

Institución Universitaria Politécnico Grancolombiano
Escuela de Tecnologías de la Información y las Comunicaciones
Facultad de Ingeniería, Diseño e Innovación
Ingeniería de Sistemas

**Diseño e implementación de un prototipo de
plataforma de apoyo al aprendizaje de la
programación de computadores y la algoritmia
para los programas de ingeniería del Politécnico
Grancolombiano**

15 de enero de 2021

Marlon Alexander Estupiñán Galindo
Andrés Gustavo Osorio Jiménez

Asesor
Edwin Andrés Niño Velásquez

2021

Índice general

1. Introducción	3
2. Objetivos	5
2.1. Objetivo General	5
2.2. Objetivos Específicos	5
3. Planteamiento del Problema	7
3.1. Pregunta de Investigación	7
4. Justificación y Alcance	8
4.1. Justificación	8
4.2. Alcance	9
5. Marco Teórico	10
5.1. Educación virtual y herramientas del trabajo docente	10
5.2. Algoritmia	11
5.3. Programación Competitiva	11
5.4. Politécnico Grancolombiano y la algoritmia	12
5.5. Herramientas de apoyo para aprendizaje actuales	14
5.5.1. Sistemas de Gestión de Aprendizaje (LMS)	14
5.5.2. Jueces y plataformas de formación	14
6. Metodología	17
6.1. Descripción	17
6.2. Sprint	18
6.3. Fases de desarrollo	18
7. Desarrollo	19
7.1. Investigación	19

7.2. Diseño	19
7.2.1. Mockups	20
7.2.2. Modelo de datos relacional	21
7.3. Implementación	23
7.3.1. Tecnologías a usar	23
7.3.1.1. MySQL	23
7.3.1.2. Spring	23
7.3.1.3. Angular	24
7.3.2. Backend	24
7.3.3. Frontend	25
7.4. Pruebas y Despliegue	26
7.4.1. Pruebas API Rest	26
7.4.2. Despliegue	27
8. Resultados	28
8.1. Ambiente Educativo	29
8.2. Competencias	30
8.3. Entrenamiento Guiado	32
8.4. Artículos y Problemas	33
8.4.1. Creador de contenido	34
9. Conclusiones	36
Referencias	38

Capítulo 1

Introducción

El siguiente proyecto de grado presenta una solución tecnológica a un problema identificado en la Universidad Politécnico Grancolombiano acerca de la dificultad de la mayoría de los estudiantes de la facultad de ingeniería referente a los temas de algoritmia y programación. Se especificará el desarrollo de una plataforma destinada a ayudar a los estudiantes de la institución para facilitar su aprendizaje en estas áreas e incentivar y promover el estudio de las ciencias de computación y una cultura de programación competitiva.

Este proyecto surgió de una necesidad vista por parte de los docentes al ver la falta de practicidad de la plataforma actual para la gestión y calificación de programas implementados por los estudiantes. Se propuso el desarrollo de una plataforma web que busca solucionar este problema a partir de una implementación de un sistema de gestión de aprendizaje usando una perspectiva presente en los entornos de programación competitiva. Esta solución no busca reemplazar el sistema actual de la universidad, sino servir como complemento y punto de partida para mejorar el rendimiento de los estudiantes en este tipo de áreas que no son la especialidad de una plataforma de aprendizaje común. Además de esto, el proyecto tiene como fin funcionar como gestor y juez de competencias de programación competitiva para el Politécnico Grancolombiano.

Para la realización de este trabajo de grado se hizo uso de todos los conocimientos adquiridos a la carrera de Ingeniería de Sistemas, referente a la gestión de proyectos, metodologías de desarrollo y programación, diseño y arquitectura de software.

Concerniente a los temas del documento, se muestran los objetivos generales y específicos identificados, planteamiento del problema, justificación y alcance del proyecto, un marco teórico referente a los temas a tratar, la metodología usada para la construcción de la aplicación, y todo lo referente al desarrollo y resultados de esta.

Capítulo 2

Objetivos

2.1. Objetivo General

Diseñar e implementar un prototipo de plataforma web enfocada en el aprendizaje de programación de computadores, en un contexto de programación competitiva, dirigida a los cursos del programa de Ingeniería de Sistemas e Ingeniería de Software del Politécnico Grancolombiano.

2.2. Objetivos Específicos

- Evaluar la oferta actual de herramientas y plataformas para el apoyo del proceso de aprendizaje de temas relacionados con la algoritmia y la programación de computadores.
- Diseñar una plataforma web, centrada en programación competitiva, como herramienta de apoyo a las asignaturas asociadas con algoritmia y programación de computadores del Politécnico Grancolombiano.
- Implementar una plataforma que permita la realización de competencias de programación para fomentar el desarrollo de habilidades en resolución de problemas, diseño de algoritmos y programación.

- Implementar una plataforma que permita la administración, visualización y evaluación de cursos relacionados con algoritmia y programación de computadores.

Capítulo 3

Planteamiento del Problema

En el Politécnico Grancolombiano, se ha visto frecuentemente una dificultad entre los estudiantes de ingeniería a la hora de ser introducidos ante temas referentes al proceso de programación. Esto se debe a que estas asignaturas no hacen parte de la malla curricular de la gran mayoría de los colegios de Colombia, por lo que el primer acercamiento de estos estudiantes se encuentra en los primeros semestres de la carrera. Estas dificultades se ven reflejadas en la falta de comprensión de lógica y sintaxis de los lenguajes de programación.

3.1. Pregunta de Investigación

¿Necesita el Politécnico Grancolombiano una plataforma independiente del campus virtual o el moodle, para sus programas de ingeniería de sistemas y software, que tenga la finalidad de complementar y reforzar los temas dados en clase, así como proveer un método alternativo de evaluación de competencias?

Capítulo 4

Justificación y Alcance

4.1. Justificación

Este proyecto tiene la finalidad de desarrollar un prototipo que sirva como una plataforma independiente a las ya existentes en el Politécnico Grancolombiano. Esto, a raíz de que moodle es un sistema para educación general, lo que limita el desarrollo y la evaluación de las competencias específicas de cada materia. En el caso de las materias de programación y afines, se ven drásticamente limitadas ya que esta plataforma no provee un sistema simple para calificar código fuente. En el caso del docente, éste se ve obligado a descargar los archivos de cada estudiante, compilarlos y evaluarlos él mismo, por lo que se vuelve una tarea muy tediosa y consume mucho tiempo.

Nuestra propuesta se trata de crear una plataforma independiente que sirva para que los docentes puedan hacer petición de código a través de una serie de problemas que tiene sus casos de prueba ya estandarizados, El profesor recibirá las soluciones de sus estudiantes ya evaluados a partir de estos casos, agilizando las tareas de revisión y calificación. De igual forma, la plataforma permitirá al profesor visualizar y probar el código en línea sin necesidad de descargas.

La plataforma contará con un repositorio propio de artículos que complementan los recursos dispuestos en las clases y facilitan el proceso de aprendizaje de los estudiantes.

Por otro lado, la aplicación busca crear en los estudiantes interés en la programación

competitiva. Esto se logra mediante una serie de competencias que buscan que el estudiante aplique sus conocimientos en algoritmos y programación en un conjunto de problemas, compitiendo con sus pares.

4.2. Alcance

El proyecto tiene como propósito la creación de un prototipo de un sistema que combine los elementos esenciales de un juez en línea y un sistema de gestión del aprendizaje enfocado a la programación, junto con un gestor de repositorios de artículos matemáticos y algorítmicos y problemas de computación para el Politécnico Grancolombiano. Cabe resaltar que no se implementará un servidor de evaluación para las soluciones a problemas enviadas por los usuarios.

Como parte del proyecto, se realizó una investigación de distintas plataformas que facilitan a los profesores calificar las habilidades de programación y aplicación de algoritmos, herramientas para realizar competencias de programación competitiva y portales de información con el objetivo de enseñar y solucionar dudas relevantes para la algoritmia, las estructuras de datos y los lenguajes de programación. A su vez se realizó un sistema que incluye un modelo de datos relacional, una aplicación web que simula el funcionamiento de un juez en línea de programación competitiva, un portal de gestión y visor de problemas y artículos con formato dado por markdown, un apartado en donde docentes pueden compartir artículos, asignar y calificar distintas tareas a estudiantes de una clase, espacios por clase en donde los estudiantes pueden debatir distintos temas y un espacio donde un creador de contenido es capaz de administrar cursos, artículos, problemas y competencias, todo esto desplegado en un servidor web para su visualización.

Capítulo 5

Marco Teórico

5.1. Educación virtual y herramientas del trabajo docente

A mediados de 1990, una vez los servicios de mensajería y páginas web dejaron de ser exclusivamente proyectos militares y se abrieron al público general (dando a conocer lo que se conoce hoy como Internet) (Rahman, s.f.), se popularizaron los sistemas de educación a distancia, específicamente la enseñanza en línea, convirtiéndose en un área de aplicación fundamental dentro de lo que conocemos como las Tecnologías de la Información y Comunicación, TIC.

El sistema de enseñanza virtual, como su nombre lo indica, es aquel en donde los estudiantes no necesitan asistir físicamente a un lugar en concreto, y en cambio la asistencia es proporcionada por un docente de manera virtual en conjunto con herramientas como aplicaciones o plataformas web.

Es ineludible entonces pensar en las ventajas que trajo la educación en línea (R.S/E.C, 2011) como la flexibilidad al no tener horarios o espacios físicos específicos dando facilidad al estudiante de encontrar momentos adecuados para estudiar a lo largo del día, que de otra manera lo habría hecho depender de los horarios y espacios físicos que ofrece la educación tradicional. Otra de las ventajas es el aumento de participación por parte de los estudiantes, dado que se facilitan espacios como foros, en donde tienen un apartado para solucionar o ayudar a solucionar dudas, además de esto aumenta la disciplina por parte del estudiante, dado que debe tener cierto grado de compromiso.

El Politécnico Grancolombiano, pionera en el ámbito de educación virtual (El Tiempo, 2020) y tiene la visión de ser la universidad que brinda más y las mejores alternativas educativas (Politécnico Grancolombiano, 2020d), hace uso de varias herramientas digitales para la enseñanza tales como CANVAS LMS para sesiones completamente virtuales o MOODLE como complemento de la modalidad presencial. Estas dos herramientas son sistemas de gestión de aprendizaje (Learning Management System o LMS), los cuales son software que tienen como objetivos principales la administración de cursos, creación y seguimiento de tareas por estudiante y socialización entre estudiantes y docentes (Iqbal y Qureshi, 2011). Otros ejemplos conocidos sobre estas herramientas son: Blackboard, Google Classroom.

5.2. Algoritmia

Un algoritmo es un conjunto de instrucciones finitas, ordenadas que permiten solucionar problemas, realizar cálculos o procesar datos. El funcionamiento de un algoritmo se da a partir de un estado inicial, cero o varios datos de entrada son procesados con las instrucciones y se obtiene uno o varios datos que son la solución.

Los algoritmos pueden ser expresados de varias maneras, puede ser en lenguaje natural, diagramas de flujo pseudocódigo o escrito sobre un lenguaje de programación. (Knuth, 2011) Decimos que un algoritmo es correcto y que soluciona el problema computacional relacionado si por cada una de las entradas posibles que se le proporcionen, este se detiene con la solución de datos esperada (Cormen, 2003)

Una de las características más importantes para tener en cuenta en el momento de analizar un algoritmo es su eficiencia, la cual es atribuida a su duración de ejecución y el consumo de memoria esperada. (Cormen, 2003)

5.3. Programación Competitiva

Al relacionar los conceptos de programación y concurso se crea el concepto de programación competitiva. Considerado un deporte mental, en la programación competitiva se

reúnen varios participantes cuyo objetivo es resolver la mayor cantidad de problemas lógicos o matemáticos, mediante la creación de programas de computadora (algoritmos) capaces de solucionar cada uno de estos. La solución propuesta por el participante, llegada a considerarse correcta, es juzgada según el tiempo que haya tomado en ejecutarse y la cantidad de memoria que es usada. (Lin, 2020)

Varios de estos concursos son normalmente patrocinados por grandes compañías como Google (Google, 2020) y Facebook (Facebook, 2020), y son un factor importante para la contratación en compañías como estas (Adityanathloyola, 2020).

Algunas de las ventajas de participar en esta clase de competencias es la obtención de nuevos conocimientos en algoritmos y estructuras de datos, conceptos importantes a ser empleados en campos como la inteligencia artificial, modelado matemático y simulación. Igualmente se puede reforzar la habilidad de trabajar bajo presión y límites de tiempo. Al ser competencias reconocidas a nivel mundial, a la hora de participar es posible practicar networking con distintos participantes. (Uturunco, 2019)

En el ámbito universitario, existe la Competición Internacional Universitaria ACM de Programación (ICPC-ACM), un evento patrocinado por la Association for Computing Machinery (ACM) en donde participan universidades de todo el mundo. En Colombia, como parte oficial de ICPC, se realizan los eventos de Colombian Collegiate Programming League (CCPL) en donde participan las universidades del país en su liga interna, como a su vez existe la Maratón Nacional de Programación ACIS/REDIS que da clasificatoria a los estudiantes que participarán en la Maratón Regional Latinoamericana ACM/ICPC para posteriormente ir a la final mundial de ICPC. (Politécnico Grancolombiano, 2020c) (ACIS, 2020)

5.4. Politécnico Grancolombiano y la algoritmia

Cómo el Politécnico Grancolombiano señala, “El país atraviesa por momentos trascendentales que exigen un gran número de ingenieros que sean capaces de automatizar servicios en salud, comercio, gobierno, banca, educación y otros grandes sectores, por lo que este programa está diseñado para responder a las necesidades del sector laboral, formando profesionales altamente competentes en la disciplina.” (Politécnico Grancolombiano, 2020b)

No es excepción con los ingenieros de Software y Sistemas, los cuales deben ser capaces en su campo de desarrollar proyectos soportados en soluciones de software y administrar tecnologías de la información (Politécnico Grancolombiano, 2020a).

El concepto de la algoritmia en la institución es vista por primera vez cuando los docentes enseñan a los estudiantes temas generales como la lógica en materias como pensamiento algorítmico donde el estudiante podrá empezar a desarrollar la capacidad de entender, ejecutar, evaluar y crear algoritmos. Posteriormente en el curso Programación de Computadores el estudiante escribirá en un lenguaje de programación seleccionado por el curso sus primeros algoritmos y aprenderá a usar las herramientas que el computador puede proporcionarle. En el curso de estructura de datos se introducirán distintas y eficientes formas de administrar, organizar y almacenar datos en un computador de manera general de tal forma que pueda ser usado en distintos lenguajes de programación. Como se puede ver, los programas de ingeniería de software e ingeniería de sistemas comparten varios cursos cuya base es la algoritmia.

Algunos de estos cursos son:

- Pensamiento Algorítmico
- Programación de Computadores
- Paradigmas de programación.
- Autómatas, Gramáticas y Lenguajes
- Estructura de Datos
- Análisis y Verificación de Algoritmos
- Elementos de Teoría de la Computación.
- Diseño de Algoritmos
- Herramientas de Lógica Computacional

Alternando a los cursos de cada carrera, en el Politécnico Grancolombiano se encuentran grupos de donde se enfocan en enfrentarse a los retos de programación competitiva, en donde se forjan equipos para presentarse en las maratones de programación nacionales y regionales, donde la institución ha sido sede cinco veces consecutivas (Politécnico Grancolombiano, 2019), para clasificar y competir en la final mundial de ICPC-ACM.

5.5. Herramientas de apoyo para aprendizaje actuales

5.5.1. Sistemas de Gestión de Aprendizaje (LMS)

Cómo se mencionó anteriormente, los LMS son software que se enfocan en la organización y gestión de cursos online. Para tener un ambiente similar a este, se tuvo que entender que en sus características fundamentales envuelve todos los conceptos claves de una clase tales como los temas a ver, recursos y anexos compartidos, la administración de los participantes, gestión en la asignación y entrega de tareas y foros de discusión. Todo esto se reúne al haberse realizado una investigación en las plataformas proveídas por la institución, Moodle y Canvas LMS y la plataforma gratuita perteneciente a G Suite, Google Classroom.

5.5.2. Jueces y plataformas de formación

Las competencias de programación competitiva, se realizan de manera común sobre un servidor en internet o alojado de manera local en la red (si este se realiza de manera local), existen software como BOCA (Campos y Ferreira, 2004) que sirven para crear un ambiente competitivo entre los participantes, en donde cada usuario tiene acceso al marcador de la competencia y veredictos de esta, de tal manera que el envío de soluciones, juzgamiento y clarificaciones se pueda realizar de una forma organizada y automatizada.

Paralelo a esto, existen plataformas en donde los usuarios pueden organizar, solucionar y compartir competencias de programación con cualquier otro usuario registrado en esta página. Investigando estas podemos encontrar que cada una tiene ciertas características a tomar en cuenta:

- Codeforces: Cómo su creador Mike Mirzayanov, lo indica, esta plataforma fue creada con la visión de ser un espacio para programadores en donde se puedan relacionar en distintos temas de discusión, creación y lectura de blogs. Los problemas que presenta esta plataforma tratan de ser concisos haciendo uso de lenguaje matemático. Los intentos de solución de un problema (código fuente) pueden ser vistos por cualquier usuario de la plataforma. (Mirzayanov, 2010).
- UVa OJ e ICPC Live Archive: Desde la Universidad de Valladolid (UVa) se desarrolló

uno de los más grandes jueces en línea automático, conteniendo un repositorio con más de 4300 problemas. Este juez se ha usado para varias competencias de ICPC, llegando a alojar en 2007 un nuevo servidor en la Universidad Baylor donde se cargan los problemas regionales y mundiales de ACM-ICPC. (Revilla, Manzoor, y Liu, 2008)

- VJudge: Este portal, a diferencia de los demás, está enfocado a la creación de competencias por cualquier usuario, reuniendo varios repositorios de problemas tales como CodeChef, HackerRank, Topcoder, Codeforces y UVa OJ. (Han, 2010)

Características para resaltar de estos proyectos es la manera de categorizar cada tema de los problemas por medio de etiquetas, el uso del lenguaje matemático y la posibilidad de ver el código fuente de los intentos de solución. Se logra encontrar desventajas como que el creador de competencias no puede elegir los casos de prueba con los que el problema será evaluado.

Nombrando otras plataformas que sirven como jueces en línea, existe HackerRank cuyo enfoque es servir como plataforma de reclutamiento de programadores (Kosner, 2014), CodeChef mantiene competencias de gran duración cada mes con propósito educativo, TopCoder el cual tiene el propósito de mostrar de manera general ciertos temas de algoritmia, tanto en forma de problemas de computación o artículos, lo que nos lleva al apartado de plataformas de formación (Bloomfield y Sotomayor, 2014).

- TopCoder: Presenta un conjunto de artículos sobre matemática y algoritmos hechos por administradores de la plataforma, los cuales son presentados en un formato unificado incluyendo gráficos y segmentos de código fuente para una mejor exposición de los temas.
- Geeks For Geeks: Presenta artículos creados por administradores y usuarios de la comunidad, centrándose en la solución de problemas específicos (Ya sea por algoritmo, teoría o lenguaje de programación) dando una explicación (Jain, 2016).
- Leetcode: Presenta un conjunto de diversos problemas, orientados principalmente a entrevistas. Similar a TopCoder, tiene un gran repositorio de artículos escrito por creadores de contenido manteniendo un solo tipo de formato. (LeetCode, 2020)

Uno de los puntos más importantes a destacar de estas aplicaciones es que a diferencia de los problemas, existe la carencia de categorización de los artículos, sin embargo, es fácil

apreciar el contenido mostrado en las plataformas LeetCode y TopCoder, ya que, al usar un formato unificado es más cómodo e intuitivo para el usuario obtener la información que el artículo quiere mostrar.

Capítulo 6

Metodología

A continuación, se describe la metodología usada, en qué consistió y cómo se adaptó ésta a lo largo del desarrollo del proyecto, los motivos principales y finalmente una descripción breve de cada una de las fases conformaron el proyecto.

6.1. Descripción

Se describirá la condición metodológica en el que está basado el proceso de desarrollo del proyecto descrito anteriormente, bajo el carácter de desarrollo de software, las fases que conformaron el proyecto y el marco metodológico usado para planificar, estructurar, controlar y posteriormente ejecutar el proceso de desarrollo.

Se buscó incorporar una metodología ágil como marco de trabajo, así pues gracias a la facilidad de adaptación de la metodología Scrum fue tomada como base para sacar de esta los conceptos más relevantes y adaptarlos a las necesidades del proyecto, estos son entrega de valor constante, sprint y retrospectivas, adicionalmente se usaron historias de usuario como marco de trabajo.

6.2. Sprint

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto (Huambachano, 2017). Este consiste en pequeñas iteraciones llamadas “sprint” de dos semanas en la que se produce un resultado completo que aumenta el valor al producto.

Así pues, Scrum está diseñado para grupos de desarrollo pequeños por lo que se ajusta perfectamente a las necesidades del proyecto, su simplicidad deja que se pueda dar más atención al desarrollo aumentando así el valor de la aplicación final. Para este trabajo se definió el sprint de una semana, esto permite que semanalmente se esté en constante revisión lo que facilita la búsqueda de fallos además de dar la flexibilidad de hacer cambios que fueran necesarios.

6.3. Fases de desarrollo

- Fase de investigación: Esta fase consiste en la búsqueda de tecnologías que sirvan para el desarrollo del prototipo, esto consiste en ver su curva de aprendizaje, la compatibilidad con el proyecto y la escalabilidad que ofrezca.
- Fase de diseño: En esta fase se hace la planeación de la aplicación, esto consiste en hacer los mockups que son los diseños que sirven de base del componente visual del prototipo, también se diseña el modelo de datos que es la base en la cual se soporta toda la lógica de la aplicación.
- Fase de implementación: Todo el desarrollo de la aplicación recae en esta fase, se modela en un motor de base de datos el modelo de datos y se trabaja simultáneamente en el desarrollo del back-end creando una serie de microservicios REST, junto con un front-end que provee una interfaz capaz de permitir al usuario comunicarse con el back-end.
- Fase de revisión y despliegue: Esta fase comprende la serie de pruebas que aseguran el buen funcionamiento del sistema.

Capítulo 7

Desarrollo

En este capítulo se mostrará la planeación que se llevó a cabo para la realización del proyecto, esta consiste en la realización de un esquema por el cual se va a diseñar el componente visual y la construcción de un modelo de datos sobre el que se construirá toda la aplicación.

7.1. Investigación

Dedicamos a hacer la investigación correspondiente a las herramientas existentes sobre los sistemas de gestión de aprendizaje usados por el Politécnico Grancolombiano y otros más usados por instituciones ajenas vistas a lo largo de este documento y ya que nuestro enfoque es realizar una plataforma similar a estas enfocado a un ambiente de programación competitiva, se realizó la investigación sobre plataformas de formación en este ámbito para identificar lo mejor de estas y modelar una plataforma que tenga aspectos de LMS y juez online.

7.2. Diseño

Para la realización de un diseño inicial de la aplicación, se hizo un análisis de las historias de usuario con el cual poder realizar un bosquejo de cómo debería ser el componente visual de tal forma que cumpla con las necesidades propuestas. También gracias a las historias

de usuario podremos planificar el modelo de datos con el que se registrará toda la lógica del aplicativo.

7.2.1. Mockups

Estos fueron pensados en una plataforma web se compone de dos módulos principales, el primero consiste en un ambiente educativo el cual que está conformado por espacios en donde el estudiante puede participar en foros y tener acceso a contenido como artículos y problemas computacionales que el profesor presentó con anterioridad, el segundo consiste en un módulo destinado a programación competitiva, orientada a maratones de programación, en donde se puede ver y participar en competencias, tener un acceso a los problemas, hacer envíos y estar al tanto de la estado actual de una competencia. Estos módulos contienen componentes de ambiente educativo para que de esta manera se sea intuitivo al estudiante moverse a través de la plataforma. Las figuras 7.1 y 7.2 presentan ejemplos de los mockups construidos.

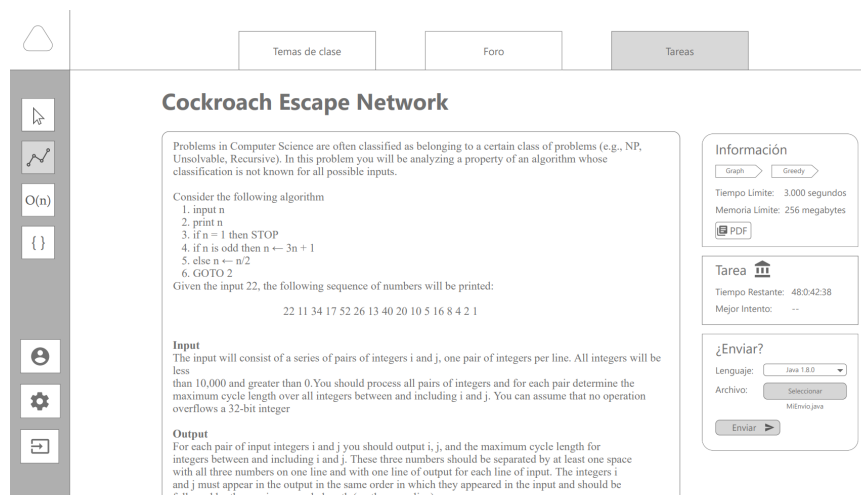


Figura 7.1: Mockup de la visualización de un problema.

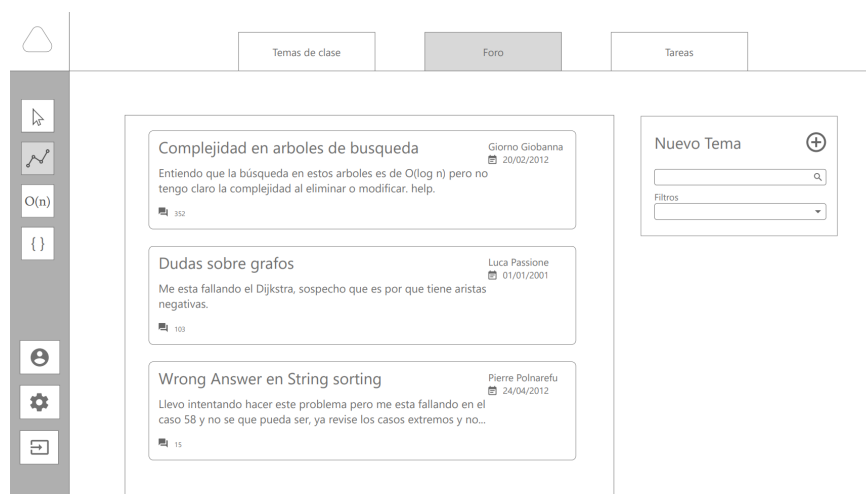


Figura 7.2: Mockup del foro de una clase.

Todos mockups se encuentran en el Anexo 1.

7.2.2. Modelo de datos relacional

Para que el funcionamiento de la aplicación fuera el adecuado, fue necesario tener una base lógica sólida, así que el diseño del modelo de datos se manejó con la misma propuesta que se usó para el diseño de los mockups, esto es dividirlo en dos partes, una que se encargará de todo el entorno de competencia y el otro fuera la parte estudiantil.

Para lograr manejar toda la lógica de una competencia primero se hizo una investigación sobre cómo los jueces más famosos manejan sus datos (Codeforces, UVa OJ, vjudge, Spoj) y se hizo una propuesta que lograra mantener esa información de manera similar a estas páginas, las competencias se mantienen principalmente por tres entidades, “Problem”, “Contest” y “Submit”. “Problem” mantiene la información del problema, esto consiste simplísimamente en el enunciado del problema, el tiempo límite, la memoria límite y sus casos de prueba, para los casos de prueba se decidió hacer una entidad externa, esto con el fin de poderlo manipular fácilmente ya que los casos van a ser individuales y por lo tanto añadir o eliminar serán las acciones más recurrentes, además permite una pequeña capa de personalización para los casos que se desea ser evaluados. Luego está la entidad “Contest”, la cual se encarga de mantener la información de cada competencia, dicta si es privada, su tiempo de inicio y su duración. Para crear la relación del listado de problemas pertenecientes a una competencia

se hizo uso de una entidad mediadora que además de hacer la relación muchos a muchos entre estas dos entidades, lleva consigo una máscara de bits que va a servir para hacer el filtro de los casos de prueba de este problema que van a ser evaluados por el juez. La entidad “Submit” es la que mantiene la información del envío que hace el usuario, la cual consiste en el código fuente, el lenguaje de programación, y también toda información correspondiente al veredicto, el cual son los casos pasados, el tiempo usado, la memoria usada y el veredicto en sí.

En el apartado estudiantil consiste en tres entidades que almacenan todo lo que es el “salón de clase”, estos son “Course”, “Section” y “Post”. La entidad “Course” es donde se guarda toda la información de las clases, los estudiantes inscritos, el profesor a cargo y una competencia asociada que es donde el docente puede poner tareas, cada curso posee una serie de secciones que son zonas en donde el profesor puede seleccionar una serie de artículos y problemas acerca de un tema específico, esto es manejado por la entidad “Section”, y por último a cada clase se le asocia un foro el cual busca ser el medio de comunicación entre los miembros de la clase, la información de cada pregunta o respuesta en el foro se almacena en la entidad “Post”. La figura 7.3 muestra el modelo de datos de la aplicación.

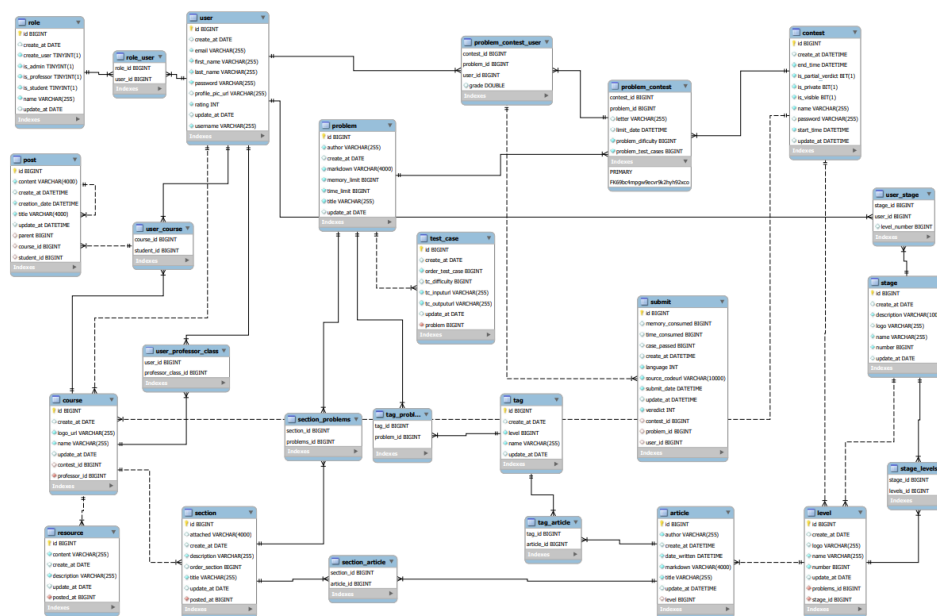


Figura 7.3: Esquema del modelo de datos.

El esquema detallado se encuentra en el Anexo 2.

7.3. Implementación

A continuación vamos describir todo el proceso de creación de la aplicación, describiendo las tecnologías que fueron usadas junto con todo el proceso creativo realizado para construir la plataforma web como el sistema de servicios y la base de datos.

7.3.1. Tecnologías a usar

7.3.1.1. MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto creado por Oracle Corporation, dado que este gestor de datos tiene la capacidad para soportar una gran cantidad de lectura de registros, facilidad de integración, baja probabilidad de corrupción de datos, alta conectividad, velocidad y seguridad lo hace un candidato idóneo para servir de gestor de base de datos dada la naturaleza del proyecto planteado.

7.3.1.2. Spring

Spring es un framework para el desarrollo de aplicaciones de código abierto, multiplataforma. Spring posee un conjunto de módulos en los que cada uno provee una variedad de servicios. Por propósitos del aplicativo solo se hizo uso de dos módulos.

El primer módulo es el de Acceso a datos, este trabaja usando Hibernate (Linwood y Minter, 2010) para mapear la base de datos relacional a una base de datos orientada a objetos virtual a través de técnicas de Mapeo Objeto-Relacional (ORM por sus siglas en inglés) (Hibernate, 2011), esto con el fin de trabajar haciendo uso de la Programación Orientada a Objetos (OOP) al interactuar con la base de datos. El segundo módulo es el de Modelo Vista Controlador (Spring MVC) (Burbeck, 1992), este módulo se encarga de proveer herramientas para montar servicios REST (World Wide Web Consortium, 2016) a partir de protocolos basados en HTTP (Gettys y cols., 1999) y servlets.

7.3.1.3. Angular

Angular es un framework para el desarrollo de aplicaciones web, el cual esta basado en HTML, CSS y javascript. Angular es un framework de código abierto desarrollado por Google. (Precht, 2016). Angular provee unas clases llamadas componentes que enlazan un template HTML con unas variables y funciones creadas en Typescript, de esta manera permite mantener una vista dinámica pero simple de mantener.

7.3.2. Backend

La parte lógica es un componente esencial para el funcionamiento de la aplicación, esta es la que va a mantener todo el sistema estable y debe ser lo suficientemente independiente para resolver cualquier excepción que pueda surgir, para el desarrollo de este sistema se hizo uso del entorno de trabajo Spring, esta herramienta es la que nos permite hacer de puente entre la base de datos y el cliente.

Primero se diseñó todo el modelo de datos a través de clases de java, y gracias al componente de ORM de Spring, estas clases, junto con sus relaciones se traducen a SQL para formar así la base de datos. Así mismo Spring provee una serie de repositorios CRUD que nos permite manipular la base de datos sin necesidad de usar SQL. Después de tener el modelo de datos construido, se desarrolló una serie de servicios REST que sirven como canal directo para que el cliente modifique o consulte en la base de datos. Estos servicios en su mayoría son de consulta, en el que destacan las consultas con paginado las cuales son ampliamente usadas durante la aplicación, por otro lado existe una línea de servicios para la creación, modificación y eliminación, diseñado para los roles de creador de contenido, administrador y docente.

El núcleo de la aplicación radica en la funcionalidad de las competencias, para esto es necesario un sistema que pueda recibir una petición con un envío, resolverla y actualizar en la base de datos el nuevo veredicto, para esto se usó el enfoque de un sistema de mensajería por lo que se hizo uso de Java Message Service (Curry, 2004) para el manejo de la cola de peticiones, esta interfaz está diseñada para el uso en mensajería instantánea, sin embargo la manera en la que se reciben y responden los mensajes funciona de forma similar a la que lo se busca en la cola de envíos.

7.3.3. Frontend

El componente visual es una parte esencial del proyecto puesto que es el medio por el cual el usuario va a estar interactuando con la aplicación durante todo su ciclo de vida, de esta manera se requiere que el componente gráfico siga unos lineamientos que sean familiares a un entorno educativo y así facilitar la entrada de los estudiantes y profesores al sistema.

Para la sección de clases se buscó un ambiente similar a moodle, con esto en cada una de estas resalta las secciones propuestas por el profesor de tal forma que el estudiante siempre tenga en primer plano lo que el docente considera que debe prestar atención, también hay una serie de accesos a las tareas y a los foros que intentan que el estudiante esté siempre consciente de los trabajos propuestos y en constante comunicación con la clase por medio de preguntas en el foro. El docente posee la misma distribución visual que un estudiante, con la diferencia de tener la capacidad de editar y reordenar el contenido, para darle la libertad de personalizar la clase de la forma en que él considere adecuada. Para este componente se usó las tarjetas (cards) como principal recurso visual, esto con el objetivo de darle un estilo más moderno al salón de clase (classroom) habitual en este tipo de sitios.

En el apartado de competencias se mantuvo la línea visual estándar que se usa en los sitios de programación competitiva. Los problemas y artículos son enlistados, y cada uno tiene una serie de etiquetas (tags) dispuestos de una manera agradable con el fin que el usuario pueda fácilmente distinguir un problema (o artículo) de otro basado en su contenido sin la necesidad de acceder a él. Los problemas y los artículos se visualizan usando la herramienta de formateo de texto markdown, de esta forma todos los problemas (y artículos) tienen el mismo formato y la misma estructura, lo que facilita al usuario la lectura y le da uniformidad a la aplicación. La estructura de una competencia mantiene el aspecto de un juez en línea regular, el cual se compone de una pantalla en donde se distingue una lista de los problemas a resolver, un visor para cada problema y un tablero en donde se puede visualizar el estado de todos los participantes en la competencia, y siempre en la parte superior el tiempo transcurrido de la competencia, desde esta pantalla el usuario puede realizar todas las acciones necesarias para participar tales como enviar y consultar su posición respecto a otros participantes. Todas las competencias se enlistan y se distinguen según su estado, estos son si la competencia está actualmente corriendo, si ya acabó o si es una competencia que aún no inicia.

Para la sección de entrenamiento guiado se buscó un entorno más gamificado, dando

principal atención a los iconos y a la descriptibilidad de cada botón, de esta forma el usuario tiene en todo momento claridad sobre lo que se desarrolla en cada nivel. Los niveles tienen una estructura visual diferente a la de una competencia a pesar de ser en esencia lo mismo, la estructura de un nivel consiste en una serie de problemas y artículos en donde conforme el usuario va resolviendo problemas se le van desbloqueando otros, aquí se busca fomentar la individualidad de cada usuario quitando el componente competitivo y dando principal atención al entrenamiento y fortalecimiento de los temas tratados en cada nivel.

Ya que la aplicación está pensada principalmente para usuarios con conocimientos informáticos, se eligió una paleta de colores inspirados en esta “cultura informática”, de esta forma se hizo uso de un tema oscuro, utilizando los grises para crear contrastes entre los paneles y el fondo, y se eligió el verde como color principal de la aplicación para dar el aspecto de una consola de comandos, estas elecciones se hicieron siempre buscando mantener un aspecto moderno y agradable para el usuario final.

7.4. Pruebas y Despliegue

7.4.1. Pruebas API Rest

Con el fin de mantener un desarrollo libre de errores se diseñaron una serie de pruebas unitarias para asegurar que todos los servicios, de los que hace uso la aplicación, funcionen de la manera esperada, para realizar estas pruebas se hizo uso de una herramienta llamada Postman. Postman es una plataforma para el diseño de APIs, desde esta se pueden hacer llamadas https y obtener un detallado reporte de las respuestas dadas por el servidor, así mismo se puede configurar una respuesta esperada para así contrastar si el servicio se ejecutó correctamente.

Por medio de Postman se diseñaron más de cincuenta pruebas para probar todo el listado de servicios ofrecidos, estas pruebas consistieron en probar todos los servicios CRUD. La figura 8.8 muestra un ejemplo de las pruebas API hechas en Postman.

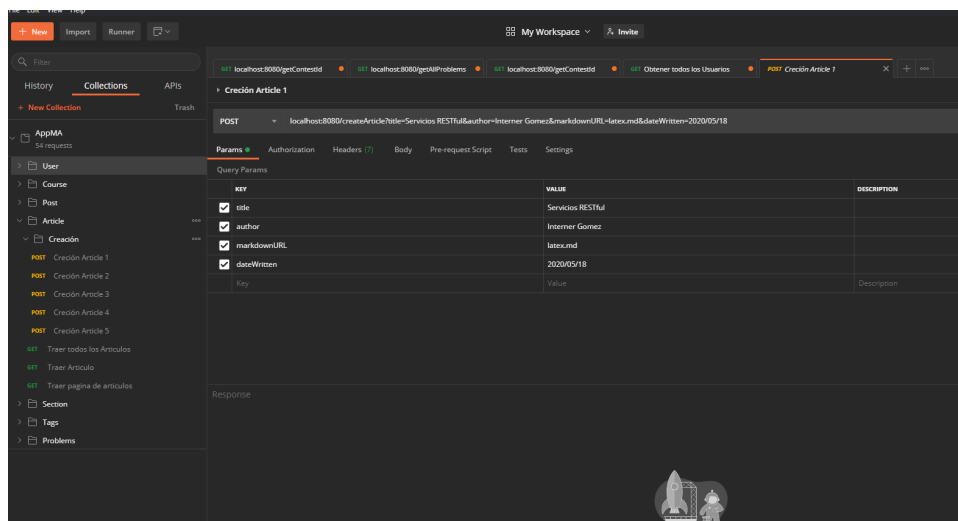


Figura 7.4: Ejemplo de prueba REST en Postman

La lista de pruebas completa se encuentran en los Anexo 3.

7.4.2. Despliegue

Con el fin de simular un ambiente de producción se hizo el montaje de la capa de Frontend sobre la plataforma GitHub, el cual nos ofrece un mantenimiento y ejecución de esta sobre un repositorio creado, y a través de la plataforma Heroku se montó el servidor para mantener la capa de Backend referente a la lógica del negocio, los servicios y la base de datos.

Este es un despliegue de prueba para mostrar un prototipo ya que estas plataformas se encuentran limitadas por uso, por lo que el proyecto no puede ser usado de manera oficial.

Capítulo 8

Resultados

Una vez completada la aplicación, y relacionando los objetivos, es posible evidenciar de forma específica el cumplimiento de estos ya que existen grandes núcleos en los que se basa la aplicación.

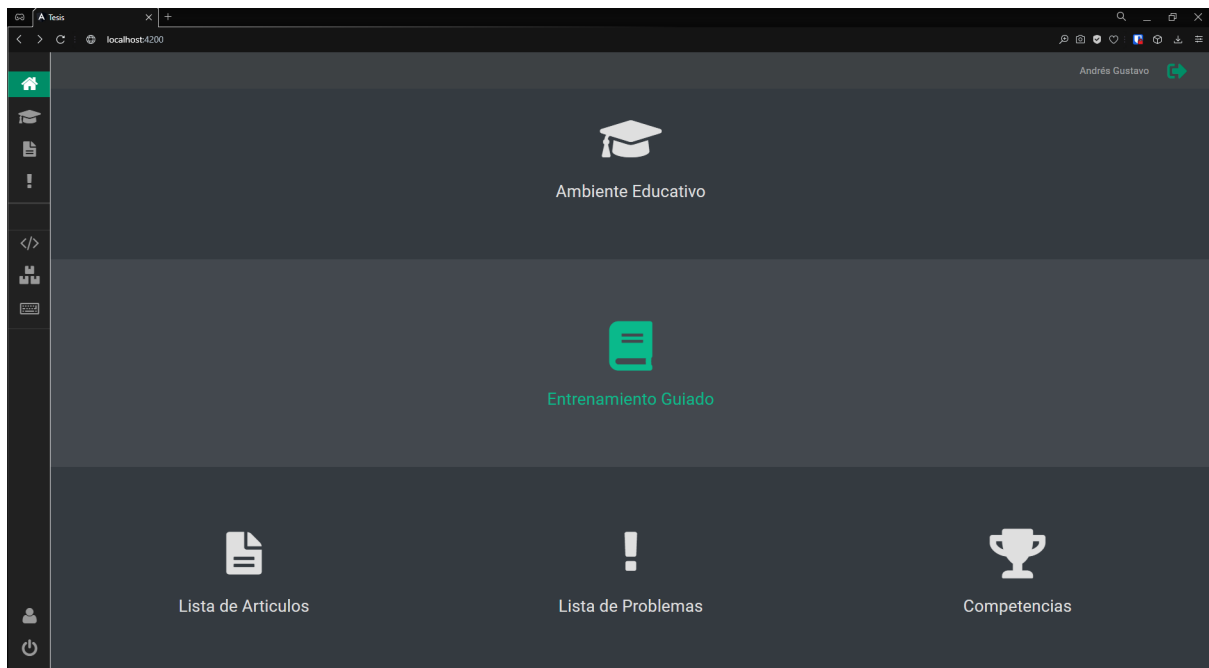


Figura 8.1: Pantalla principal de la plataforma

Podemos ver entonces que hay tres componentes fundamentales del sistema, el Ambiente Educativo, un apartado para Competencias y el Entrenamiento Guiado para usuarios menos

experimentados.

8.1. Ambiente Educativo

Aquí podemos encontrar un sistema de gestión de aprendizaje (LMS), en donde los estudiantes y los tutores de distintos cursos pueden interactuar mediante un foro de discusión, sugerencias de artículos y actividades y la asignación de problemas de programación competitiva como tareas.

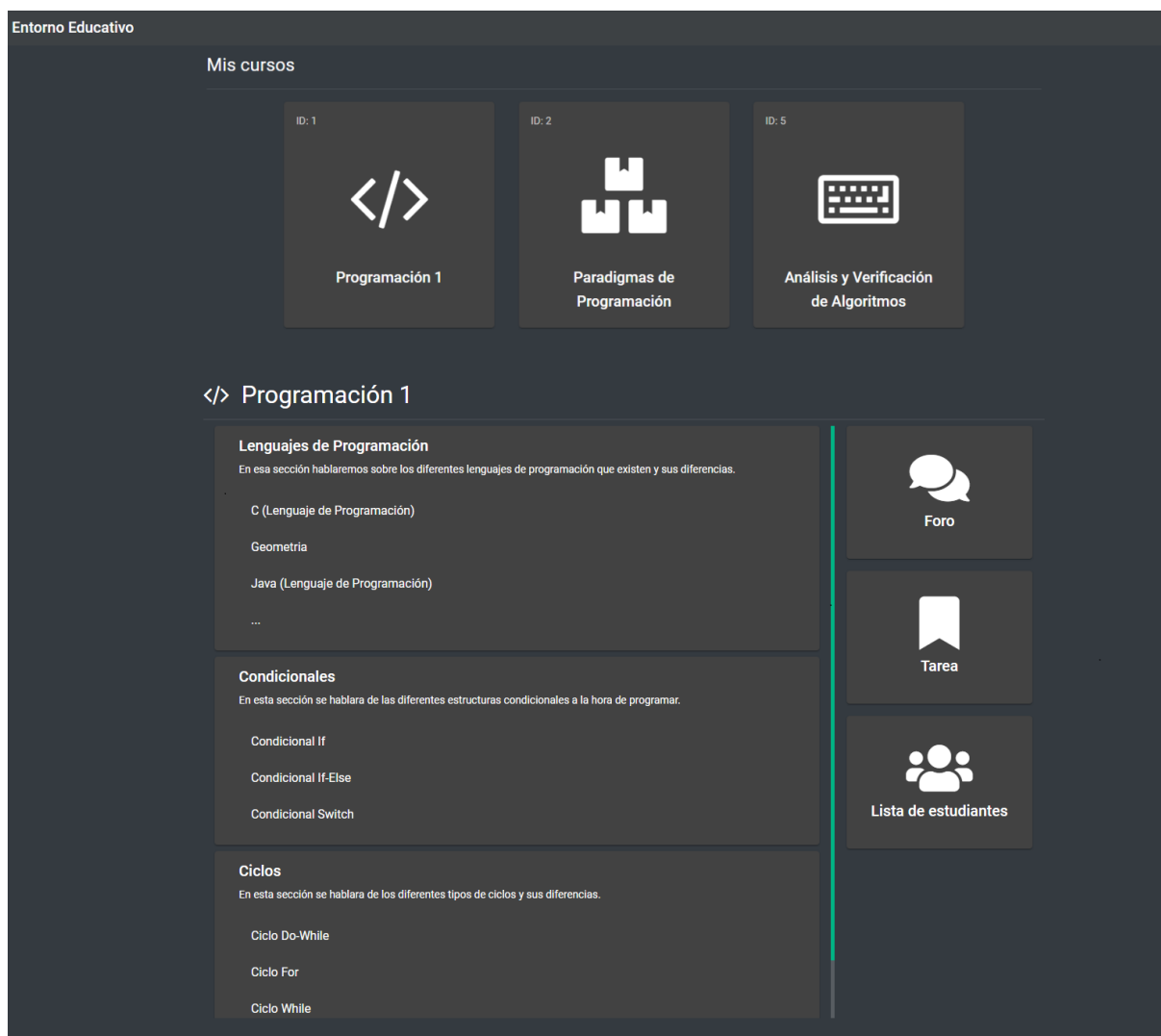


Figura 8.2: Pantalla de selección de cursos y página principal de el curso 'Programación 1'

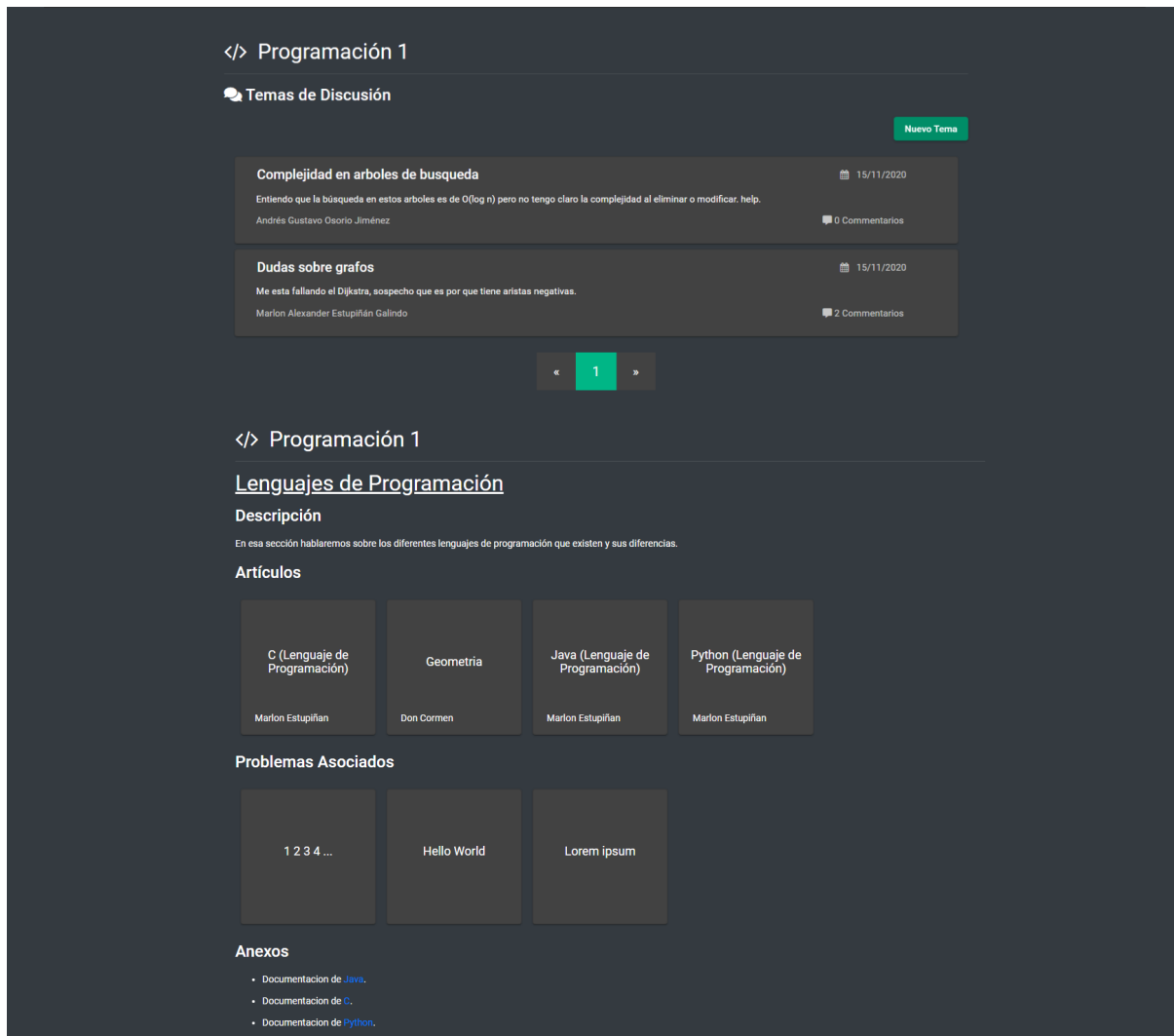


Figura 8.3: Pantalla de foro de discusión y vista principal de una sección del curso 'Programación 1'

Se muestra entonces una plataforma alternativa donde el docente puede gestionar de manera cómoda el curso, creando y organizando una serie de secciones en donde puede agregar artículos y problemas del repositorio de la aplicación, del mismo modo tiene el gestor de tareas donde puede administrar las calificaciones de los estudiantes de la clase.

8.2. Competencias

Una vez el creador de contenido (administrador) organice las competencias, los usuarios podrán encontrar un sistema de competencias de programación competitiva tradicional en

el cual pueden tener información básica de esta tal como su nombre, tiempos de inicio y finalización y estado como lo fundamental; una vista previa de los problemas, estado de los envíos de los usuarios participantes y su respectiva tabla de clasificación.

Competencia Fin de Semestre

Inicio: 2020-11-15 21:33 Final: 2020-11-15 23:33

Tiempo transcurrido: 15:31 Estado: Contenido Tiempo restante: 01:44:28

Vista General Problemas Estado Marcador

Soluciones/Intentos	Problema	Título
0 / 4	A	New Year Transportation
0 / 0	B	Dijkstra?
0 / 1	C	Favorite Time
0 / 0	D	Game On Leaves
1 / 1	E	Subsequence Hate

Vista General Problemas Estado Marcador

#	Usuario	Problema	Lenguaje	Resultado	Tiempo	Memoria	Fecha
16	GDAndres98	A	C++ 11	Error de Ejecución	449	193	2020/11/15 - 21:50:30
15	GDAndres98	C	C++ 11	Error de Ejecución	202	44	2020/11/15 - 21:50:15
14	GDAndres98	C	C++ 11	Límite de Tiempo Excedido	1000	43	2020/11/15 - 21:50:11
13	GDAndres98	C	C++ 11	Aceptado	149	62	2020/11/15 - 21:50:00
12	GDAndres98	A	C++ 11	Límite de Tiempo Excedido	2000	183	2020/11/15 - 21:49:52
11	GDAndres98	A	Phyton 3	Límite de Tiempo Excedido	2000	102	2020/11/15 - 21:49:44
10	GDAndres98	C	Phyton 3	Error de Ejecución	347	34	2020/11/15 - 21:49:30
9	GDAndres98	A	Phyton 3	Error de Presentación	998	31	2020/11/15 - 21:49:12
8	Alex_gal	E	Phyton 3	Aceptado	111	5	2020/11/15 - 21:47:48
7	Alex_gal	C	Phyton 3	Límite de Memoria Excedida	882	64	2020/11/15 - 21:47:40

« 1 2 »

Vista General Problemas Estado Marcador

Puesto	Usuario	Puntaje	Penalización	A	B	C	D	E
1	Alex_gal	1	15	X 4		X 1		14:48
2	GDAndres98	1	37	X 4		37:00		X 1

Figura 8.4: Vista de una competencia donde se aprecia la vista previa de los problemas, envíos de los usuarios y tabla de posiciones.

Se implementó entonces, para la comunidad de programación competitiva, una plataforma de competencias que puede ser organizada para los usuarios de esta, a partir de problemas subidos al sistema.

8.3. Entrenamiento Guiado

Aquí los usuarios pueden encontrar una guía gamificada y lineal de distintos temas de los que un usuario pueda mostrar interés. Cada nivel está conformado por artículos relacionados a problemas que el usuario debe solucionar para avanzar al siguiente nivel.

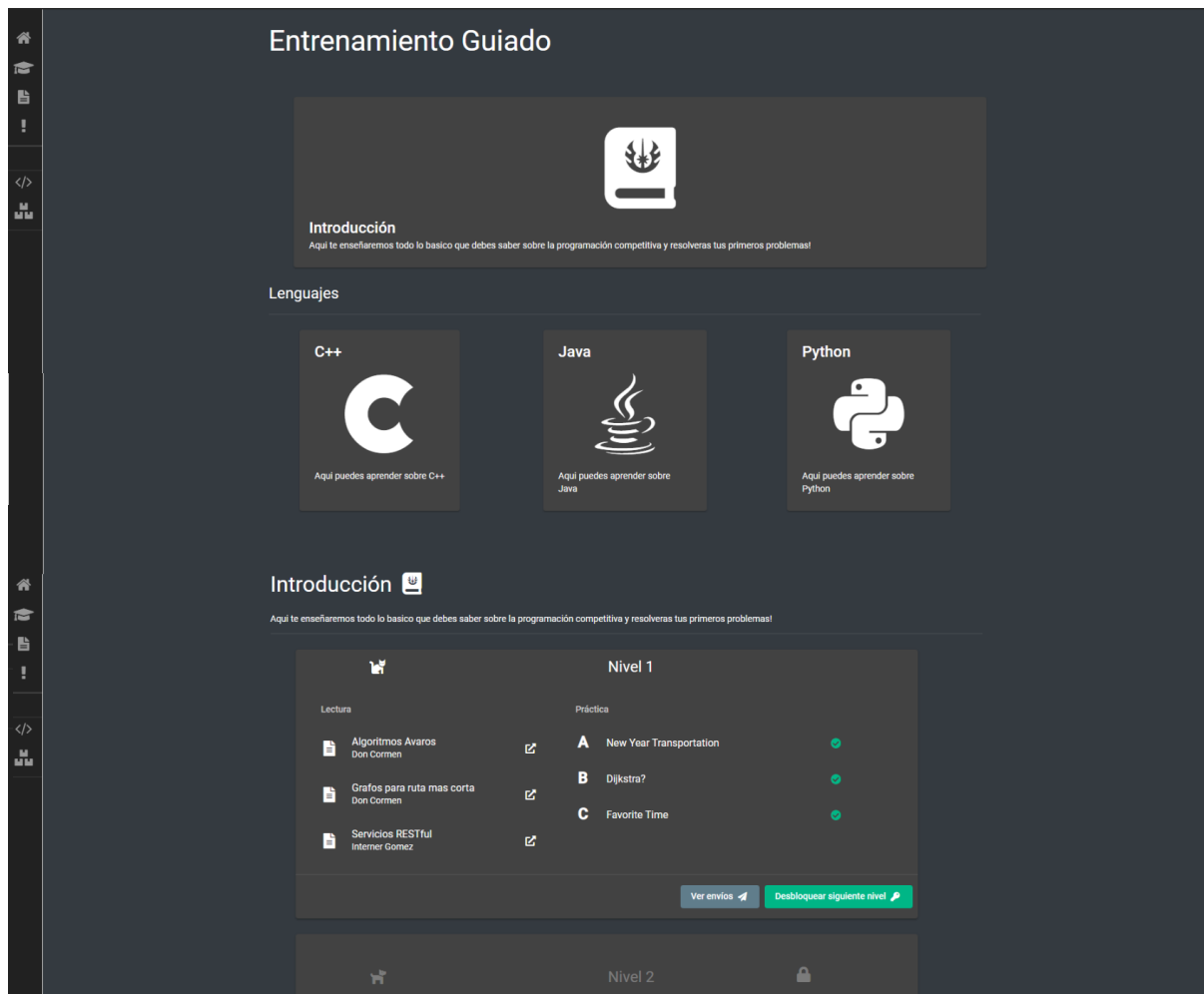


Figura 8.5: Vista principal del la pantalla de Entrenamiento Guiado, y vista del tema de Introducción.

A través de esto se busca fomentar de forma atractiva, para los usuarios más nuevos, el aprendizaje de algoritmia y la resolución de problemas mediante la programación.

8.4. Artículos y Problemas

Para la plataforma se creó todo un sistema que pudiera albergar tanto un repositorio de artículos como uno de problemas, esto con el motivo de que tanto cursos creados por docentes, el sistema de competencias y entrenamiento guiado puedan consumir estos.

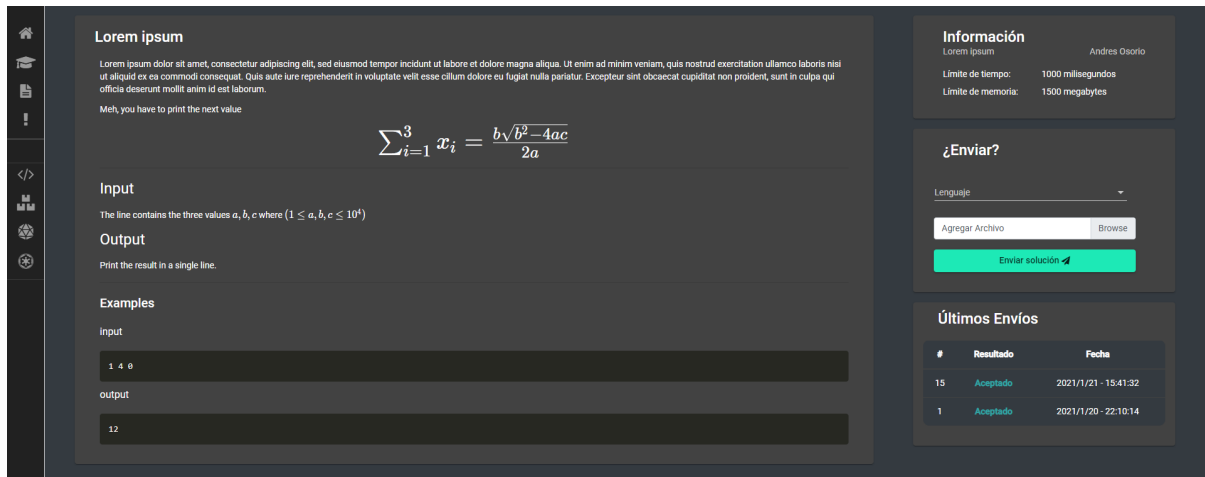
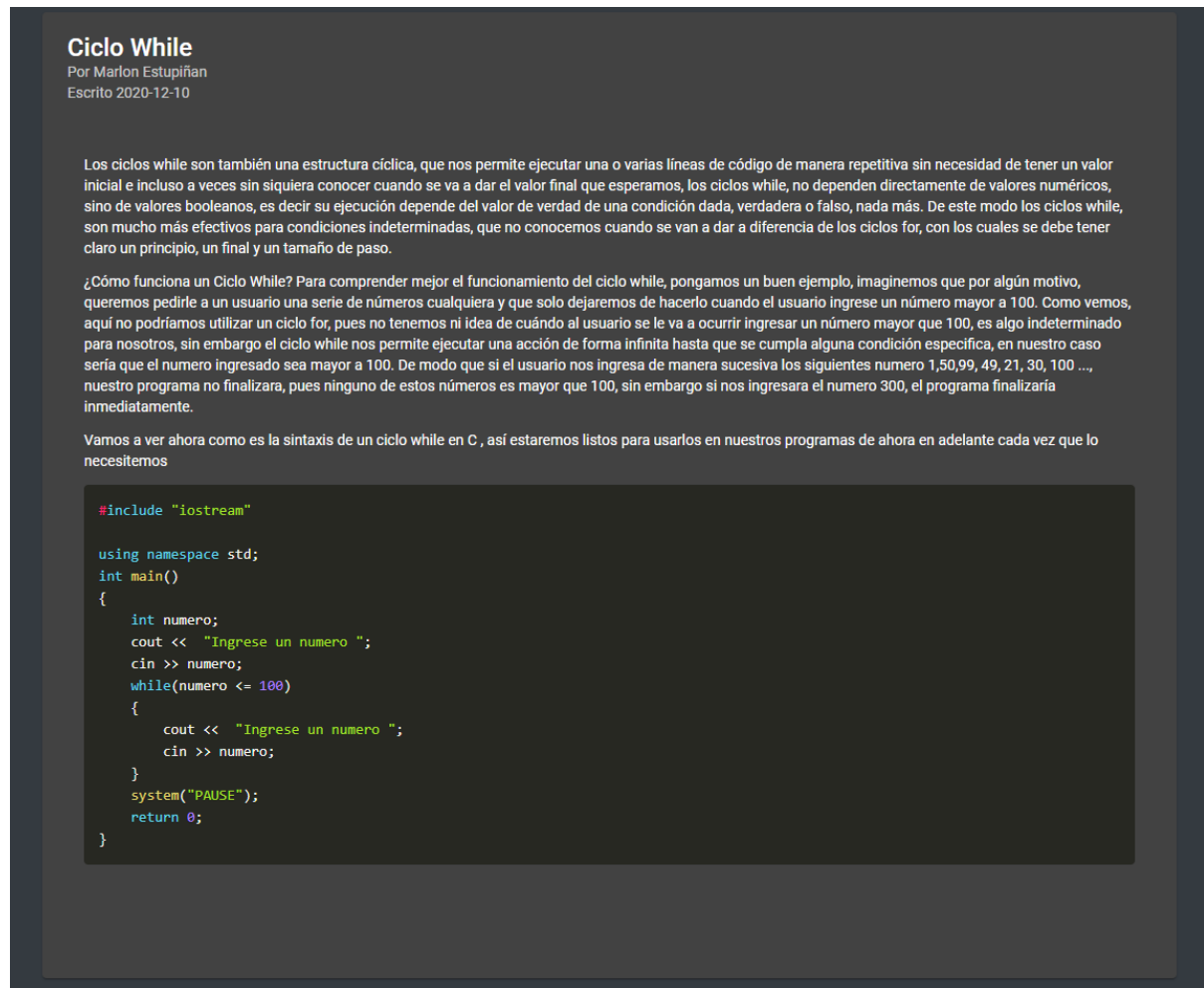


Figura 8.6: Vista de un problema de programación competitiva.

A través de esto se busca fomentar de forma atractiva, para los usuarios más nuevos, el aprendizaje de algoritmia y la resolución de problemas mediante la programación.



Ciclo While
Por Marlon Estupiñán
Escrito 2020-12-10

Los ciclos while son también una estructura cíclica, que nos permite ejecutar una o varias líneas de código de manera repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final que esperamos, los ciclos while, no dependen directamente de valores numéricos, sino de valores booleanos, es decir su ejecución depende del valor de verdad de una condición dada, verdadera o falso, nada más. De este modo los ciclos while, son mucho más efectivos para condiciones indeterminadas, que no conocemos cuando se van a dar a diferencia de los ciclos for, con los cuales se debe tener claro un principio, un final y un tamaño de paso.

¿Cómo funciona un Ciclo While? Para comprender mejor el funcionamiento del ciclo while, pongamos un buen ejemplo, imaginemos que por algún motivo, queremos pedirle a un usuario una serie de números cualquiera y que solo dejaremos de hacerlo cuando el usuario ingrese un número mayor a 100. Como vemos, aquí no podríamos utilizar un ciclo for, pues no tenemos ni idea de cuándo al usuario se le va a ocurrir ingresar un número mayor que 100, es algo indeterminado para nosotros, sin embargo el ciclo while nos permite ejecutar una acción de forma infinita hasta que se cumpla alguna condición específica, en nuestro caso sería que el número ingresado sea mayor a 100. De modo que si el usuario nos ingresa de manera sucesiva los siguientes número 1,50,99, 49, 21, 30, 100 ..., nuestro programa no finalizará, pues ninguno de estos números es mayor que 100, sin embargo si nos ingresara el número 300, el programa finalizaría inmediatamente.

Vamos a ver ahora como es la sintaxis de un ciclo while en C, así estaremos listos para usarlos en nuestros programas de ahora en adelante cada vez que lo necesitemos

```
#include "iostream"

using namespace std;
int main()
{
    int numero;
    cout << "Ingrese un numero ";
    cin >> numero;
    while(numero <= 100)
    {
        cout << "Ingrese un numero ";
        cin >> numero;
    }
    system("PAUSE");
    return 0;
}
```

Figura 8.7: Vista de un artículo.

Estos problemas y artículos son procesados por Markdown y Latex para una experiencia más agradable de lectura y un fácil entendimiento para el usuario.

8.4.1. Creador de contenido

A usuarios definidos del sistema se le puede asignar el rol de Administrador, el cual aparte de tener la capacidad de asignar roles a docentes y organizar competencias, es el creador de contenido para la creación, edición y eliminación de Artículos, Problemas y Cursos.

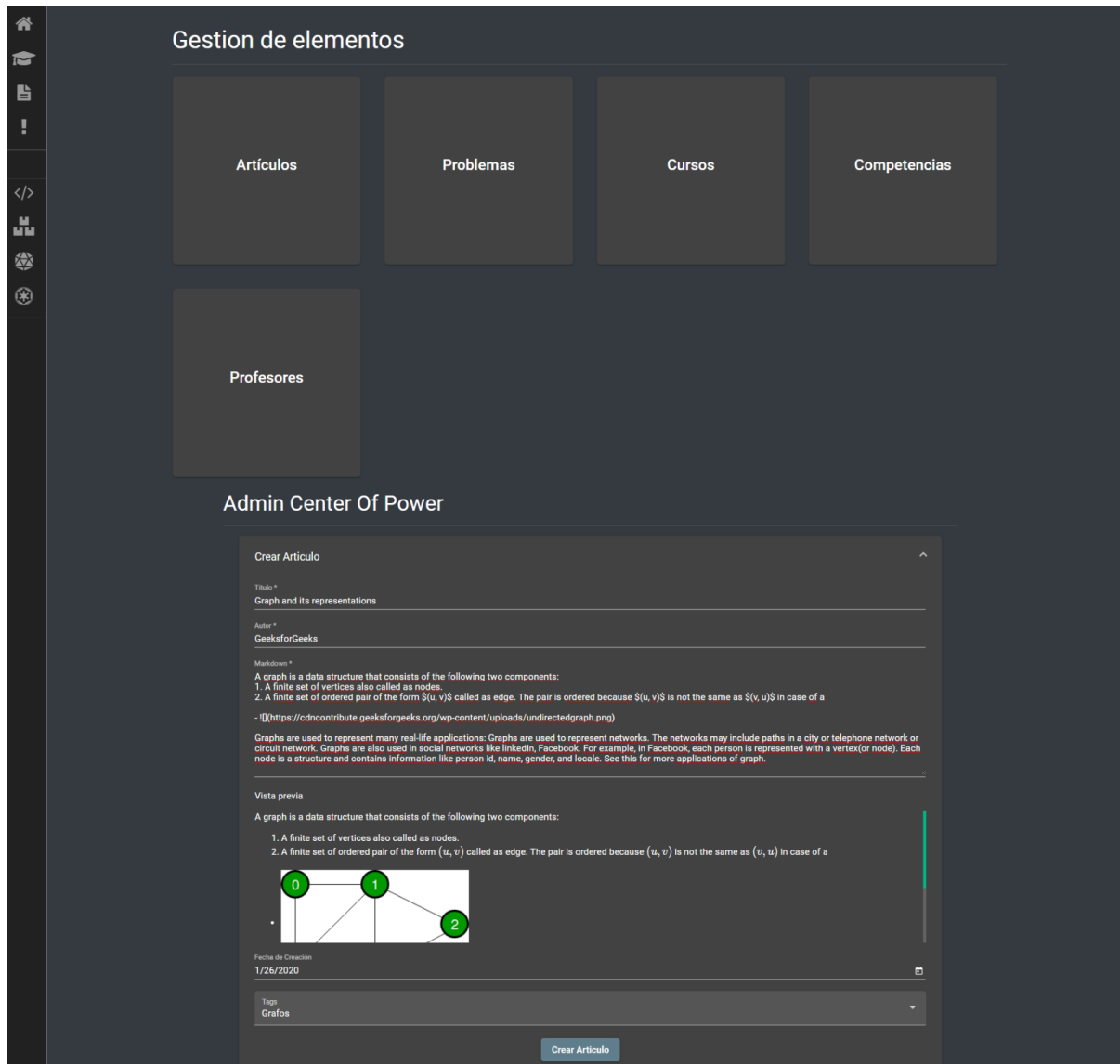


Figura 8.8: Vista principal del Administrador y vista de edición de un artículo

Capítulo 9

Conclusiones

Se realizó una ardua investigación de los sistemas de gestión de aprendizaje más reconocidos junto con los que usa el Politécnico Grancolombiano, como de las herramientas usadas por las comunidades inspiradas por la programación competitiva, con el propósito de resaltar características de estas y adaptarlas al sistema.

Durante todo el proceso de búsqueda y planteamiento del proyecto, los conocimientos adquiridos del programa de Ingeniería de Sistemas, fueron fundamentales para la correcta ejecución del desarrollo de la solución propuesta. La implementación de la metodología de desarrollo Scrum fue de vital importancia para lograr una medición y ejecución de tiempos y facilitar la priorización de tareas de las necesidades del aplicativo, buscando siempre maximizar el valor del producto.

Gracias a que es un modelo de desarrollo por capas y haciendo uso del patrón de modelo vista controlador, se facilitó el trabajo en equipo ya que al estar bien definida la separación entre las capas de presentación, lógica del negocio y acceso de datos permite un desarrollo simultáneo de todo el sistema y reduce significativamente el riesgo de errores.

El uso de tecnologías modernas, como una versión actual de Angular y Spring Boot, asegura una flexibilidad del proyecto para el soporte de nuevas tecnologías, evitando que el sistema y sus componentes requieran grandes cambios debido a su obsolescencia.

Este proyecto tiene como fin servir como un prototipo que busca incentivar el desarrollo

de una plataforma más grande que se pueda adaptar a distintos entornos diferentes a los de programación, e incluso ingeniería, y que se pueda escalar para soportar el volumen de usuarios de la universidad incluso ser capaz de expandirse a otras instituciones ajenas al Politécnico Grancolombiano.

Referencias

- ACIS. (2020). *Maratón de programación*. Descargado de <https://acis.org.co/portal/content/acis-maratón-de-programación>
- Adityanathloyola. (2020). *What is competitive programming and how to prepare for it?* Descargado de <https://www.geeksforgeeks.org/how-to-prepare-for-competitive-programming/>
- Bloomfield, A., y Sotomayor, B. (2014). *A programming contest strategy guide*. Descargado de <https://people.cs.uchicago.edu/~borja/pubs/sigcse2016-programming-contests.pdf>
- Burbeck, S. (1992). *How to use model-view-controller (mvc)*. Descargado de <https://web.archive.org/web/20120429161935/http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
- Campos, C. P., y Ferreira, C. E. (2004). *Boca: um sistema de apoio a competicoes de programacao*. Descargado de <https://www.ime.usp.br/~cassio/boca/campos-ferreira-wei2004.pdf>
- Cormen, T. H. (2003). *Introduction to algorithms*. Massachusetts: The MIT Press.
- Curry, E. (2004). Message-oriented middleware. *Qusay H Mahmoud*, 1-28.
- El Tiempo. (2020). *¿cómo está la educación virtual en colombia frente a la región?* Descargado de <https://www.eltiempo.com/vida/educacion/como-esta-la-educacion-virtual-en-colombia-frente-a-la-region-497476>
- Facebook. (2020). *Facebook hacker cup*. Descargado de <https://www.facebook.com/hackercup/>
- Gettys, R. T., Mogul, J., C., J., Frystyk, N. H., Masinter, L., Leach, P. J., y Berners-Lee, T. (1999). *Hypertext transfer protocol – http/1.1*. Descargado de <https://tools.ietf.org/html/rfc2616>
- Google. (2020). *Codejam*. Descargado de <https://codingcompetitions.withgoogle.com/codejam>
- Han, X. (2010). *Virtual judge?* Descargado de <https://vjudge.net/>

- Hibernate, J. (2011). *What is object/relational mapping?* Descargado de <http://hibernate.org/orm/what-is-an-orm/>
- Huambachano, J. F. (2017). *¿qué es scrum?* Descargado de <https://www.scrum.org/resources/blog/que-es-scrum/>
- Iqbal, S., y Qureshi, I. (2011, 10). Learning management systems (lms): Inside matters. *Inf. Manag. Bus. Rev.*, 3, 206-216. doi: 10.22610/imbr.v3i4.935
- Jain, C. (2016). *Geeksforgeeks - a computer science portal for geeks.* Descargado de <https://yourstory.com/mystory/245fe4c759-geeksforgeeks-a-computer-science-portal-for-geeks>
- Knuth, D. (2011). *Algorithms.* Descargado de https://courses.cs.vt.edu/cs2104/Fall17Barnette/notes/T16_Algorithms.pdf
- Kosner, A. W. (2014). *Hackerrank solves tech hiring crisis by finding programmers where they live.* Descargado de <https://www.forbes.com/sites/anthonykosner/2014/06/12/hackerrank-solves-tech-hiring-crisis-by-finding-programmers-where-they-live/?sh=3ddd16c2b644>
- LeetCode. (2020). *Leetcode.* Descargado de <https://leetcode.com/explore/>
- Lin, W. (2020). *What is competitive programming?* Descargado de https://www.youtube.com/watch?v=ueNT-w70luw&ab_channel=WilliamLin
- Linwood, J., y Minter, D. (2010). *Beginning hibernate.* Apress.
- Mirzayanov, M. (2010). *What is codeforces?* Descargado de <https://codeforces.com/blog/entry/1>
- Politécnico Grancolombiano. (2019). *El poli, es nuevamente la sede por quinta vez de la maratón nacional de programación.* Descargado de <https://www.poli.edu.co/noticias/el-poli-es-nuevamente-la-sede-por-quinta-vez-de-la-maraton-nacional-de-programacion>
- Politécnico Grancolombiano. (2020a). *Ingeniería de sistemas - bogotá.* Descargado de <https://www.poli.edu.co/profesional/ingenieria-de-sistemas-bogota>
- Politécnico Grancolombiano. (2020b). *Ingeniería de software - virtual.* Descargado de <https://www.poli.edu.co/profesional/ingenieria-de-software-virtual>
- Politécnico Grancolombiano. (2020c). *Maratón regional de programación.* Descargado de <https://www.poli.edu.co/maratonprogramacion>
- Politécnico Grancolombiano. (2020d). *Quiénes somos.* Descargado de <https://www.poli.edu.co/content/quienes-somos>
- Precht, P. (2016). *Angular 2 is out - get started here.* Descargado de <https://blog.thoughtram.io/angular/2016/09/15/angular-2-final-is-out.html>

- Rahman, K. A. (s.f.). *Fundamental concept of information technology*. Descargado de https://www.streetdirectory.com/travel_guide/126188/technology/fundamental_concept_of_information_technology.html
- Revilla, M. A., Manzoor, S., y Liu, R. (2008). *Competitive learning in informatics: The uva online judge experience*. Descargado de <https://ioinformatics.org/journal/INFOL035.pdf>
- R.S/E.C. (2011). *Algunas ventajas y desventajas de la formación 'online'*. Descargado de <https://www.20minutos.es/noticia/1261104/0/ventajas/desventajas/formacion-distancia/>
- Uturunco, R. C. C. (2019). *¿qué es programación competitiva? y ¿por qué debería comenzar a aprenderla?* Descargado de <https://medium.com/elemental-school/qué-es-programación-competitiva-y-por-qué-debería-comenzar-a-aprenderla-66f44dff53f3>
- World Wide Web Consortium. (2016). *Relationship to the world wide web and rest architectures*. Descargado de <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest>