SERVICIO DE GESTION DE NORMAS LEGISLATIVAS

INFORME DE PRÁCTICA PROFESIONAL SEGUNDO AÑO

Autor: David Ernesto Pino Díaz

Tutor: M.Sc Marlen Mora Molina

HOLGUÍN 2025

RESUMEN

Esta investigación tiene como objetivo desarrollar un servicio *API* (*Application Programming Interface***)** que permita a aplicaciones móviles ya desarrolladas dinamizar el contenido de aprendizaje de leyes mediante técnicas de gamificación. Dada la complejidad de los conceptos jurídicos y la necesidad de métodos educativos más interactivos, el estudio busca mejorar la retención y motivación de los estudiantes. La solución propuesta consiste en una API *GraphQL* que centraliza y actualiza el contenido legal, facilitando su integración en apps existentes. Para su desarrollo, se emplearán tecnologías como Nest.js (*backend*) y metodologías ágiles como *XP* (*Extreme Programming*) para garantizar flexibilidad y calidad. La implementación de este sistema permitirá evaluar su impacto en la experiencia de aprendizaje y su escalabilidad para futuras integraciones.

Palabras clave: *API*, aplicaciones móviles, gamificación, aprendizaje de leyes, Nest.js, metodología *XP*, educación legal interactiva.

ABSTRACT

This work focuses on developing mobile applications for law learning using gamification techniques. Given the complexity of legal concepts and the need for more interactive educational methods, this proposal aims to enhance student engagement and knowledge retention. Key objectives include designing an intuitive app, integrating game-like elements such as rewards and challenges, and assessing its impact on learning. Technologies like Nest.js (as backend framework) to guide the development, the XP (Extreme Programming) agile development methodology will be used. Results demonstrated increased user participation and significant improvement in legal content comprehension, validating gamification’s effectiveness in this field.

Keywords**:** API, mobile applications, gamification, legal learning, Nest.js, XP methodology, interactive legal education.

ÍNDICE

[RESUMEN 2](#_Toc202864691)

[ABSTRACT 3](#_Toc202864692)

[ÍNDICE 4](#_Toc202864693)

[INTRODUCCIÓN 1](#_Toc202864694)

[DESARROLLO 4](#_Toc202864695)

[Requerimientos Funcionales 4](#_Toc202864696)

[Formato 4](#_Toc202864697)

[Epígrafes 4](#_Toc202864698)

[CONCLUSIONES 5](#_Toc202864699)

[RECOMENDACIONES 6](#_Toc202864700)

[REFERENCIAS BIBLIOGRÁFICAS Y BIBLIOGRAFÍA 7](#_Toc202864701)

[GLOSARIO 8](#_Toc202864702)

[ANEXOS I](#_Toc202864703)

INTRODUCCIÓN

La gamificación es una técnica que incorpora elementos propios de los juegos, como recompensas, desafíos, niveles, puntuaciones y competencias en contextos no lúdicos, con el fin de motivar la participación, mejorar el compromiso y facilitar el aprendizaje. En el ámbito educativo, esta estrategia ha demostrado ser efectiva para aumentar la retención de conocimientos, fomentar la colaboración y hacer más interactivo el proceso de enseñanza. (Martínez Rodríguez, 2023)

Diversos estudios respaldan que la gamificación estimula la dopamina, un neurotransmisor asociado con la motivación y la satisfacción, lo que favorece un aprendizaje más significativo. Plataformas como Duolingo, ¡Kahoot! y Quizlet han popularizado este enfoque, mostrando cómo mecánicas de juego pueden transformar experiencias educativas tradicionales en dinámicas más atractivas y efectivas. (Gómez & Fernández, 2022)

El estudio de las leyes y normativas jurídicas suele presentar desafíos significativos para los estudiantes, debido a su alto nivel de abstracción, la densidad de su terminología y la necesidad de memorizar numerosos artículos y casos prácticos. Los métodos tradicionales de enseñanza, basados en lectura pasiva y repetición, a menudo resultan poco estimulantes, lo que genera desinterés y bajo rendimiento académico. (Gómez & Fernández, 2022)

Conscientes del potencial de la gamificación para transformar la educación jurídica, la Universidad de Holguín ha desarrollado aplicaciones móviles innovadoras que integran mecánicas de juego en el aprendizaje de leyes. Estas herramientas buscan abordar los desafíos tradicionales de la disciplina ,como la complejidad terminológica, la abstracción conceptual y la necesidad de memorización, mediante dinámicas interactivas.

Tras la implementación de estas aplicaciones móviles, se identificaron limitaciones significativas que afectaban la experiencia de aprendizaje. Entre las principales deficiencias destacan:

1. Estaticidad de las leyes:

* La información legal se encuentra embebida directamente en el código de las aplicaciones
* Requiere actualizaciones manuales mediante nuevas versiones de la app
* Existe un desfase temporal entre reformas legales y su incorporación a la plataforma
* Falta de un sistema centralizado de gestión de contenidos

1. Ausencia de Elementos Gamificados Esenciales:

* Carece completamente de un sistema de ranking o clasificación de usuarios
* No existen mecanismos de competencia sana entre estudiantes
* Falta de logros o *badges* que reconocieran el progreso académico
* No se implementan recompensas por cumplimiento de objetivos

1. Limitaciones en el Proceso de Apredizaje:

* Imposibilidad de generar reportes sobre el progreso
* Falta de datos para identificar áreas problemáticas comunes
* No existe retroalimentación automatizada para los estudiantes

1. Problemas Técnicos y de Experiencia de Usuario:

* Ausencia de personalización según perfil de aprendizaje
* No aprovecha el potencial de notificaciones *push*

Estas limitaciones reducen el potencial interactivo de las apps y su capacidad para mantener el *engagement* (compromiso) de los estudiantes, señalando la necesidad de un mecanismo centralizado que garantizara contenido jurídico actualizable en tiempo real y métricas de desempeño comparables.

El principal objetivo de esta investigación es desarrollar una *API* robusta que solucione las limitaciones detectadas en las aplicaciones móviles de aprendizaje jurídico de la Universidad de Holguín, centrándose en:

1. Dinamización de contenidos:

* Garantizar la actualización en tiempo real de leyes, normativas y casos prácticos mediante una base de datos centralizada.
* Implementar un sistema de sincronización automática para que las apps reflejen cambios legislativos sin requerir actualizaciones manuales.

1. Sistema de *ranking* y gamificación avanzada:

* Diseñar un modelo de puntuación basado en logros, niveles y competencias entre usuarios.
* Integrar tablas de clasificación (*leaderboards*) para fomentar la motivación mediante la comparación de progreso.

1. Estudio del proceso de aprendizaje:

* Generar informes personalizados sobre áreas de dificultad, tiempo de estudio y dominio de temas.

1. Personalización de la experiencia de aprendizaje:

* Mostrar contenidos acordes al progreso y nivel de conocimiento del usuario.

# DESARROLLO

La Universidad de Holguín (UHo), fundada en 1973, es una institución pública cubana comprometida con la formación integral de profesionales, la investigación científica y la vinculación social. Su misión se centra en contribuir al desarrollo sostenible de la sociedad mediante la docencia de calidad, la generación de conocimientos innovadores y la transferencia tecnológica, con especial énfasis en las necesidades del territorio holguinero. Entre sus objetivos estratégicos destacan: fortalecer la formación académica con enfoque multidisciplinario, promover investigaciones que resuelvan problemas locales y nacionales (ej.: desarrollo agroindustrial, tecnologías educativas y energía renovable), y fomentar la extensión universitaria mediante proyectos comunitarios y culturales. La UHO ofrece programas en ingenierías, ciencias sociales, humanidades y pedagogía, integrando tecnologías digitales en sus procesos formativos. (Universidad de Holguín, 2022)

El proyecto se desarrollará utilizando un *stack* tecnológico moderno y robusto centrado en Nest.js, un *framework* progresivo de Node.js que permite construir aplicaciones del lado del servidor eficientes, confiables y escalables gracias a su arquitectura modular y soporte nativo para TypeScript. Nest.js facilita la creación de APIs RESTful y GraphQL mediante decoradores y módulos organizados, integrando perfectamente características como inyección de dependencias y programación orientada a aspectos. Para el manejo de datos, se implementará GraphQL con Apollo Server, que ofrece un enfoque flexible y eficiente para las consultas de datos, permitiendo a los clientes móviles especificar exactamente qué información necesitan, reduciendo el ancho de banda y mejorando el rendimiento. Como sistema de gestión de base de datos se utilizará PostgreSQL, un sistema de base de datos relacional *open-source* conocido por su confiabilidad, robustez y soporte avanzado para consultas complejas, combinado con TypeORM como ORM (*Object*-*Relational* *Mapping*) para facilitar la interacción entre la aplicación y la base de datos mediante un enfoque orientado a objetos, permitiendo una fácil definición de entidades, relaciones y transacciones.(NestJS Team, 2023)

El entorno de desarrollo principal será Visual Studio Code (VSCode), un editor de código ligero pero potente, altamente personalizable y con un ecosistema extenso de extensiones que mejoran la productividad. Entre las extensiones clave se incluirán ESLint para mantener un código limpio y consistente mediante el análisis estático, Prettier para el formateo automático del código, REST Client para probar endpoints API directamente desde el editor, y Docker para gestionar contenedores y servicios. Para garantizar la calidad del código, se implementará un conjunto robusto de pruebas automatizadas utilizando Jest como framework de pruebas unitarias e Supertest para pruebas de integración y E2E (end-to-end), permitiendo validar el comportamiento de la API en diferentes escenarios. La integración y entrega continuas (CI/CD) se gestionarán mediante GitHub Actions, automatizando procesos como la ejecución de pruebas, el análisis de código y los despliegues en diferentes entornos. Para garantizar la consistencia entre los entornos de desarrollo, pruebas y producción, se utilizará Docker para la contenedorización de la aplicación y sus dependencias, permitiendo una configuración reproducible y aislada.(Microsoft, 2023)

El desarrollo del proyecto seguirá los principios y prácticas de Extreme Programming (XP), una metodología ágil que enfatiza la calidad del software, la adaptabilidad a cambios y la satisfacción del cliente. XP se implementará mediante iteraciones cortas (sprints de 1-2 semanas) que permitirán entregas incrementales y feedback constante. Entre las prácticas clave se incluirá Pair Programming (programación en parejas), donde dos desarrolladores trabajarán juntos en una misma estación de trabajo, fomentando la colaboración, el intercambio de conocimientos y la detección temprana de errores. El Desarrollo Guiado por Pruebas (TDD) será fundamental, escribiendo primero las pruebas unitarias que definan el comportamiento esperado del código antes de implementar la funcionalidad, asegurando un diseño robusto y mantenible. La Integración Continua (CI) garantizará que los cambios en el código se integren y prueben automáticamente varias veces al día, detectando y corrigiendo problemas rápidamente. Además, se realizarán reuniones diarias (stand-ups) para mantener al equipo sincronizado, identificar obstáculos y ajustar prioridades. XP también fomenta la simplicidad en el diseño, evitando sobre-ingeniería y priorizando funcionalidades que agreguen valor real al cliente, así como la retroalimentación continua con los stakeholders para ajustar requisitos y asegurar que el producto final cumpla con las expectativas.(Beck, 2000)

Requerimientos Funcionales

**1. Gestionar dinámicamente leyes**  
1.1 Añadir leyes  
1.2 Editar leyes   
1.3 Mostrar leyes

1.4 Eliminar leyes

**2. Implementar sistema de gamificación y ranking**  
2.1 Calcular puntuaciones basadas en actividades completadas  
2.2 Generar tablas de clasificación (*leaderboards*) por materias jurídicas

**3. Realizar seguimiento y análisis del aprendizaje**  
3.1 Registrar el progreso individual  
3.2 Generar reportes estadísticos

**4. Administrar usuarios y permisos**  
4.1 Registrar usuario

4.2 Asignar rol al usuario (administrador, estudiante, profesor)

4.3 Editar usuario

4.4 Mostrar usuario

4.5 Eliminar usuario

**5. Gestionar comunicación y notificaciones**

5.1 Crear notificaciones

5.2 Eliminar notificaciones  
5.3 Enviar notificaciones push sobre actualizaciones legales relevantes

**6. Personalizar la experiencia de usuario**  
6.1 Adaptar contenidos según nivel de conocimiento del usuario  
6.2 Recomendar rutas de estudio personalizadas

**7. Gestionar dinamicidad de los tests**

7.1 Añadir test

7.2 Editar test

7.3 Mostrar test

7.4 Eliminar test

Diagrama de Actores

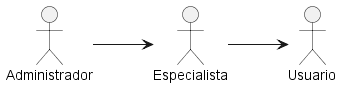


Figura 1: Diagrama de Actores

Diagrama de Casos de Uso

Figura 2: Diagrama de Caso de Uso “Gestionar dinamicidad de los tests

---------------------------------------------------------------------------------------------------------------------------------------

Aquí se ponen los aspectos a reflejados en el trabajo y sus resultados. Entre 15 y 20 hojas

## Formato

El documento a entregar seguirá el formato siguiente:

* Hoja carta con 2 cm de margen a cada lado.
* Letra Arial 12 con interlineado 1.5

## Epígrafes

Puede emplear los epígrafes que considere necesarios para explicar un tema más específico dentro de otro más general.

**Desarrollo**

En el desarrollo debe empezar por explicar grosso modo la entidad en la que está haciendo las prácticas. De tal forma que se comprenda qué es lo que va a hacer, para quién lo va a hacer, dónde lo va a hacer y los beneficios que le traerá a la entidad para la que lo va a realizar.

Exponer bien claramente la solución dada por usted a la problemática planteada.

En el ciclo de desarrollo de software debe contener la solución propuesta hasta la etapa de diseño detallado. (Según la metodología de software escogida)

Debe hacer un análisis de los resultados obtenidos.

Debe analizar los beneficios que se obtienen con los resultados de la solución del problema. Estos beneficios pueden ser:

* Beneficios de carácter científico, tecnológico y técnico.
* Beneficios económicos tangibles.
* Beneficios intangibles (económicos, sociales y políticos).

# CONCLUSIONES

Resultados fundamentales a los que se arribó

# RECOMENDACIONES

Cualquier aspecto que usted considere que se debe seguir tratando o darle seguimiento una vez concluido el trabajo.

# REFERENCIAS BIBLIOGRÁFICAS Y BIBLIOGRAFÍA

Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.

Gómez, A., & Fernández, T. (2022). *Gamificación en la educación legal: Tendencias y oportunidades*. Editorial Jurídica Digital.

Martínez Rodríguez, Y. (2023). Experiencias de gamificación en la enseñanza del derecho en Cuba. *Memorias del 6to Taller Nacional de Derecho e Innovación*, 89-95.

Microsoft. (2023). *Visual Studio Code* [Software]. https://code.visualstudio.com

NestJS Team. (2023). *NestJS Documentation*. https://docs.nestjs.com

Universidad de Holguín. (2022). *Plan Estratégico de Desarrollo 2022-2025*. Ediciones Universidad de Holguín. https://www.uho.edu.cu

# GLOSARIO

(Opcional) Listado de términos ordenados alfabéticamente, que son usados en el trabajo y cuyo significado es poco conocido. Se indica para cada término su significado.

# ANEXOS

* (Opcional) Documentos que complementan el cuerpo del trabajo, pero que no son indispensables para entenderlo. Todos deben ser referenciados en el cuerpo del documento.
* Se organizarán en orden consecutivo según su aparición en el desarrollo del trabajo
* En ellos pueden exponerse tablas, gráficos, detalles de instrumentos y técnicas utilizadas, resultados estadísticos y materiales complementarios recomendados para la introducción de los resultados de investigación en la práctica
* Los anexos bien utilizados pueden ser un recurso efectivo para lograr ajustarse al número de páginas exigidas para el cuerpo de la tesis (60 páginas)
* Los anexos se comienzan a enumerar desde uno con números romanos (I, II, III, IV, V etc.)

**Normas de formato para las tesis de Diplomas**

* Tamaño: máximo 60 páginas, desde la Introducción hasta las Referencias Bibliográficas
* Configuración de página: formato carta 8 ½ \* 11, en una columna
* Las páginas se numerarán con números arábigos consecutivamente en la parte inferior derecha de la página
* Formato de fuente: Arial 12
* Formato de párrafo, a un espacio y medio (interlineado 1.5), sin sangría, con márgenes:

margen izquierdo---------- 3,5 cm

margen superior------------3,0 cm

margen inferior-------------2,5 cm

margen derecho-----------2,5 cm

**Otras consideraciones**.

* Los títulos correspondientes a los capítulos del texto se escribirán con mayúsculas sostenidas, precedidos del número de orden correspondiente y separadas de éste por un punto y un espacio
* La Introducción y el Capítulo I en conjunto debe ser un tercio del total del documento de tesis final
* Los títulos correspondientes a los diferentes acápites en que se divide cada capítulo se escribirán en minúsculas, precedidos del número de orden correspondiente
* El espacio final de una línea no deberá llenarse con guiones o signos, ni usarse la tecla de subrayar para separar sílabas
* Los términos que aparezca en el texto en otro idioma, así como las siglas deben escribirse en *cursiva*.
* Las siglas se comienzan a declarar la primera vez que se escribe el término a partir de la introducción, no del resumen

<Ejemplos:>

Tecnologías Semánticas (TS)

Empresa Nacional de Proyectos e Ingeniería (ENPA)

Unidad Empresarial de Base (UEB)

*Framework* para la Descripción de Recursos (*RDF*, por sus siglas en inglés)

Lenguaje de Consulta para *RDF* (*SPARQL*, por sus siglas en inglés, *Simple Protocol and RDF Query Language*)

* Los números enteros desde cero hasta nueve, cuando se usen aisladamente deberán escribirse con letras. Cuando estos números aparezcan como parte de un intervalo o de una serie, con otro u otros iguales a 10 y mayores, se escribirán con cifras. Se utilizará la coma para separar los números decimales y un espacio en blanco para separar las unidades de mil, excepto al tratarse de una fecha, en cuyo caso se escriben sin separarlos

<Ejemplos>

* + También en los años 90 surge la metodología *METHONTOLOGY* que es una de las propuestas más completas
  + El procedimiento transita por cuatro fases que rigen el proceso de gestión del conocimiento como se muestra en la Figura 8
* Ubicar correctamente las citas textuales

<Ejemplo:>

* Un indicador es una variable que describe el estado de un sistema (Heink and Kowarik, 2010).
* Autores como (Tartir et al., 2005; Sicilia et al., 2012) realizan el análisis de la calidad de una ontología basada en métricas.
* En *OntoClean* propuesta por (Guarino & Welty, 2002) y analizada posteriormente por (Tartir et al., 2010) se sigue un enfoque basado…