u> y creación del proyecto
Quiero	Herramientas de ambientes debidamente instalados y documentados
Para	Desarrollar <u>backend</u> del video juego

Criterios de Aceptación

Contexto
Título: Criterios de aceptación <ul style="list-style-type: none"><li>Evidenciar en computador local para mostrar el ambiente y creación del proyecto</li><li>Manual de instalación.</li></ul>

Figura 5

Ambientes y Workspace (Elaboración Propia)

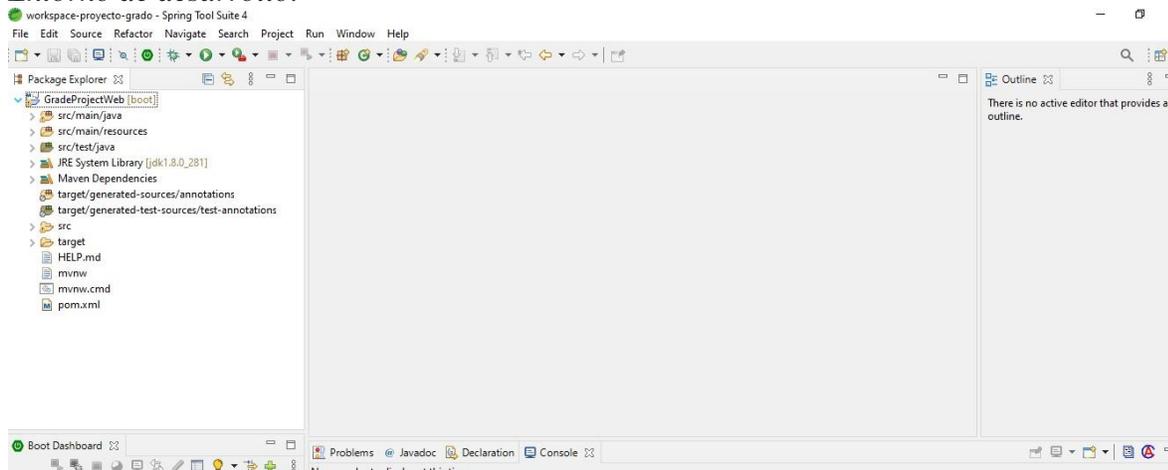


Nota: Creación de ambientes y workspace para el desarrollo del proyecto.

Después de crear las tablas en la base de datos, de acuerdo a la historia 2, **Figura 4**, se creó el proyecto, donde se van configuran los componentes necesarios para el avance del desarrollo como, por ejemplo, el workspace, **Figura 5**, el datasource para el conexión a la base de datos y las dependencias necesarias para trabajar con la arquitectura de servicios Rest. En el proyecto se crearán los microservicios que tendrán diferentes capas de desarrollo con el Framework de Springtool el cual provee de una serie de herramientas para facilitar el desarrollo de los microservicios como anotaciones, librerías y plugin.

**Figura 6**

*Entorno de desarrollo.*



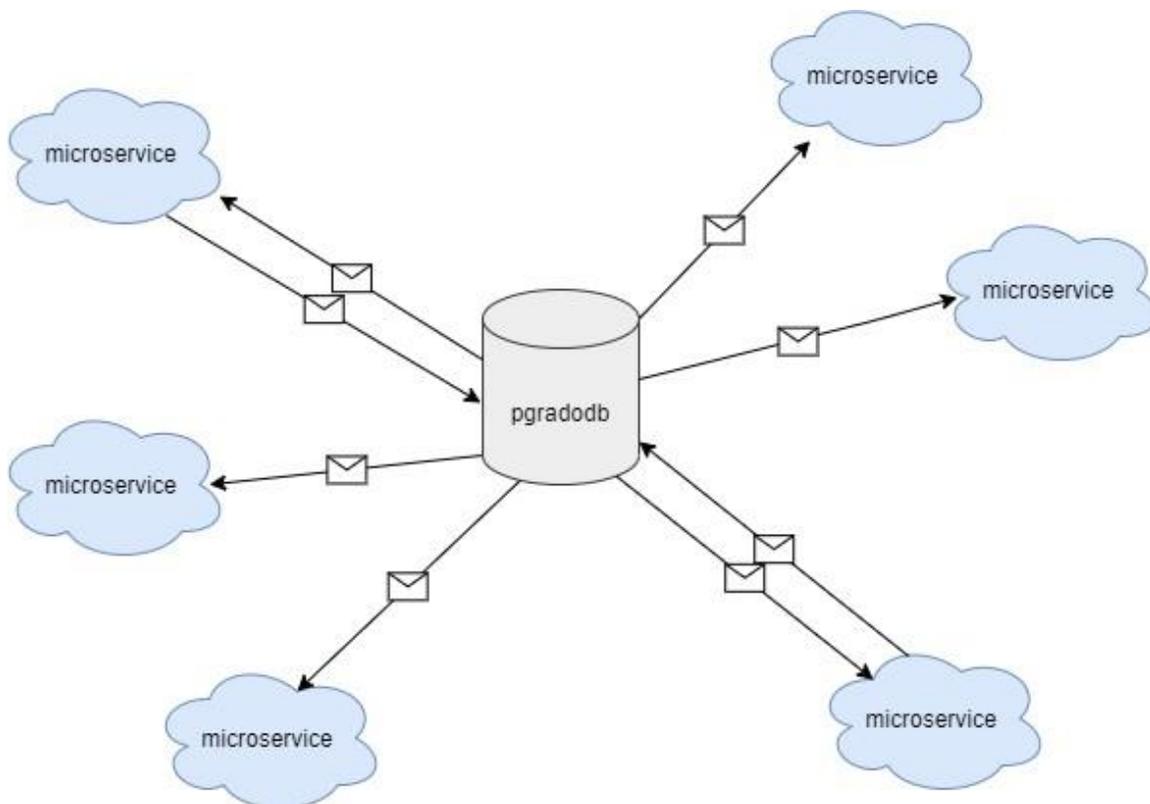
Nota: Se realiza configuración del eclipse para realizar el desarrollo de los microservicios.

(Elaboración Propia)

Luego de realizar la configuración de ambientes locales **Figura 6** y estructurar la base de datos se procede a crear y desarrollar cada uno de los componentes modulares dentro de un mismo workspace asignando sus respectivos archivos de configuración y despliegue. A estos componentes modulares serán los denominados microservicios **Figura 7**, y su arquitectura nos irá permitiendo que en la medida que el sistema vaya necesitando nuevas funcionalidades se puedan ir agregando sin alterar las otras existentes y sin afectar los ambientes actuales.

**Figura 7**

*Diseño global de cómo es el flujo de información de los microservicios.*



Nota: En este modelo se esboza el flujo de mensajería de información entre la base de datos y los microservicios por medio de peticiones http. (Elaboración Propia)

En la imagen anterior se refleja el flujo de información que van a tener los microservicios con la base de datos mediante peticiones asíncronas por medio de las cuales se va a consultar, registrar, actualizar y eliminar la información relevante del video juego.

A continuación, se presenta un diseño global del flujo de información del sistema backend mediante microservicios en comunicación con la base de datos:

## Figura 8

Historia 3. (Elaboración Propia)

### 1 DEFINICIÓN DE HISTORIAS DE USUARIO

**Identificador de funcionalidad:** H5 – Creación de microservicio que genere un Json Web Token(JWT)



<b>Sprint:</b> # 5	<b>Puntos de historia:</b>
--------------------	----------------------------

#### Funcionalidad:

<b>Cómo</b>	Creación de microservicio que genere un <u>Json Web Token</u>
<b>Quiero</b>	Un microservicio que permita la creación de un Token para la seguridad del video juego
<b>Para</b>	La seguridad de los usuarios

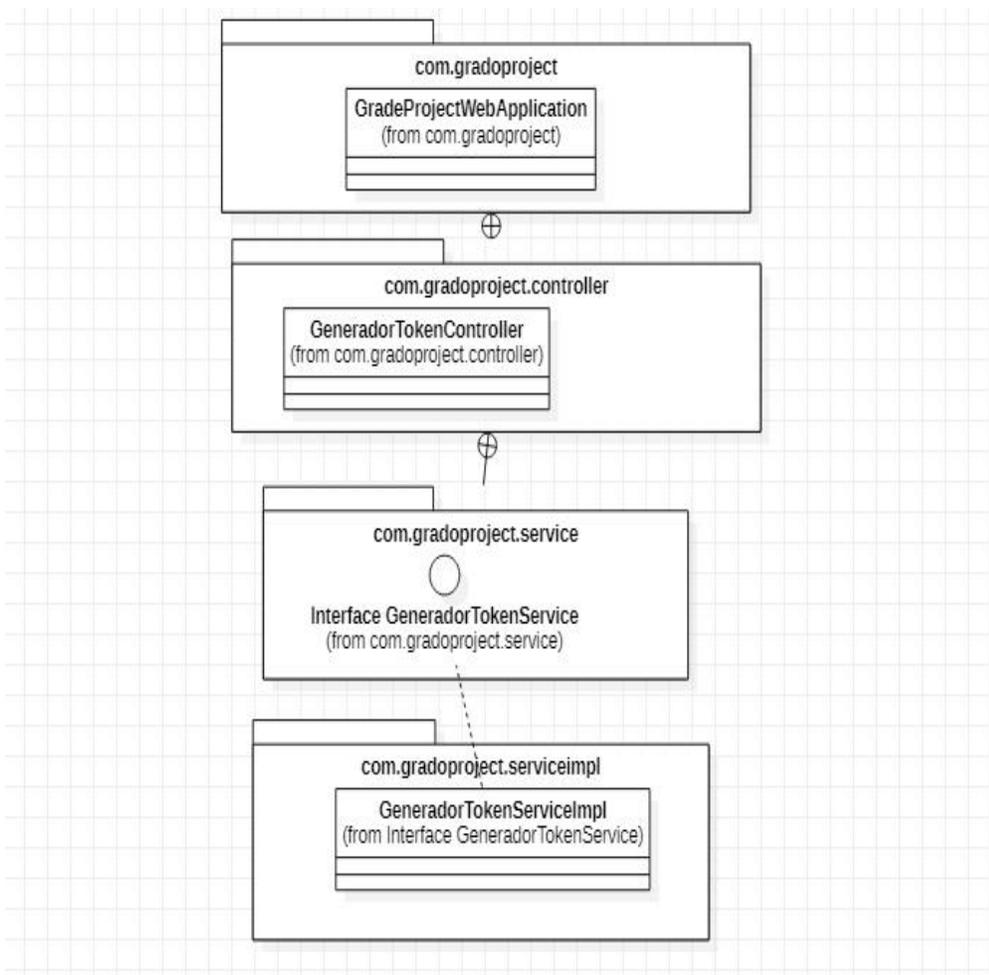
#### Criterios de Aceptación

Contexto
Título: Criterios de aceptación <ul style="list-style-type: none"><li>El microservicio debe generar el <u>Json Web Token</u>.</li></ul>

En el siguiente modelo se exponen cada uno de los paquetes que harán parte de la arquitectura

**Figura 9**

*Paquetes Generador Token. (Elaboración Propia)*

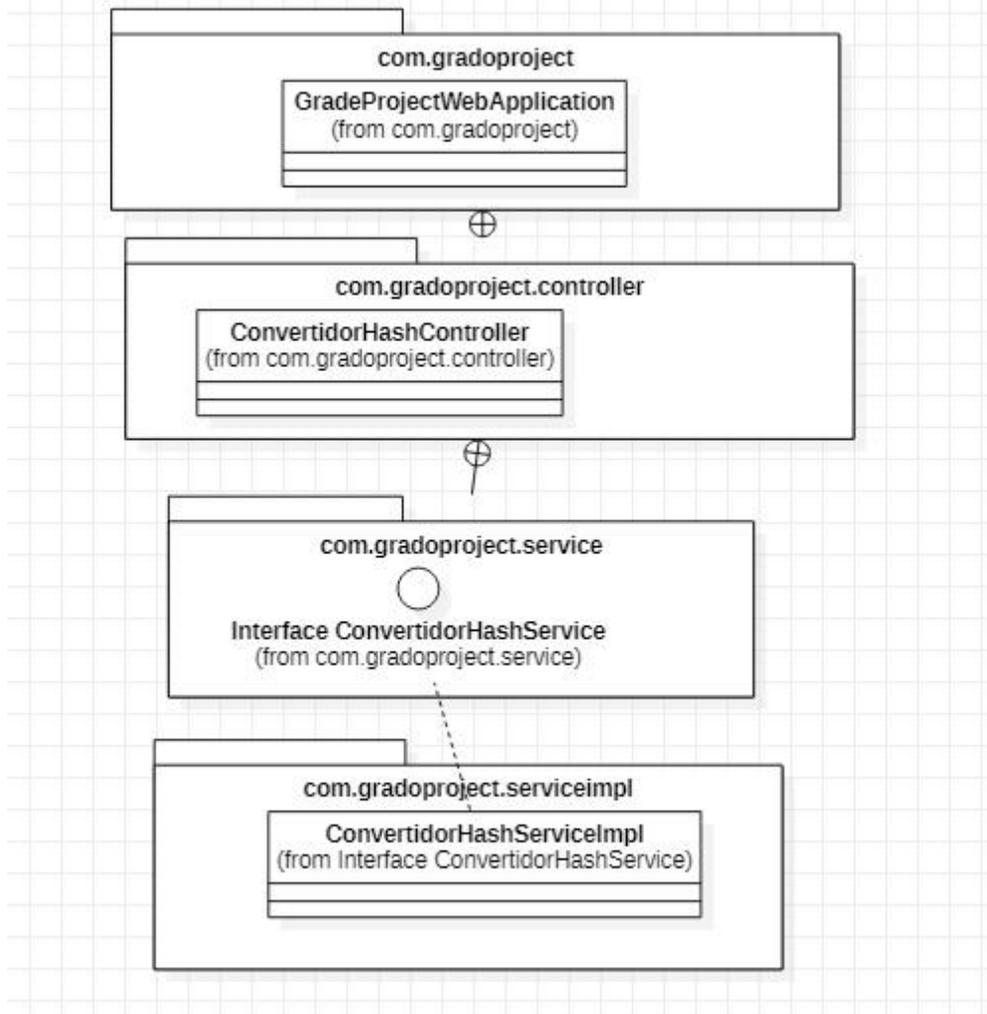


En la historia **Figura 8** se solicitaba crear un servicio que manejara los tokens del sistema, por lo cual se desarrolló un generador del token, el cual es un componente que encripta los datos del json para generar mayor seguridad en la información que se está enviando. En los paquetes, **Figura 9**, se establecen diferentes tipos de capas dentro de la aplicación: capa controller que sirve como interfaz del microservicio; capa service que maneja la lógica de negocio, y la capa repository donde se maneja la persistencia a la base de datos.

Así mismo, se pueden evidenciar en los siguientes diagramas de ejemplo.

**Figura 10**

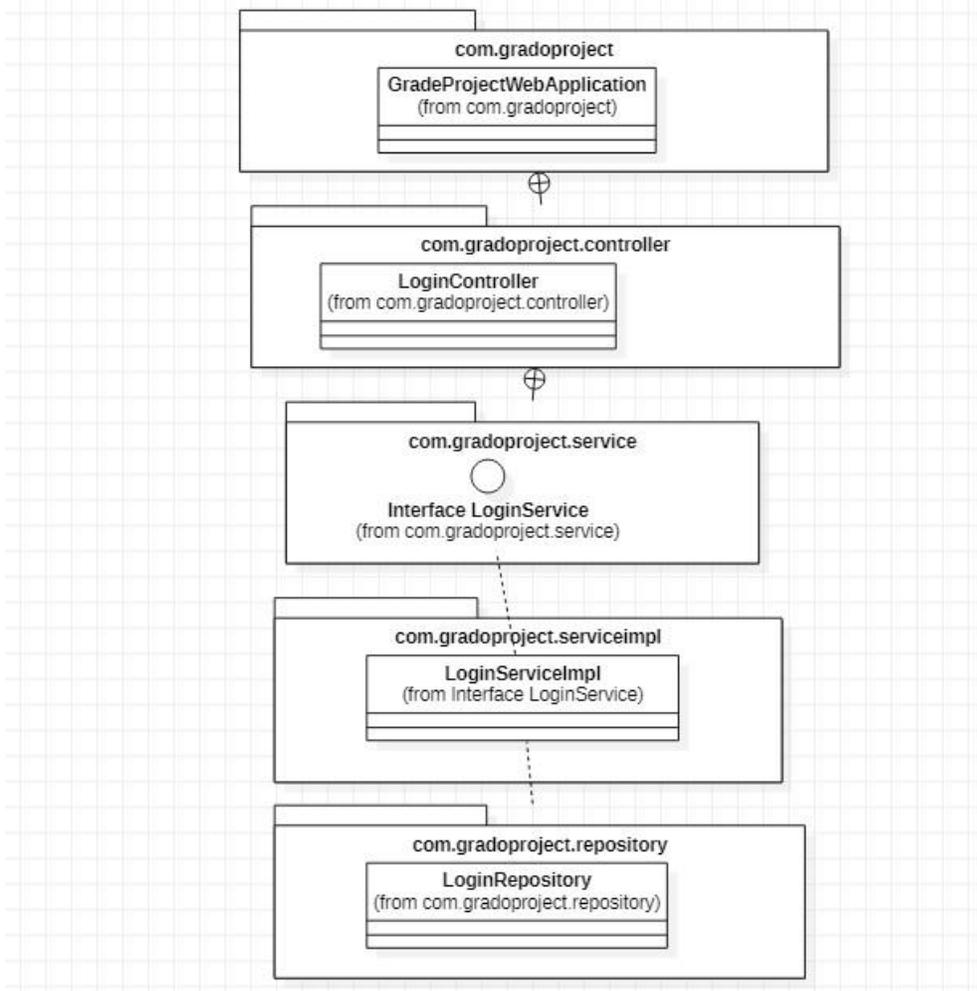
*Paquetes Convertidor Hash. (Elaboración Propia)*



En el convertidor hash se hace un proceso de encriptación de las password del usuario cuando se va a guardar en base de datos, **Figura 10**, y uno, de desencriptación cuando se realiza alguna consulta de las credenciales en la base de datos.

**Figura 11**

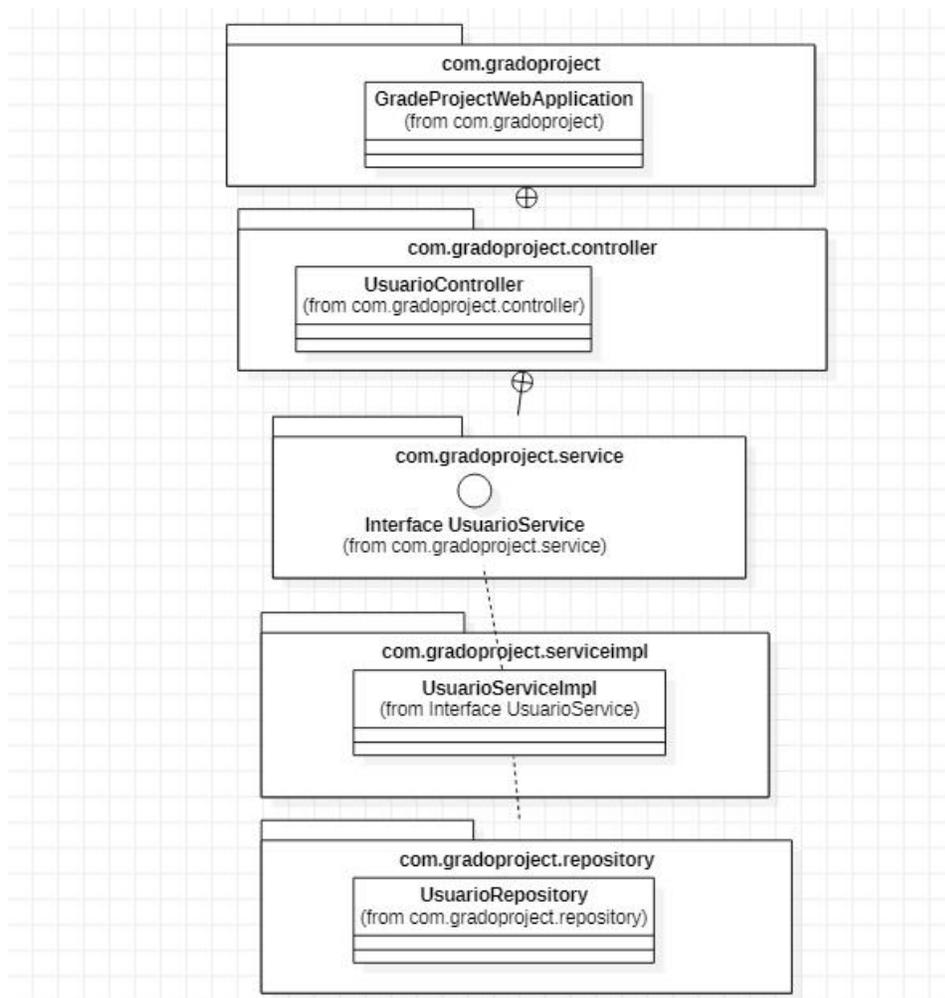
*Paquetes Login.* (Elaboración Propia)



En el componente del login se realiza la validación de credenciales. Para esta validación fue importante, no solo validar las credenciales del usuario frente a la base de datos, sino que también fue importante realizar la implementación del token, **Figura 11**.

**Figura 12**

*Paquetes Usuario.* (Elaboración Propia)



En el componente del usuario, en el paquete repository se llaman a los queries de consultas de usuarios por correo, edición y registro de usuarios, **Figura 12**.

**Figura 13**

*Respuesta del microservicio consulta de roles. (Elaboración Propia)*

**1 DEFINICIÓN DE HISTORIAS DE USUARIO**

**Identificador de funcionalidad:** H3 – Creación de microservicio para el registro de usuario



<b>Sprint:</b> # 3	<b>Puntos de historia:</b>
--------------------	----------------------------

**Funcionalidad:**

<b>Cómo</b>	Creación de registro del usuario
<b>Quiero</b>	Un microservicio con una operación que permita registrar a un usuario
<b>Para</b>	Registrar al usuario del video juego



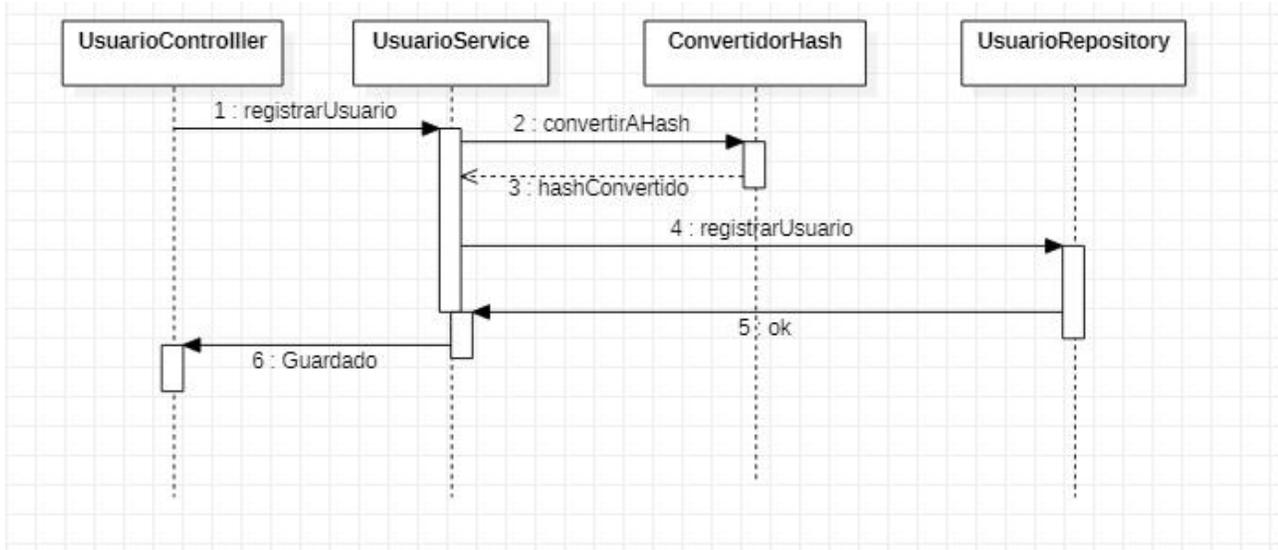
**Criterios de Aceptación**

Contexto
Título: Criterios de aceptación <ul style="list-style-type: none"><li>• El microservicio debe poderse consumir desde el video juego</li><li>• El microservicio debe permitir guardar al usuario en base de datos</li></ul>

De acuerdo con la historia 3, **Figura 13**, se propuso realizar un componente de software para poder permitirle al front realizar el registro de los usuarios

**Figura 14**

*Secuencia registro usuario* (Elaboración Propia).



En el diagrama anterior, **Figura 14**, se puede visualizar la secuencia lógica y el flujo de información en la operación del registro de un usuario, en donde el controlador hace la petición al componente service, el componente service recibe la petición y realiza alguna validación de seguridad con el hash, y, por último, el componente repository donde se realizan las peticiones a las base de datos.

## Figura 15

Historia 4. (Elaboración Propia)

### 1 DEFINICIÓN DE HISTORIAS DE USUARIO

**Identificador de funcionalidad:** H4 – Creación de microservicio para la validación de credenciales



<b>Sprint:</b> # 4	<b>Puntos de historia:</b>
--------------------	----------------------------

**Funcionalidad:**

<b>Cómo</b>	Crear microservicio que valide si un usuario está en base de datos
<b>Quiero</b>	Un microservicio que me responda si un usuario existe o no
<b>Para</b>	Enviar al usuario al video juego



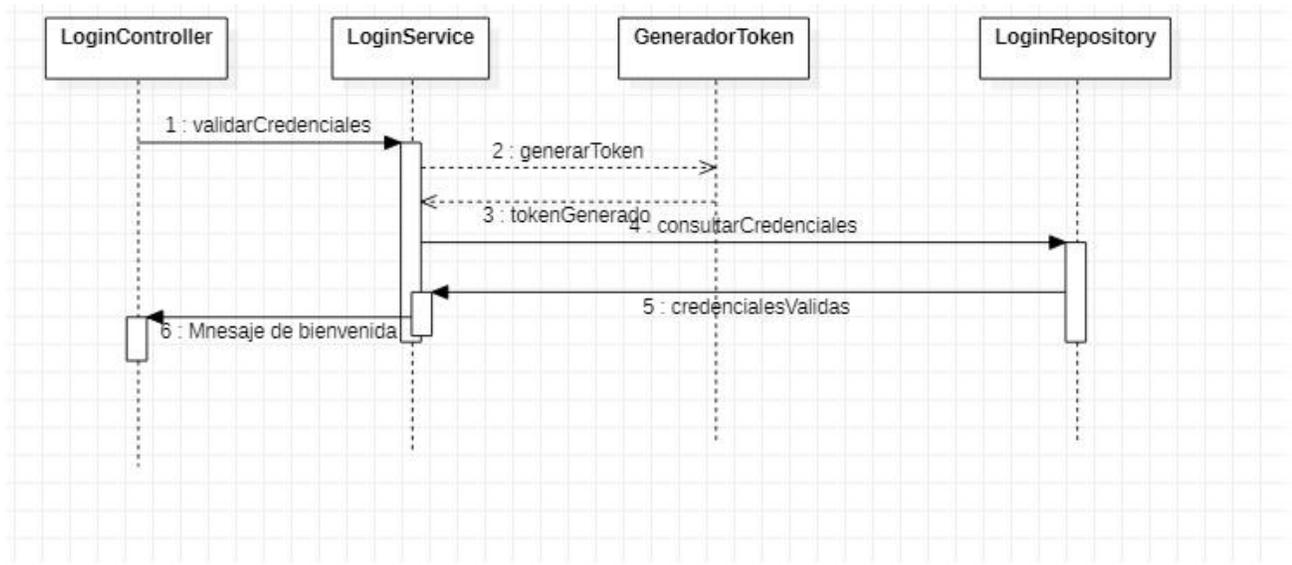
**Criterios de Aceptación**

Contexto
Título: Criterios de aceptación <ul style="list-style-type: none"><li>• El microservicio debe responder un mensaje: "Las credenciales son correctas"</li><li>• El microservicio, en caso de que no exista, debe responder un mensaje: "Las credenciales son incorrectas"</li></ul>

En la historia 4, **Figura 15**, se vió la necesidad de crear el componente de para poder hacer las validaciones de los estudiantes que realizaran previamente el registro del video juego.

**Figura 16**

*Secuencia validación de credenciales.* (Elaboración Propia)



En la validación de credenciales, **Figura 16**, se realiza el mismo flujo de la información que se presenta en la secuencia del registro (controller, service y repository) pero en este caso con la excepción de que no se hace la validación con el componente hash, si no que en este caso se hace con el componente que genera el token, para poder encriptar la información del cuerpo del Json.

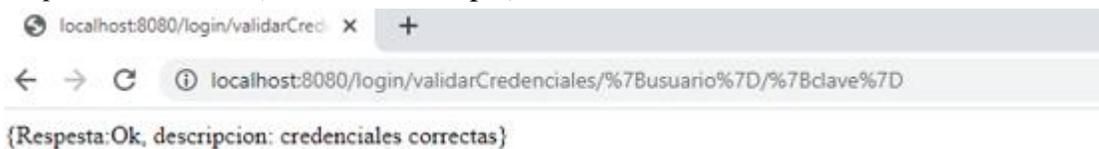
## CODIGO DEL SISTEMA

Para el desarrollo del sistema las primeras funcionalidades que se implementaron por medio de la arquitectura de microservicios fueron: el registro de los estudiantes, y validación de credenciales como se ven en las imágenes: **Figura 17** y **Figura 18** .

Imagen de la capa del logín de la capa del video juego.

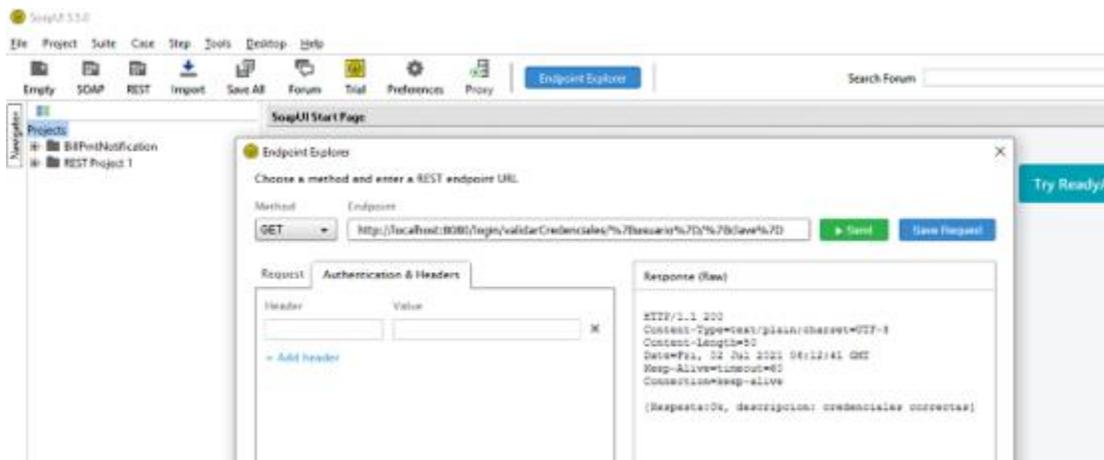
*Figura 17*

*Respuesta servicio. (Elaboración Propia)*



*Figura 18*

*Respuesta en SOAP UI del servicio. (Elaboración Propia)*



En la anterior imagen se valida la respuesta de validación de credenciales donde se presenta la respuesta de “credenciales correctas”.

**Figura 19**

Interface controller en código Java. (Elaboración Propia)

```
@GetMapping(path="/validarCredenciales/{usuario}/{clave}")
public String validarCredenciales(@PathVariable String usuario,@PathVariable String clave) {}
return service.validarCredenciales(usuario,clave);
```

**Figura 20**

Capa Service en código Java. (Elaboración Propia)

```
1 package com.gradoproject.service.impl;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5
6
7
8
9
10 @Service
11 public class LoginServiceImpl implements LoginService {
12
13     @Autowired
14     private UsuarioRepository usuarioRepository;
15
16     @Override
17     public String validarCredenciales(String usuario,String clave) {
18         // TODO Auto-generated method stub
19         String respuesta = "";
20         int respuestaDb = usuarioRepository.validarUsuarioContrasenia(usuario,clave);
21         System.out.println("respuesta query cuantos correos----->"+resp
22             if (respuestaDb==1) {
23                 respuesta="{Respuesta:Ok, descripcion: credenciales correctas}";
24             }else {
25                 respuesta="{Respuesta:400, descripcion: credenciales Incorrectas}";
26             }
27         return respuesta;
28     }
29 }
30 }
31
```

En el código (**Figura 19, Figura 20 y Figura 21**) previo se evidencia la validación lógica que se realiza confrontando la base de datos para saber si el usuario que se está autenticando es el correcto.

**Figura 21**

Validación de credenciales con mensajes de salida. (Elaboración Propia)

```
public int validarUsuarioContrasenia(String usuario,String clave) {
    String respuesta="";
    String myQuery = CONSULTA_CREDENCIALES+usuario+"AND clave="+clave;
    System.out.println("consulta armada "+myQuery);
    int h=query.queryForObject(myQuery, Integer.class);
    System.out.println("respuesta query cuantos correos----->"+h);
    if (h==1) {
        respuesta="{Respuesta:Ok, descripcion: credenciales correctas}";
    }else {
        respuesta="{Respuesta:400, descripcion: credenciales Incorrectas}";
    }
    return h;
}
```

En el siguiente sprint, en la historia 6, **Figura 22**, se realizó la consulta para obtener los estados los estados del usuario, de su fecha de registro y avances del proyecto, para esto fue necesario crear una tabla nueva intermedia donde se asociará el usuario registrado y la fecha de registro. Se creó otra tabla también, para registrar y consultar la fecha de inicio de sesión y de salida del estudiante.

**Figura 22**

*Historia 6.* (Elaboración Propia)

**1 DEFINICIÓN DE HISTORIAS DE USUARIO**

**Identificador de funcionalidad:** H6 – Consulta de usuarios por correos

<b>Sprint:</b> # 6	<b>Puntos de historia:</b>
--------------------	----------------------------

**Funcionalidad:**

<b>Cómo</b>	Creación de microservicio con la consulta de datos
<b>Quiero</b>	Un microservicio que permita consultar los correos de los usuarios
<b>Para</b>	Poder consultar que los usuarios se estén registrando con <u>dominio</u> de las instituciones educativas

**Criterios de Aceptación**

<b>Contexto</b>
<p>Título: Criterios de aceptación</p> <ul style="list-style-type: none"> <li>• El microservicio debe generar una lista de correos</li> </ul>

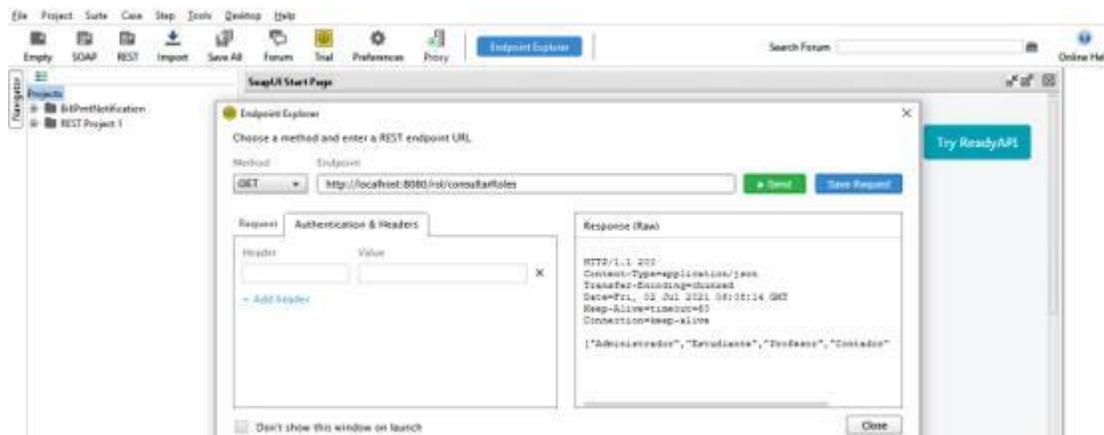
**Figura 23**

Respuesta del microservicio consulta de roles. (Elaboración Propia)



**Figura 24**

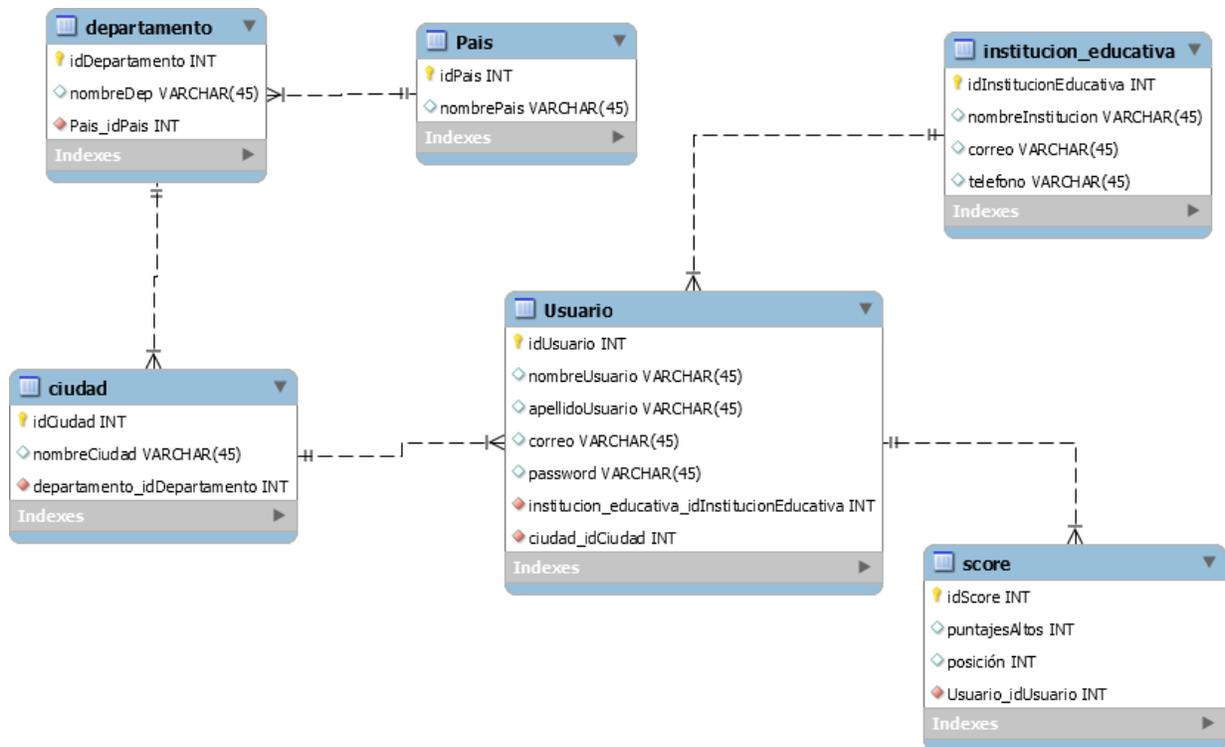
Ambientes y Workspace. (Elaboración Propia)



Además, el docente también tendrá un rol, de acuerdo la consulta de roles que se haga (**Figura 23** y **Figura 24**), único donde podrá monitorear a los estudiantes por medio de la información que se consulte en este requerimiento. Por ejemplo, el registro y la fecha de logeo le va a servir al docente para llevar un control diario para saber si el estudiante está estudiando y practicando los ejercicios y temas de álgebra por medio del video juego, que avances tiene, cual es la constancia del estudiante, y que temas son los que le pueden causarle problemas en el aprendizaje. El avance se va a medir por medio de estrellas las cuales se darán de acuerdo con los aciertos que tenga el estudiante cada vez que resuelva los ejercicios correctamente por medio de la preparación de recetas.

**Figure 26**

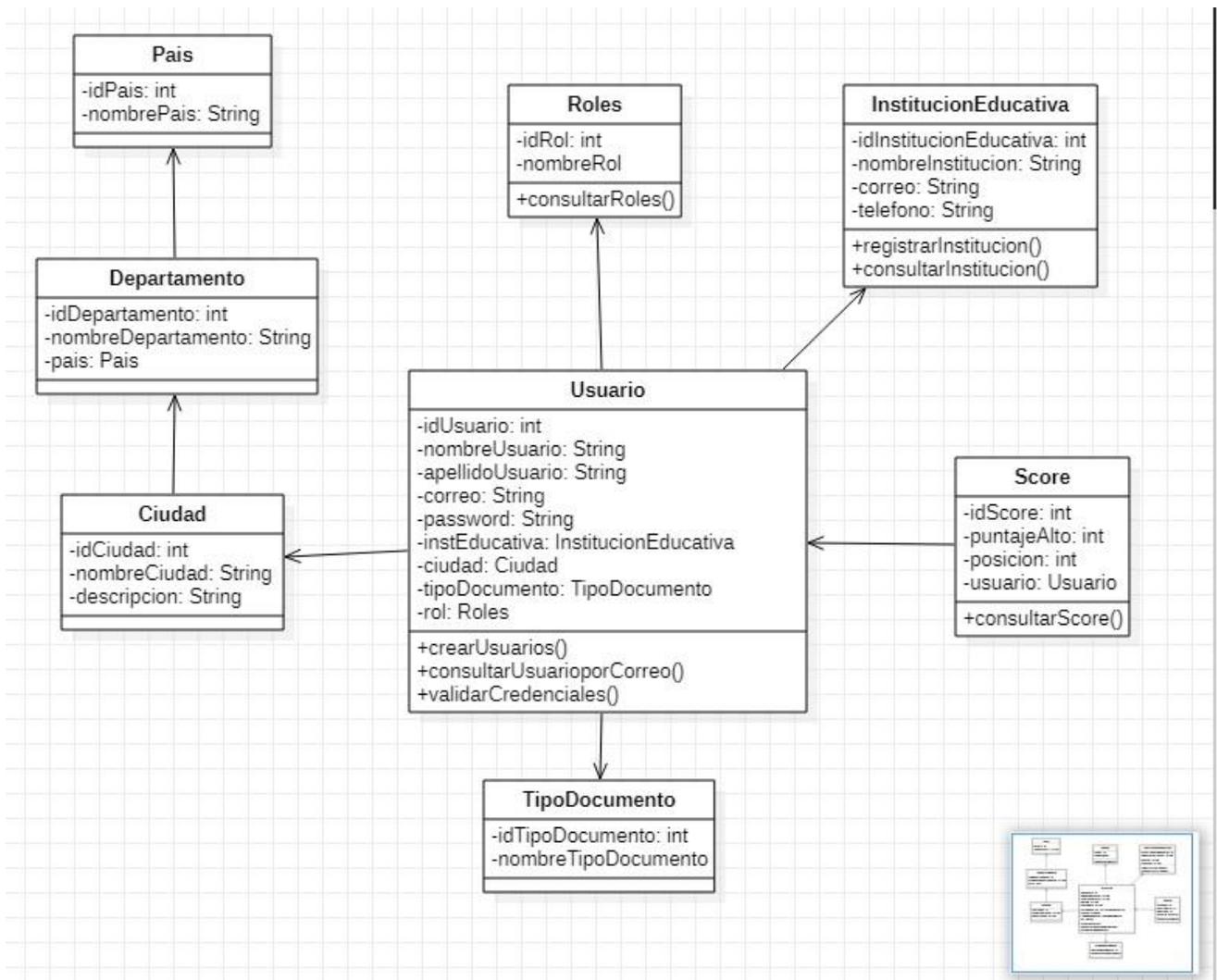
*Modelo relacional global. (Elaboración Propia)*



Como se puede evidenciar en la **Figura 24**, de acuerdo al avance que se tuvo en la creación de tablas de la base de datos, se obtuvo un modelo relacional casi completo al cual se le iban a hacer ciertas modificaciones mas adelante. Por buenas practicas de desarrollo y diseño también se elaboró un diagrama de clases, **Figura 25**, realizando el mapeo de la base de datos, por medio de clases denominadas Dtos, las cuales, en el lenguaje Java, tienen los mismos atributos, pero, además, se les crean unos métodos, que se necesitan para su dominio lógico.

Figura 25

Clases del sistema. (Elaboración Propia)



En el diagrama de clases se valida la relación, el grado de acoplamiento y las operaciones que van a tener cada una de las clases, las cuales se establecen en base al modelo anterior del sistema. Además, en las siguientes imágenes se muestra cada una de ellas en el código(Figura 26, Figura 27, Figura 28 y Figura 29).

**Figura 26**

*Dto Rol.* (Elaboración Propia)

```
1 package com.gradoproject.dto;
2
3 public class Roles {
4
5     private int idRol;
6     private String nombreRol;
7
8     public int getIdRol() {
9         return idRol;
10    }
11    public void setIdRol(int idRol) {
12        this.idRol = idRol;
13    }
14    public String getNombreRol() {
15        return nombreRol;
16    }
17    public void setNombreRol(String nombreRol) {
18        this.nombreRol = nombreRol;
19    }
20
21 }
22
23 }
```

Figura 27

Clase Dto Score. (Elaboración Propia)

```
1 package com.gradoproject.dto;
2
3 public class Score {
4
5     private int idScore;
6     private int puntajeAlto;
7     private int posicion;
8     private Usuario usuario;
9
10    public int getIdScore() {
11        return idScore;
12    }
13    public void setIdScore(int idScore) {
14        this.idScore = idScore;
15    }
16    public int getPuntajeAlto() {
17        return puntajeAlto;
18    }
19    public void setPuntajeAlto(int puntajeAlto) {
20        this.puntajeAlto = puntajeAlto;
21    }
22    public int getPosicion() {
23        return posicion;
24    }
25    public void setPosicion(int posicion) {
26        this.posicion = posicion;
27    }
28    public Usuario getUsuario() {
29        return usuario;
30    }
31    public void setUsuario(Usuario usuario) {
32        this.usuario = usuario;
33    }
34
35
36
37 }
38
```

Figura 28

Clase Dto Usuario. (Elaboración Propia)

```
1 package com.gradoproject.dto;
2
3 public class Usuario {
4
5     private int idUsuario;
6     private String nombreUsuario;
7     private String apellidoUsuario;
8     private String correo;
9     private String password;
10    private InstitucionEducativa instEducativa;
11    private Ciudad ciudad;
12    private TipoDocumento TipoDocumento;
13    private Roles rol;
14
15    public int getIdUsuario() {
16        return idUsuario;
17    }
18    public void setIdUsuario(int idUsuario) {
19        this.idUsuario = idUsuario;
20    }
21    public String getNombreUsuario() {
22        return nombreUsuario;
23    }
24    public void setNombreUsuario(String nombreUsuario) {
25        this.nombreUsuario = nombreUsuario;
26    }
27    public String getApellidoUsuario() {
28        return apellidoUsuario;
29    }
30    public void setApellidoUsuario(String apellidoUsuario) {
31        this.apellidoUsuario = apellidoUsuario;
32    }
33    public String getCorreo() {
34        return correo;
35    }
36    public void setCorreo(String correo) {
37        this.correo = correo;
38    }
39    ... ..
40 }
```

**Figura 29**

*Clase Dto Institución Educativa.* (Elaboración Propia)

```
1 package com.gradoproject.dto;
2
3 public class InstitucionEducativa {
4
5     private int idInstitucionEducativa;
6     private String nombreInstitucion;
7     private String correo;
8     private String telefono;
9
10    public int getIdInstitucionEducativa() {
11        return idInstitucionEducativa;
12    }
13    public void setIdInstitucionEducativa(int idInstitucionEducativa) {
14        this.idInstitucionEducativa = idInstitucionEducativa;
15    }
16    public String getNombreInstitucion() {
17        return nombreInstitucion;
18    }
19    public void setNombreInstitucion(String nombreInstitucion) {
20        this.nombreInstitucion = nombreInstitucion;
21    }
22    public String getCorreo() {
23        return correo;
24    }
25    public void setCorreo(String correo) {
26        this.correo = correo;
27    }
28    public String getTelefono() {
29        return telefono;
30    }
31    public void setTelefono(String telefono) {
32        this.telefono = telefono;
33    }
34 }
35 }
```

En el siguiente sprint, revisando el backlog y de acuerdo con la conversación con el estudiante que está desarrollando el video juego, se acordó que se iba a desarrollar la historia de usuario referente a visualizar el listado de los estudiantes ordenados de mayor a menor puntaje, resaltando los cinco mejores que encabezaran un podio. Este listado no solo debe ser visualizado, sino que, además, el sistema debía permitirle al usuario exportarlo como un Excel.

En el siguiente sprint se analizó que se debería registrar la fecha de nacimiento del estudiante, y los temas aprendidos para que el que el docente pueda medir el aprendizaje de acuerdo con la

edad de los estudiantes en relación con los temas aprendidos.

En este sprint, el sistema de microservicios debería permitir guardar y consultar mensajes para que el video juego tenga un instrumento que le permita ser dinámico e intuitivo para el estudiante.

## RESULTADOS

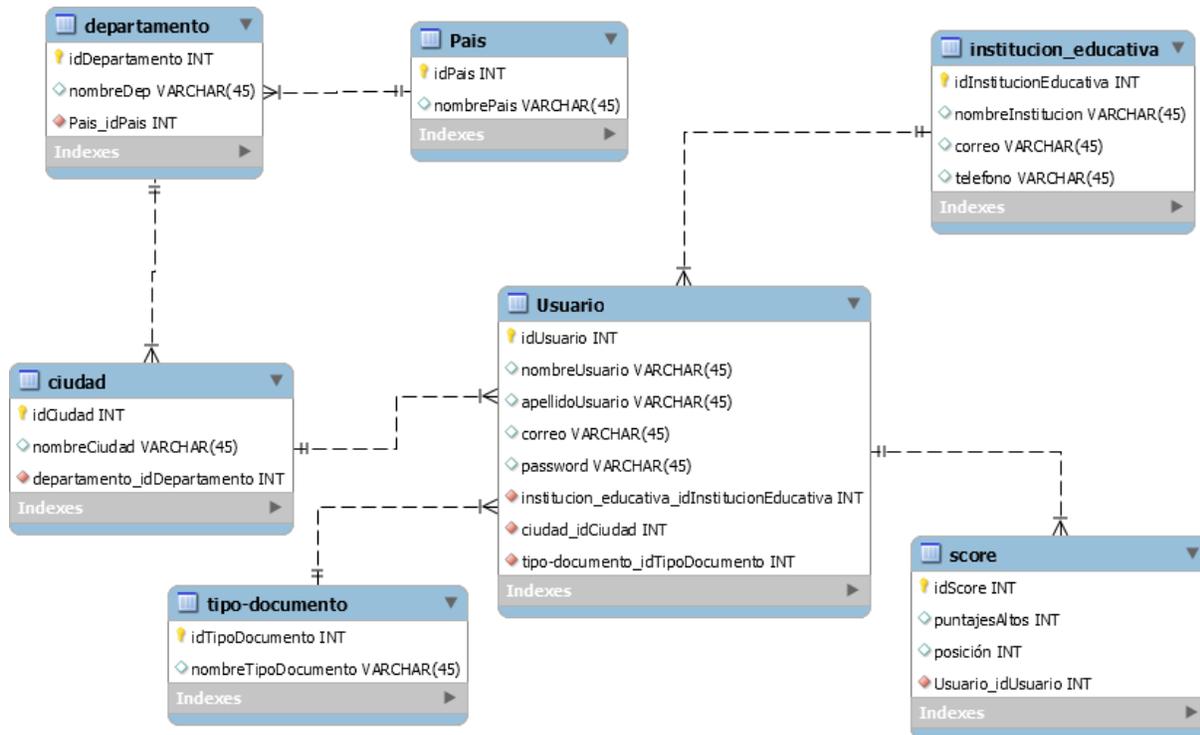
### 5.1 Resultados del Desarrollo e implementación

Durante el proceso de implementación de este proyecto se tuvo como resultado un sistema de información conformado por un conjunto de componentes de software totalmente desacoplados los cuales están conformados por diferentes funcionalidades y métodos, y cuya vida útil es totalmente independiente de la existencia de los otros componentes.

Para lograr construir este sistema, se dejó documentada las base de datos mediante el modelo **Figura 30** y se creó una base de datos en el motor de MySQL que de acuerdo con un plugin llamado vulnerability-managerdb, se logró en compañía del framework evitar ataques de las inyecciones SQL.

**Figura 30**

*Modelo relacional con la tabla tipo documento. (Elaboración Propia)*



También se deja implementado y probado el sistema de microservicios el cual tiene módulos desacoplados y son administrados a través de un módulo padre por medio de este archivo POM,

### Figura 31.

#### Figura 31

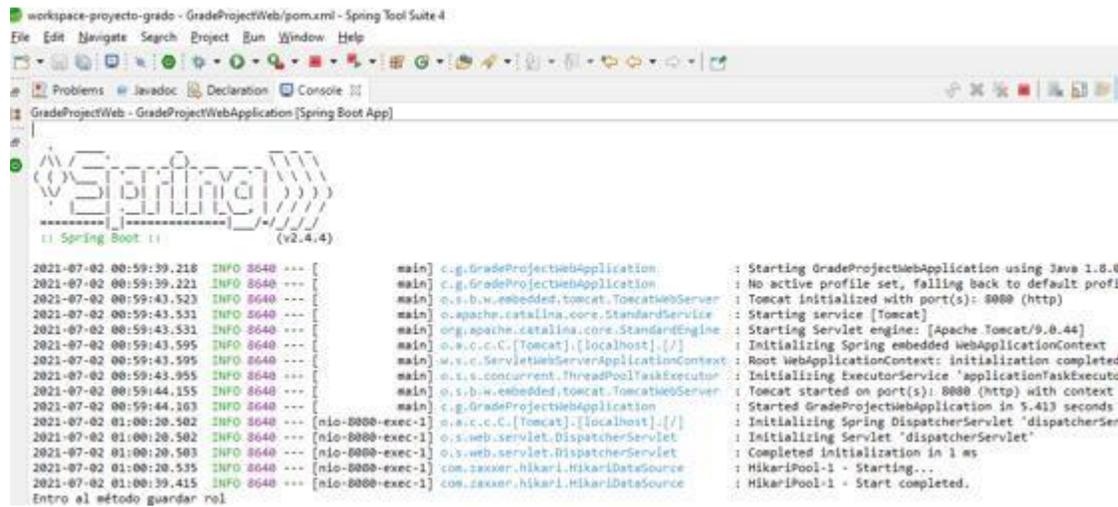
Pom con dependencias. (Elaboración Propia)

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-in:
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.1
<modelVersion>4.0.0</modelVersion>
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.4.4</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.gradoproject</groupId>
<artifactId>GradeProjectWeb</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>GradeProjectWeb</name>
<description>Proyecto de grado web</description>
<properties>
  <java.version>11</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
</project>
```

Como se puede evidenciar también cumple con estándares de buen performance ya que es un sistema eficiente que despliega módulos de una manera rápida demorándose 1 milisegundo como lo muestra el servidor, **Figura 32**.

**Figura 32**

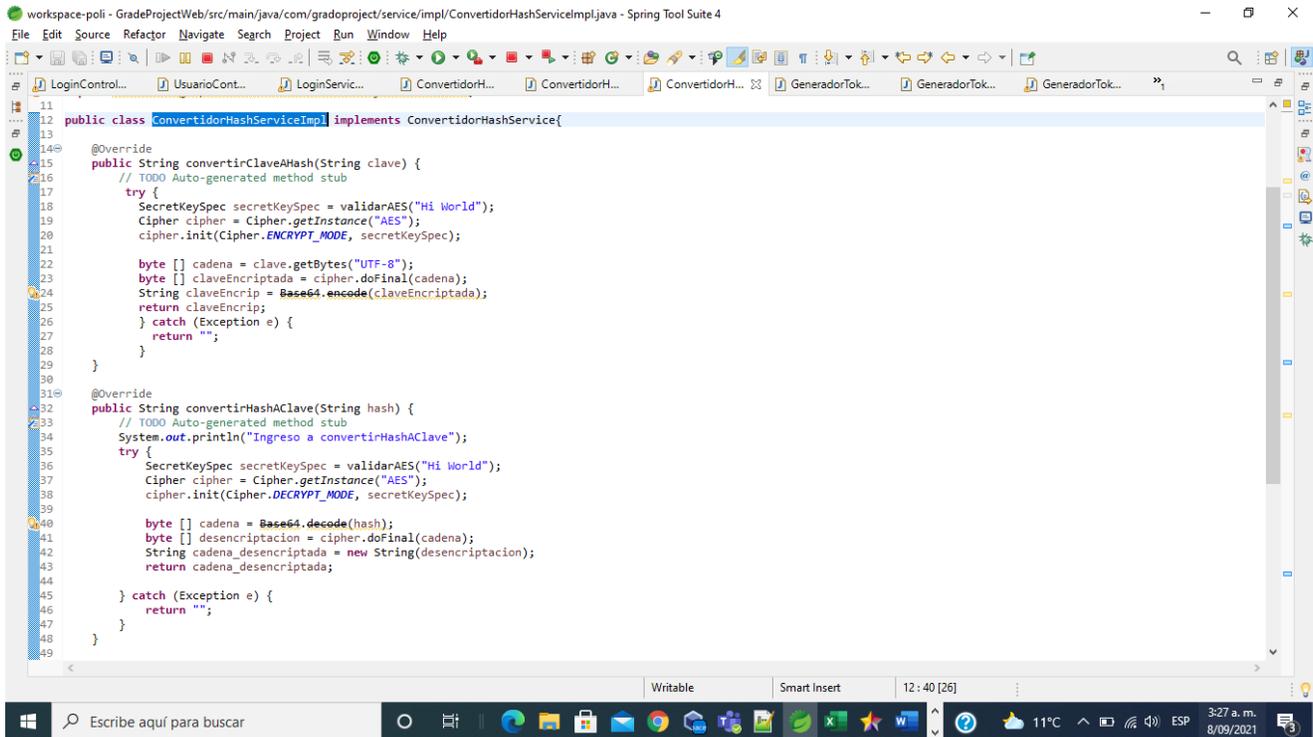
*Inicio del servidor.* (Elaboración Propia)



También se demuestra que el sistema es completamente mantenible y escalable ya que las funcionalidades de los componentes son más fáciles de reconocer, cambiar y adicionar por el tipo de arquitectura en la que está implementado porque como se puede ver en el caso de la imagen cada uno tiene funcionalidades específicas, y en una sola clase no se crean cientos de líneas de código, además, al ser componentes independientes se pueden corregir y desplegar sin afectar todo el sistema que está funcionando Se implementó un microservicio que encripta los datos de la contraseña cuando se guarda un usuario, y los desencripta cuando el usuario los consulta o se va a hacer alguna validación como se puede distinguir en **Figura 33** y en log **Figura 34**.

Figura 33

Capa Service java. (Elaboración Propia)



```
11 public class ConvertidorHashServiceImp implements ConvertidorHashService {
12
13
14 @Override
15 public String convertirClaveAHash(String clave) {
16 // TODO Auto-generated method stub
17 try {
18     SecretKeySpec secretKeySpec = validarAES("Hi World");
19     Cipher cipher = Cipher.getInstance("AES");
20     cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);
21
22     byte [] cadena = clave.getBytes("UTF-8");
23     byte [] claveEncriptada = cipher.doFinal(cadena);
24     String claveEncrip = Base64.encode(claveEncriptada);
25     return claveEncrip;
26 } catch (Exception e) {
27     return "";
28 }
29 }
30
31 @Override
32 public String convertirHashAClave(String hash) {
33 // TODO Auto-generated method stub
34 System.out.println("Ingreso a convertirHashAClave");
35 try {
36     SecretKeySpec secretKeySpec = validarAES("Hi World");
37     Cipher cipher = Cipher.getInstance("AES");
38     cipher.init(Cipher.DECRYPT_MODE, secretKeySpec);
39
40     byte [] cadena = Base64.decode(hash);
41     byte [] descriptacion = cipher.doFinal(cadena);
42     String cadena_descriptada = new String(descriptacion);
43     return cadena_descriptada;
44 } catch (Exception e) {
45     return "";
46 }
47 }
48
49 }
```

Figura 34

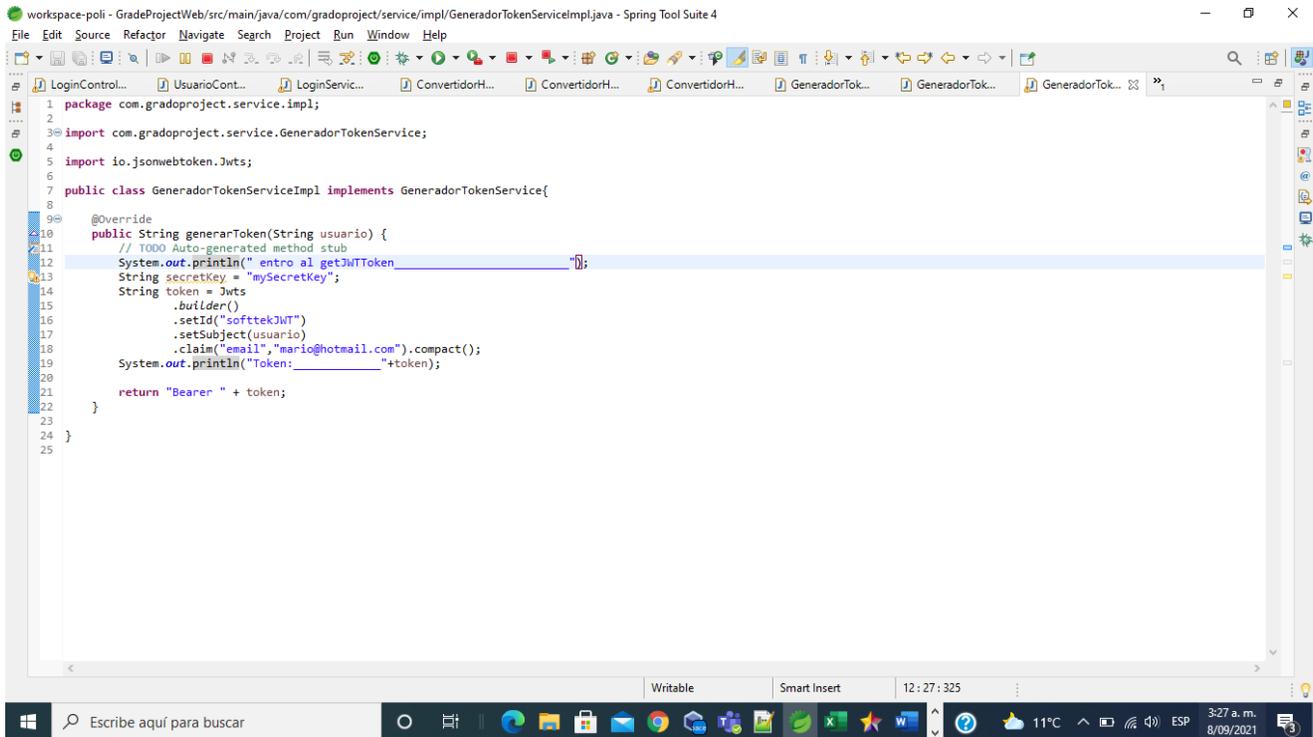
Log de servidor con hash Aes. (Elaboración Propia)

```
encriptada-_____Q8bgYWR+YqUZGfjZCNvgmA==
Ingreso a encriptar aes
cadena _____[B@751867c0
md _____SHA-1 Message Digest from SUN, <initialized>

cadena2 _____[B@7c194984
cadena3 _____[B@2355da65
secretKeySpecEncr _____javax.crypto.spec.SecretKeySpec@fffe8dd1
Ahora vamos a desencriptar
desencriptada-_____ 'c1avelIno'
```

**Figura 35**

*Construcción hash java. (Elaboración Propia)*



**Figura 36**

*Log con token(jwt). (Elaboración Propia)*

```
2021-09-07 19:23:36.587 INFO 5992 --- [nio-8585-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
entro al validarCredenciales_____
entro al getJWTToken_____
Token: _____eyJhbGciOiJIub251In0.eyJqdGkiOiJzb2Z0dGVRSlduIiwic3ViIjoicm9vdCIsmVtYnlzIjoibWYyaW9AaG90bWVpbC5jb20ifQ.
Token: _____Bearer eyJhbGciOiJIub251In0.eyJqdGkiOiJzb2Z0dGVRSlduIiwic3ViIjoicm9vdCIsmVtYnlzIjoibWYyaW9AaG90bWVpbC5jb20ifQ.
```

Para finalizar el resultado del sistema backend, de acuerdo a las imágenes **Figura 35** y **Figura 36** en cada una de las transacciones nos podrá generar un token de seguridad.

## CONCLUSIONES

### 6.1 Objetivos

1. Identificar una arquitectura de software para garantizar estándares de calidad (bajo acoplamiento, escalabilidad, mantenibilidad y seguridad) para el correcto funcionamiento del video juego.

Se identificó la arquitectura basada en microservicios, la cual trabaja bajo componentes modulares totalmente separadas de la capa del video juego, y de una posible capa front, con esto, garantizar un bajo acoplamiento entre las capas del sistema y sus componentes.

Además, ya que son componentes separados, cada componente cuenta con sus archivos de configuración, y en el momento de la comunicación de un componente con otro, solo se necesitará la interfaz dando como resultado que, en el momento de hacer alguna implementación o algún mantenimiento, solo se toque y despliegue ese componente y que los demás servicios del sistema sigan funcionando sin ningún inconveniente.

Por último, para la seguridad del software, por un lado, se implementó Spring Security (Servicio que ofrece el framework spring para la seguridad del sistema) el cual es un servicio que brinda una capa adicional de seguridad para que en el momento en el que se vaya a utilizar algún microservicio web, se puedan evitar ataques de inyección SQL, y, por otro lado, se implementó JWT (Jason Web Token) para generar códigos de seguridad

al usuario que se registre, y con esto evitar que otros usuarios, que hayan obtenido las credenciales de forma corrupta, puedan ingresar al sistema.

2. Estructurar una base de datos que pueda servirle al video juego como banco de información, para que este pueda almacenarla y consultarla de una manera más segura y eficiente.

Se trabajo con el motor de base de datos MySQL el cual sirvió como programa para diseñar y desarrollar cada una de las tablas que iban a almacenar la información del video juego. Además, se usaron prácticas de normalización de base de datos para optimizar las tablas y evitar que haya data basura en el sistema.

3. Demostrar que existen mejores prácticas y formas de desarrollo de un video juego para que futuros programadores las puedan implementar.

Por medio del proyecto se demuestra que el trabajar por medio de una arquitectura de microservicios ayuda a que los desarrolladores de video juegos se puedan concentrar en la lógica y la historia del video juego y además tengan un sistema de alto rendimiento y seguro.

## 6.2 Pregunta de Investigación

1. ¿Bajo qué concepto se escogió el video juego como medio de aprendizaje?

Se escogió bajo el concepto o técnica llamada “gamificación”.

2. ¿Qué es la Gamificación?

La gamificación es una técnica de aprendizaje para los estudiantes por medio de la cual se propone el juego como un medio de aprendizaje, utilizando la motivación, la recompensa y la experiencia para que el estudiante tenga como resultado un resultado significativo y funcional. Esto quiere decir que quién va a trabajar bajo esta técnica puede tomar cualquier tema y aplicarla por medio de un juego, a esto se le puede llamar gamificación o juego serio. (Juego serio es cuando en un concepto académico o laboral se aplica la gamificación para tener algún tipo de resultado).

La gamificación apoya el concepto de la motivación y promueve la competencia, esto lo hace a partir de retos, diversión, y aprovechamiento del tiempo libre. El aprendizaje está ligado a la motivación, entonces la idea es utilizar la gamificación como potenciador para que se mejore el aprendizaje.

3. ¿Por qué se escogió el álgebra como tema de estudio para el video juego?

Se escogió porque el álgebra es la raíz del cálculo, y es la raíz de la trigonometría, por lo tanto, se considera como la parte más importante de las matemáticas escolares. Es el tema

más difícil de asimilar para un niño, porque es donde comienza a ver el termino de manejo de variables.

4. ¿Bajo qué concepto se escogió el video juego como medio de aprendizaje?

Se escogió bajo el concepto o técnica llamada “gamificación”.

### **6.3 Conclusión**

1. La tecnología cada día está evolucionando, esto hace que quienes trabajan en aquellas áreas relacionadas estén obligados a documentarse ya que como se ve en este documento existen otras maneras desarrollar video juegos, y otro tipo de aplicaciones que puedan tener un mejor performance.
2. Es muy importante verificar que, si creamos algún componente tecnológico, sepamos qué consecuencias puede traer el no evaluar posibles riesgos como por ejemplo riesgos de seguridad.
3. La arquitectura de microservicios es una alternativa que no solo sirve para comunicar y enviar mensajes a través de software empresariales, sino que también puede acondicionarse a diferentes aspectos.

## **TRABAJO FUTURO**

Para el trabajo futuro, con el objetivo de complementar el sistema planteado y desarrollado en este documentado, se propone desarrollar un sistema web para el frontend que muestre la data del video juego, el cual esté construido por medio de alguna tecnología que esté en la vanguardia, como por ejemplo Angular o React. Esto con el fin de seguir implementando buenas prácticas, seguir teniendo un bajo acoplamiento entre capas, y además contar con un framework que posibilite seguir trabajando con una arquitectura modular.

## REFERENCIAS

[1 J.D. López Hinojosa «Universidad Técnica del Norte» (2017) Arquitectura de Software ] Basada en Microservicios para Desarrollo de Aplicaciones web de la asamblea nacional [En línea]. Available:

<http://repositorio.utn.edu.ec/bitstream/123456789/7603/1/PG%20569%20TESIS.pdf>.

[2 J.M. Pascagaza Gutierrez, «Universidad Católica» (2018). Desarrollo de un Sistema de ] Información para la Gestión de los Proyectos de Responsabilidad Social del Programa de Ingeniería de Sistemas de la Universidad Católica de Colombia [En línea]. Available:

[https://repository.ucatolica.edu.co/bitstream/10983/16047/1/Trabajo%20de%20grado\\_juan%20manuel\\_625353.pdf](https://repository.ucatolica.edu.co/bitstream/10983/16047/1/Trabajo%20de%20grado_juan%20manuel_625353.pdf)

[3 F. S. García Victoria, F. E. Daza Gonzalez «Universidad de San Buenaventura,»(2017)

] *Diseño y construcción de un componente para el generador de código ZathuraCode que*

*genere proyectos con arquitectura basada en microservicios en Java* [En línea]. Available:  
[http://bibliotecadigital.usb.edu.co/bitstream/10819/4581/1/Dise%C3%B1o\\_Construcci%C3%B3n\\_Co  
mponente\\_2017.pdf](http://bibliotecadigital.usb.edu.co/bitstream/10819/4581/1/Dise%C3%B1o_Construcci%C3%B3n_Co<br/>mponente_2017.pdf)

[4 A. Villardon Garcia, «Facultat de Belles Arts de Sant Carles,» Investigación y Desarrollo de  
] un video Juego, [En línea]. Available:

[https://riunet.upv.es/bitstream/handle/10251/93814/VILLARD%C3%93N%20-  
%20INVESTIGACI%C3%93N%20Y%20DESARROLLO%20DE%20UN%20VIDEOJUE  
GO%20PROTOTIPO%20VISUAL%20DE%20VIDEOJUEGO.pdf?sequence=1](https://riunet.upv.es/bitstream/handle/10251/93814/VILLARD%C3%93N%20-<br/>%20INVESTIGACI%C3%93N%20Y%20DESARROLLO%20DE%20UN%20VIDEOJUE<br/>GO%20PROTOTIPO%20VISUAL%20DE%20VIDEOJUEGO.pdf?sequence=1).

[5 E.G. Maida, «Universidad Católica de Argentina,» 2015. Metodologías de desarrollo de  
] software [En línea]. Available:

[https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-  
software.pdf](https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-<br/>software.pdf).

[6 A. Kuznik, «Universitat Autònoma de Barcelona,» (2007) Marco conceptual y  
] metodològicopara un estudio sobre las tareas de traducción e interpretación desempeñadas  
en el ámbito laboral. [En línea]. Available: <https://core.ac.uk/download/pdf/13284116.pdf>.

[7 J. A. Sarmiento Forero, «Universidad Libre,» Sistema de Información y Gestión de Proyectos de  
] Grado [En línea]. Available:

[https://repository.unilibre.edu.co/bitstream/handle/10901/8875/Sistema%20de%20  
informacion%20y%20gestion%20de%20proyectos%20de%20grado.pdf?sequence=1](https://repository.unilibre.edu.co/bitstream/handle/10901/8875/Sistema%20de%20<br/>informacion%20y%20gestion%20de%20proyectos%20de%20grado.pdf?sequence=1).

[8 G. C. Plataforma, «Google Cloud Plataforma,» Sistema de Monitoreo de Temperatura y  
] Humedad del Data Center de Halliburton calle 113. [En línea]. Available:

<https://repository.usta.edu.co/bitstream/handle/11634/3379/Diazmanuel2017.pdf?sequence=1>

[9 D. G. Mohd Rehan Ghazi, «Hadoop, mapreduce and HDFS: A developers perspective,»  
] *Procedia Computer Science*, vol. 48, pp. 45-50, 2015.

[1 J. R. Restrepo Escobar, «Universidad EAN,»(2012) Aplicación web para el control y  
0] seguimiento de avances del proyecto de grado3. [En línea]. Available:

<https://repository.ean.edu.co/bitstream/handle/10882/4386/RestrepoJose2012.pdf;jsessionid=C8F935082F2BAF03E0B5B396E8A6E36A?sequence=1>

[1 S. Restrepo Martinez, «UNAD,» Proyecto de grado desarrollo aplicativo web instituto ABC.  
1] [En línea]. Available:

<https://repository.unad.edu.co/jspui/bitstream/10596/38212/3/srestrepom.pdf>

[1 E. Londoño Gutierrez, «Universidad Autónoma de Occidente,» (2016) Implementación de un  
2] servicio web para la comunidad de profesionales universitarios de Colombia. [En línea].

Available:

<https://red.uao.edu.co/bitstream/handle/10614/9477/T07147.pdf?sequence=1&isAllowed=y>

[1 F. A. Chesquesillo Sanchez, «Universidad Privada Ada A Byron,»(2012) Desarrollo e  
3] Implementación de un Portal Web como Alternativa de solución para mejorar la calidad del  
servicio del Cementerio General en la provincia de Chincha. [En línea]. Available:

<http://repositorio.autonmadeica.edu.pe/bitstream/autonmadeica/45/3/FREDY%20ALDO%20CHOQUESILLO%20SANCHEZ%20%20-%20DESARROLLO%20E%20IMPLEMENTACION%20DE%20UN%20PORTAL%20WEB.pdf>

[1 L. Lara Otalora, L. K. Tovar Quintero, L.S. Martinez Barreto. «Fundación Universitaria los  
4] Libertadores,»(2015) Proyecto de grado desarrollo aplicativo web instituto ABC. [En línea].

Available:

<https://repository.libertadores.edu.co/bitstream/handle/11371/584/LaraOtaloraLizeth.pdf?sequence=2&isAllowed=y>

[1 A. V. Gonzalez. , «Universidad del Aconcagua,»(2001) Aprendizaje en el Niño con trastorno  
5] por deficit de atención e hiperactividad. [En línea]. Available:

[http://bibliotecadigital.uda.edu.ar/objetos\\_digitales/6/tesis-364-aprendizaje.pdf](http://bibliotecadigital.uda.edu.ar/objetos_digitales/6/tesis-364-aprendizaje.pdf)

[1 J. M. Vela Gomez, J. C. ORTIZ SORIANO, «Universidad de San buenaventura,»(2008)

6] Desarrollo de un Aplicativo para el Registro y Seguimiento de la información de los procesos  
judiciales. [En línea]. Available:

<http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/42796.pdf>

[1 Cyberseguridad , «Cyberseguridad,»(2021) Ataques de cyberseguridad a la red. [En línea].

7] Available: [https://ciberseguridad.com/normativa/espana/medidas/seguridad-red/#Ataque\\_de\\_seguridad\\_a\\_la\\_red](https://ciberseguridad.com/normativa/espana/medidas/seguridad-red/#Ataque_de_seguridad_a_la_red)

[1 S. Sassone, «S. Sassone,»(2016) Sistema de Recomendación con conocimiento Web. [En línea].

8] Available: <https://www.welivesecurity.com/la-es/2016/08/16/seguridad-desarrollo-de-videojuegos/>

[1 J. A. Recio García, «Universidad Complutense de Madrid,»(2015) Sistema de Recomendación  
9] con conocimiento Web. [En línea]. Available:

<https://eprints.ucm.es/id/eprint/32695/1/Memoria.pdf>

- [2 B. Chavarria Neira, E. Gudiño de la A «Universidad Politécnica Salesiana,»(2017)
- 0] Implementación de un Servidor Web y un Diseño de una Página utilizando Herramientas de Software Libre para el dispensario "Sagrada Familia de la Ciudad de GUayaquil". [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/14162/1/GT001840.pdf>
- [2 J. H. Robayo Lopez, R. M. Rodriguez Rodriguez «UNAD» (2015) Aseguramiento de los Sistemas Computacionales de la Empresa Sitiosdima.net ". [En línea]. Available: 1] <https://repository.unad.edu.co/bitstream/handle/10596/3818/79626344.pdf?sequence=5>

García Victoria, F. S., & Daza Gonzalez, E. F. (2017). *Diseño y construcción de un componente para el generador de código ZathuraCode que genere*. Santiago de Cali.

Lopez Hinojosa, J. D. (2017). *Arquitectura de Software Basada en Microservicios Para el desarrollo de Aplicaciones Web de La Asamblea Nacional*. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/7603/1/PG%20569%20TESIS.pdf>

López Hinojosa, J. D. (2017). *Arquitectura de software de microservicios para el desarrollo de aplicaciones web de la asamblea nacional*. Ibarra.

Lopez Hinojosa, J. D. (s.f.). *Arquitectut*. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/7603/1/PG%20569%20TESIS.pdf>

Pascagaza Gutierrez, J. M. (2018). *Desarrollo de un sistema de información para la gestión de los proyectos de responsabilidad Social del programa de Ingenieria de Sistemas de la Universidad Católica de Colombia*. Bogotá.

Pascagaza Gutierrez, J. M. (s.f.). *Desarrollo de Insistema de Información Para la gestión de los*

*Proyectos de Responsabilidad Social del programa de Ingeniería de Sistemas de la Universidad Católica de Colombia. Obtenido de*

[https://repository.ucatolica.edu.co/bitstream/10983/16047/1/Trabajo%20de%20grado\\_juan%20manuel\\_625353.pdf](https://repository.ucatolica.edu.co/bitstream/10983/16047/1/Trabajo%20de%20grado_juan%20manuel_625353.pdf)

Villardón García, A. (2017). *Investigación y desarrollo de un video juego*. Valencia.

Villardón García, Á. (2017). *Investigación y desarrollo de un video Juego*. Obtenido de

<https://riunet.upv.es/bitstream/handle/10251/93814/VILLARD%C3%93N%20-%20INVESTIGACI%C3%93N%20Y%20DESARROLLO%20DE%20UN%20VIDEOJUEGO%20PROTOTIPO%20VISUAL%20DE%20VIDEOJUEGO.pdf?sequence=1>

Villardón García, A. (s.f.). *Investigación y desarrollo de un video juego*. Obtenido de

<https://riunet.upv.es/bitstream/handle/10251/93814/VILLARD%C3%93N%20-%20INVESTIGACI%C3%93N%20Y%20DESARROLLO%20DE%20UN%20VIDEOJUEGO%20PROTOTIPO%20VISUAL%20DE%20VIDEOJUEGO.pdf?sequence=1>

## ANEXOS

### 9.1 Manual Técnico

Para conocer el funcionamiento de los componentes lógicos del sistema se debe ir al siguiente manual:



*Anexo 1 Manual Técnico*

### 9.2 Manual de Instalación y Configuración

Con el siguiente manual se podrá tener una guía para conocer los archivos de instalación y configuración y su proceso:



*Anexo 2 Manual de Instalación*

### 9.3 Manual de Despliegue

Se debe utilizar el siguiente manual de despliegue para conocer los componentes del sistema que se deben desplegar en el servidor y como se debe hacer el proceso.



*Anexo 3 Manual de despliegue*

### 9.4. Plan de Trabajo



*Anexo 4 Plan de trabajo*

### 9.5. Scripts Mysql



*Anexo 4 Scripts Mysql*

### 9.6. Fuentes





