

Caso de Estudio: Implementación del Sistema Integral de Gestión Académica con Trazabilidad de Competencias y Concursos de Programación en la Universidad Católica

Contexto y Motivación

La Universidad Católica, una institución de renombre en el ámbito académico, ha decidido implementar un sistema web innovador para mejorar la gestión del progreso académico de sus estudiantes, con un enfoque especial en la programación competitiva. Esta iniciativa surge como respuesta a la creciente demanda de profesionales altamente capacitados en programación, un campo que ha experimentado un auge en la educación superior y que requiere un enfoque práctico y retador para el desarrollo de habilidades.

El sistema propuesto no solo permitirá a los estudiantes visualizar y gestionar su progreso académico en términos de competencias, sino que también les brindará la oportunidad de participar en maratones de programación, donde podrán medir sus habilidades contra otros estudiantes, con el objetivo de promover una cultura de competencia sana y continua mejora.

El Profesor: Motivación y Enfoque

El Profesor Diego Oliveros es un apasionado por la enseñanza de la programación competitiva. A lo largo de su carrera académica, ha observado que los estudiantes de la Universidad Católica tienen un gran potencial, pero a menudo carecen de un sistema estructurado que les permita medir su progreso y participar en desafíos reales que pongan a prueba sus habilidades en situaciones prácticas. Diego cree firmemente que, para fomentar un verdadero interés y excelencia en la programación, es necesario ofrecer a los estudiantes una plataforma que no solo les permita aprender, sino también competir, aprender de sus errores, y recibir reconocimiento por su esfuerzo.

Diego ha sido elegido para liderar la implementación de este proyecto en la universidad. En su rol de líder de semillero y coordinador del Grupo MITO, busca motivar a sus estudiantes, no solo como profesor, sino también como mentor que los acompañará en su proceso de crecimiento. Él sabe que este tipo de proyectos no solo los preparará para el mundo profesional, sino que también les permitirá desarrollar habilidades de trabajo en equipo, resolución de problemas complejos y programación bajo presión, características esenciales en la industria del software.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

"Este es un proyecto único", comenta Diego en la primera clase donde presenta el proyecto a sus estudiantes, "pero más allá de las competencias y premios, lo más importante es que aprenderán a enfrentarse a problemas reales, a superarse, y a competir al más alto nivel. Esto les permitirá ver la programación como un desafío constante y emocionante. Este sistema no es solo para ver su progreso; es un espacio para crecer, cometer errores y aprender de ellos. Y, sobre todo, para demostrar lo que son capaces de lograr."

La Competencia Real: Scalapp y la Carrera por los Fondos

En un giro emocionante, los estudiantes de la Universidad Católica no solo están desarrollando este sistema, sino que también están compitiendo contra un equipo de desarrollo profesional de la empresa Scalapp, una destacada firma de software reconocida por sus soluciones innovadoras en programación y tecnología. Scalapp ha sido invitada a ser parte del proyecto con el fin de evaluar las soluciones que los estudiantes diseñen y ofrecer financiamiento a aquellos grupos que desarrollen la mejor propuesta.

Este contexto competitivo crea un ambiente en el que los estudiantes deben enfrentarse a un reto que simula una situación real de trabajo, donde tendrán que entregar soluciones innovadoras y efectivas bajo plazos estrictos. **"Como parte de Scalapp, sabemos que este proyecto tiene un potencial increíble. Pero lo que más nos interesa es la creatividad y el enfoque único que los estudiantes puedan aportar,"** afirma Diego Oliveros, CEO de Scalapp, **"les damos una oportunidad de mostrar cómo pueden resolver problemas de la vida real, pero también ser capaces de defender sus soluciones ante un público exigente."**

Los estudiantes, divididos en equipos de cinco, tienen la misión de entregar soluciones parciales del proyecto cada dos semanas. Esta estructura de entregas parciales no solo simula un entorno de trabajo real, sino que también permite recibir retroalimentación continua, lo que les ayudará a mejorar su solución final. Cada equipo debe presentar una propuesta única, diferenciándose de las demás tanto en la implementación técnica como en la creatividad en las funcionalidades.

Los grupos que presenten las mejores soluciones a lo largo del proceso serán los que puedan acceder a los fondos otorgados por Scalapp para continuar desarrollando y perfeccionando el sistema, lo que representa una valiosa oportunidad para los estudiantes, que además de aprender, podrán contar con recursos para llevar sus ideas a otro nivel.

Desarrollo del Proyecto

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

El proyecto se desarrolla de manera iterativa, siguiendo las fases descritas en el plan. Cada semana, los estudiantes se enfrentan a nuevas funcionalidades que deben implementar, lo que les permite aplicar sus conocimientos de programación de manera práctica y efectiva. La estructura de la competencia está diseñada para que cada entrega tenga una evaluación y retroalimentación constante.

1. **Planificación y Diseño Inicial:** Los estudiantes comienzan definiendo los requisitos del sistema y diseñando la arquitectura, lo que les permite aplicar sus conocimientos de gestión de proyectos y desarrollo de software.
2. **Desarrollo del Backend y Frontend:** En las semanas siguientes, los estudiantes se sumergen en la implementación del backend y frontend, aplicando tecnologías como Node.js y React, mientras que desarrollan el sistema de trazabilidad de competencias y recompensas. Además, se encargan de la implementación de maratones de programación, que les permitirá practicar y mejorar en sus habilidades de programación competitiva.
3. **Implementación de Maratones y Competencias:** A medida que avanzan, los estudiantes se enfocan en crear el sistema de maratones de programación, una de las partes más desafiantes y emocionantes del proyecto. Aquí se enfrentan a problemas que requieren el uso de algoritmos avanzados, estructuras de datos eficientes y la habilidad de trabajar bajo presión.
4. **Gamificación y Visualización de Rankings:** El último paso consiste en la implementación de la gamificación, donde los estudiantes pueden ver su progreso y competir en rankings. Esta fase también involucra la integración de un sistema de premios y medallas que les permitirá sentir una sensación de logro y superación.

Impacto y Beneficios

Este proyecto no solo beneficia a los estudiantes al ofrecerles una plataforma para medir su progreso académico, sino que también los prepara para el mundo laboral al proporcionarles experiencia práctica en el desarrollo de software y la resolución de problemas complejos. Además, al competir contra un equipo profesional, los estudiantes aprenden a adaptarse a diferentes retos y a mejorar continuamente su trabajo, una habilidad esencial para el futuro.

En palabras de Diego: "**Quiero que cada uno de ustedes vea este proyecto como una oportunidad de demostrar lo que pueden hacer, no solo para la asignatura, sino para ustedes mismos. Este es un problema real, con desafíos reales. Cada entrega, cada maratón, cada línea de código que escriban, les ayudará a convertirse en mejores programadores.**"

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

Este enfoque de competencia y colaboración no solo fortalece el conocimiento técnico de los estudiantes, sino también su habilidad para trabajar en equipo, enfrentar la presión y adaptarse a cambios en proyectos reales.

Proyecto: Sistema Integral de Gestión Académica con Trazabilidad de Competencias y Concursos de Programación

Objetivo Principal:

Desarrollar un sistema web donde los estudiantes puedan ver su progreso académico, recibir premios por avances en competencias, participar en maratones de programación, y tener la trazabilidad de sus habilidades a través del tiempo.

Funcionalidades del Sistema:

1. Gestión de Competencias y Avances:

- El sistema debe permitir a los profesores definir competencias que los estudiantes deben alcanzar durante el curso.
- Los estudiantes deben registrar sus avances en estas competencias a medida que avanzan.
- Los profesores o el sistema otorgan "premios" o "reconocimientos" (pueden ser medallas virtuales, puntos, rangos, etc.) al avanzar en cada competencia.

2. Trazabilidad de Competencias:

- El sistema debe mostrar un gráfico o reporte que refleje la evolución de las competencias adquiridas por cada estudiante.
- Los estudiantes pueden ver en qué nivel se encuentran respecto a las competencias definidas.

3. Propuestas de Problemas para Resolución:

- El sistema debe ofrecer un espacio donde los estudiantes puedan resolver problemas prácticos relacionados con las competencias que están aprendiendo.
- Estos problemas podrían estar relacionados con programación, matemáticas, algoritmos, etc., dependiendo de la especialización de la asignatura.

4. Maratones de Programación:

- El sistema debe permitir la creación y gestión de **concursos de programación** o maratones, donde los estudiantes puedan participar para demostrar sus habilidades.
- Estos concursos pueden incluir problemas de dificultad variable y un sistema de puntuación para premiar a los ganadores.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

5. Panel de Control:

- Los estudiantes deberían tener un dashboard donde puedan ver:
 - El progreso de sus competencias.
 - Los premios o logros obtenidos.
 - Los problemas resueltos y las estadísticas de su rendimiento en las maratones de programación.
- Los profesores también deben tener un panel para gestionar competencias, problemas y concursos, además de visualizar los avances de los estudiantes.

Tecnologías Propuestas:

- **Frontend:** HTML, CSS, JavaScript, React (o Angular), para el desarrollo de interfaces interactivas.
- **Backend:** Node.js con Express.js, o Django (Python) para la lógica del servidor y manejo de la base de datos.
- **Base de Datos:** PostgreSQL o MySQL para almacenar datos de estudiantes, competencias, problemas resueltos, resultados de maratones, etc.
- **Autenticación:** JWT para la autenticación de estudiantes y profesores.
- **Sistema de Maratones:** Se puede integrar una API de programación competitiva como HackerRank o Codeforces, o desarrollar un sistema propio para resolver y puntuar problemas.
- **Premios y Gamificación:** Implementar un sistema de puntos y medallas que puedan visualizarse en el perfil del estudiante.

Fases del Proyecto:

1. **Semana 1-2: Planificación y Diseño**
 - Definición de requisitos y funcionalidades.
 - Diseño de la arquitectura del sistema.
 - Modelado de la base de datos (tablas para estudiantes, competencias, problemas, resultados, concursos, etc.).
2. **Semana 3-5: Desarrollo del Backend**
 - Crear API para gestionar estudiantes, competencias, avances, problemas y concursos.
 - Implementar sistema de autenticación para estudiantes y profesores.
 - Desarrollar la lógica de manejo de premios y avance de competencias.
3. **Semana 6-8: Desarrollo del Frontend (Interfaz de Usuario)**
 - Implementación de pantallas para la gestión de competencias, visualización de avances y resultados de concursos.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

Av. Caracas No. 46 - 72 PBX: 3277300 A.A 029832
www.ucatolica.edu.co Bogotá D.C. - Colombia

- Crear dashboards para los estudiantes y paneles de control para los profesores.
- Implementación de interacción con el backend mediante peticiones AJAX.
- 4. **Semana 9-11: Sistema de Maratones de Programación**
 - Desarrollo del sistema de maratones: creación de problemas, gestión de tiempos y puntuación.
 - Integración de la visualización de las competencias en tiempo real.
 - Añadir retroalimentación (ranking, mensajes de motivación).
- 5. **Semana 12-13: Gamificación y Recompensas**
 - Implementación de logros y premios virtuales.
 - Integrar un sistema de notificaciones para alertar sobre avances y premios obtenidos.
 - Añadir tablas de clasificación para motivar a los estudiantes.
- 6. **Semana 14-15: Pruebas y Corrección de Errores**
 - Realización de pruebas de funcionalidad (unitarias, integración, pruebas de maratones).
 - Arreglar errores detectados en el proceso de pruebas.
- 7. **Semana 16: Presentación del Proyecto y Entrega Final**
 - Preparar una presentación en la que los estudiantes muestren cómo funciona el sistema y las lecciones aprendidas durante el desarrollo.
 - Documentación del código y funcionamiento del sistema.

Evaluación del Proyecto:

- **Documentación Técnica y de Usuario:** ¿Está bien documentado el sistema?
- **Funcionalidad:** ¿Cumple el sistema con todas las funcionalidades que se definieron?
- **Interactividad y Diseño:** ¿Es fácil de usar y tiene un diseño atractivo?
- **Rendimiento:** ¿El sistema es rápido y eficiente?
- **Innovación y Creatividad:** ¿El proyecto introduce algo nuevo o tiene algún elemento creativo interesante?

Beneficios del Proyecto:

- Los estudiantes aprenderán a construir un sistema completo desde cero.
- Tendrán experiencia con **gestión de usuarios, base de datos, desarrollo de interfaces de usuario, gamificación, concursos de programación y backend.**
- El sistema tiene una aplicación real en el entorno académico, lo que le da un propósito más tangible.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

Plan de Avances por Componentes

Mes 1: Gestión de Estudiantes y Competencias

Semana 1-2: Componente 1 – Sistema de Registro y Autenticación

- **Funcionalidad:** Crear el sistema de registro de estudiantes y profesores, autenticación con usuario y contraseña.
- **Entregable:** Los estudiantes deben poder registrarse, iniciar sesión y gestionar sus credenciales de forma segura.

Semana 3-4: Componente 2 – Gestión de Competencias

- **Funcionalidad:** Implementar la posibilidad de que los profesores definan competencias, las visualicen y gestionen (crear, editar, eliminar).
 - **Entregable:** Los estudiantes pueden ver la lista de competencias definidas por el profesor, pero no aún gestionar su progreso.
-

Mes 2: Trazabilidad de Competencias y Recompensas

Semana 5-6: Componente 3 – Registro de Avances de Competencias

- **Funcionalidad:** Los estudiantes pueden registrar sus avances en las competencias. Esto podría incluir completar una actividad o alcanzar una meta.
- **Entregable:** Los estudiantes pueden ver su progreso en cada competencia, y el sistema refleja los cambios.

Semana 7-8: Componente 4 – Sistema de Recompensas (Medallas o Puntos)

- **Funcionalidad:** Implementar un sistema de recompensas automáticas al completar una competencia o alcanzar un nivel determinado. Las recompensas pueden ser medallas, puntos o rangos.
- **Entregable:** Los estudiantes reciben una notificación o medalla cuando alcanzan un avance significativo en una competencia.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

Av. Caracas No. 46 - 72 PBX: 3277300 A.A 029832
www.ucatolica.edu.co Bogotá D.C. - Colombia

Mes 3: Maratones de Programación y Problemas de Resolución

Semana 9-10: Componente 5 – Creación de Maratones de Programación

- **Funcionalidad:** Crear un sistema donde los profesores puedan configurar maratones de programación, incluyendo una lista de problemas, tiempos de inicio y finalización.
- **Entregable:** Los maratones pueden ser creados y los estudiantes pueden unirse a ellos.

Semana 11-12: Componente 6 – Resolución de Problemas y Puntuación

- **Funcionalidad:** Implementar la capacidad para que los estudiantes resuelvan problemas de programación dentro de los maratones. El sistema debe calificar las soluciones y asignar puntuaciones.
- **Entregable:** Los estudiantes resuelven problemas dentro del maratón y reciben una puntuación en función de su rendimiento.

Mes 4: Optimización, Gamificación y Pruebas

Semana 13-14: Componente 7 – Gamificación y Visualización de Rankings

- **Funcionalidad:** Introducir un sistema de rankings que muestre los estudiantes con el mejor desempeño en competencias y maratones. Además, se pueden agregar niveles o logros adicionales para fomentar la participación.
- **Entregable:** Los estudiantes pueden ver su posición en el ranking, y se visualizan logros adicionales (por ejemplo, "nivel avanzado" o "top 10").

Semana 15-16: Componente 8 – Seguridad y Optimización Final

- **Funcionalidad:** Mejorar la seguridad del sistema, optimizar la base de datos y realizar pruebas de usabilidad y rendimiento.
- **Entregable:** El sistema es más rápido, seguro y está listo para ser utilizado sin errores. Además, se realiza una revisión general para asegurar que todo funcione correctamente.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

Resumen de los Avances por Componentes:

- **Mes 1:** Gestión de usuarios (registro, login) y gestión de competencias.
- **Mes 2:** Trazabilidad de competencias (registro de avances) y sistema de recompensas (medallas, puntos).
- **Mes 3:** Maratones de programación (creación, participación) y sistema de resolución de problemas con puntuación.
- **Mes 4:** Gamificación (rankings, logros) y optimización general del sistema (seguridad, pruebas finales).

Evaluación de Progreso:

Cada dos semanas, se valida el avance de los componentes y se asegura que los entregables están funcionando correctamente antes de continuar con la siguiente fase. Esto permite que se pueda tener un seguimiento continuo del proyecto, y que cada entregable sea completamente funcional y testeado antes de continuar.

Con este plan, el desarrollo del producto es **modular y estructurado**, permitiendo un avance constante con entregas tangibles al final de cada mes, y avances parciales cada dos semanas.

Resumen de Entregas Funcionales:

- **(Mes 1):** Base del sistema de gestión de estudiantes y competencias. Los estudiantes pueden registrarse y visualizar competencias.
- **(Mes 2):** Trazabilidad de competencias, avance de estudiantes y recompensas (medallas).
- **(Mes 3):** Implementación de maratones de programación y problemas con sistema de puntuación.
- **(Mes 4):** Optimización del sistema con gamificación, seguridad y pruebas finales.

Plan de Avances por Componentes Cada 2 Semanas:

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación

Av. Caracas No. 46 - 72 PBX: 3277300 A.A 029832
www.ucatolica.edu.co Bogotá D.C. - Colombia

Cada dos semanas, se avanza en una funcionalidad o componente específico, y se realiza una validación de su funcionamiento. El proceso se verá como una sucesión de **mini-entregas** en lugar de una única entrega final.

Evaluación:

Cada **Componente** termina con una revisión de lo que se ha entregado, se validan las funcionalidades implementadas, se reciben comentarios y se ajusta el producto en la siguiente iteración. Además, al final de cada mes, se realiza una **demo** del producto completo para asegurarse de que está alineado con los requisitos iniciales.

Diego Iván Oliveros Acosta

Prof. M.Sc. Esp. Eng. Eng. Líder Semillero Mito

<https://orcid.org/0000-0002-9668-5117>

Docente, investigador, Coach Maratones de programación.

Facultad de Ingeniería - Programa Ingeniería de Sistemas y Computación