

Proyecto Final de Carrera

“Desarrollo de una herramienta para
evaluar conocimientos de programación en
alumnos de primer nivel de la carrera
Ingeniería en Sistemas de Información”

Integrante: Francisco Bertone

Director: Ingeniero Daniel Ambort

Universidad: Universidad Tecnológica
Nacional

Facultad: Facultad Regional de Santa Fe

Año: 2021

Índice

1. Introducción -----	2
1.1. Glosario -----	2
1.2. Temática -----	3
1.3. Problema a Resolver -----	3
1.4. Objetivos Generales -----	4
1.5. Objetivos Específicos -----	4
1.6. Alcance -----	5
1.7. Marco Teórico -----	5
2. Desarrollo -----	7
2.1. Requerimientos Funcionales del Sistema -----	7
2.2. Metodología -----	8
2.3. Análisis -----	9
2.4. Velocidad Estimada y Real -----	10
2.5. Diseño -----	11
2.6. Pruebas Funcionales -----	19
2.7. Gestión de Riesgos y Contingencias Enfrentadas -----	23
2.8. Puesta en Marcha -----	26
3. Interfáz Grafica -----	26
4. Herramientas Utilizadas -----	33
5. Arquitectura -----	37
6. Trabajos Futuros -----	37
7. Conclusión -----	38
8. Bibliografía -----	39
9. Anexo -----	40

1. Introducción

En esta sección, se da una introducción formal del proyecto, se explica cómo surgió la necesidad del mismo y se definen cuestiones como el alcance y los objetivos, tanto generales como específicos, que se pretenden alcanzar.

1.1 Glosario

- **Algoritmo:** Un algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. En el caso de este sistema, es una forma de categorizar a los problemas. Los problemas que se encuentren dentro de esta categoría pueden ser resueltos con este algoritmo (o con variaciones del mismo). Por ejemplo, dentro de la categoría “Algoritmo Burbuja” podemos encontrar problemas que se puedan resolver con el método Burbuja o con variantes del mismo.
- **Alumno:** En el sistema, se le llama así a cualquier usuario que no sea administrador. Pueden ser alumnos de alguno de los cursos de programación o personas que quieran inscribirse a un concurso de programación
- **Back-End:** Es el desarrollo de la lógica del sistema, que procesa las entradas que vienen desde la interfaz de usuario.
- **Caso de uso:** Descripción de una acción o una actividad en el modelo de desarrollo clásico.
- **Concurso:** Hace referencia a cualquier competencia de programación, por ejemplo: Tecnomate
- **Curso:** Hace referencia a cualquier curso en el cual se dictan conocimientos relacionados a la programación, por ejemplo: Algoritmos y estructuras de datos, Paradigmas de Programación, Diseño e Implementación de estructuras de datos, etc.
- **Desarrollo clásico Incremental:** Un modelo de desarrollo basado en el modelo clásico, en el cual lo que se hace básicamente es realizar las etapas de Análisis-Diseño-Codificación-Prueba un N número de veces hasta que se obtiene el resultado deseado.
- **Diagrama de Clases:** Es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.
- **Diagrama de Entidad-Relación (DER):** Representación de las entidades de una base de datos y sus atributos.
- **Entorno de desarrollo:** Un IDE (Entorno de Desarrollo Integrado – Integrated Development Environment) es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software.
- **Front-End:** Es el desarrollo de la interfaz con la que interactúan los usuarios.

- **Problema:** Problema que debe ser resuelto mediante programación, en este sistema, consiste en un elemento dentro de la respectiva categoría de algoritmo que tiene un link a la web que hostea el problema en cuestión.
- **Profesor:** En el sistema, se le llama así a cualquier usuario administrador. Son profesores de cualquier cátedra de programación de la facultad.
- **Requerimiento Funcional:** Un requisito funcional en un documento de requerimientos describe una funcionalidad que debe cumplir el sistema.
- **Requerimiento No Funcional:** Un requisito no funcional o atributo de calidad es un requisito que sabe bien y especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento.
- **Servidor web:** Un servidor web es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.
- **Tema:** Un tema del Curso/Concurso correspondiente. Por ejemplo, en el curso de primer año de Algoritmos y Estructuras de Datos, algunos de los temas podrían ser Arreglos, Caracteres o Bucles.

1.2 Temática

Este proyecto final de carrera fue llevado a cabo con la finalidad de la creación de una herramienta que sirva para evaluar conocimientos de programación en alumnos de la carrera Ingeniería en Sistemas de Información.

1.3 Problema a resolver

Cuando un alumno comienza a estudiar Ingeniería en Sistemas de Información en la facultad, en algunos casos ya lo hace con conocimientos básicos de programación, ya sea por haber tenido alguna materia específica en el nivel medio, o por haber aprendido por interés personal. Es frecuente que, en esos casos, los alumnos conozcan los primeros temas de Algoritmos y Estructuras de Datos (AEDD) en mayor o menor medida. Este también es el caso de alumnos que ya terminaron su primer año, y para las siguientes materias troncales de programación, ya poseen conocimientos parciales de los desarrollados en las mismas. También es posible que alumnos que no alcanzaron a regularizar alguna de las materias ya posean ciertos conocimientos antes de volver a cursarlas.

Las cátedras de AEDD, Paradigmas de Programación y demás materias del área tienen la necesidad de una herramienta con la cual se pueda medir el nivel de

conocimiento de cada alumno al inicio del cursado, para poder así planificar el desarrollo de la materia de manera acorde, con el objetivo de organizar actividades que sean significativas para alumnos que comienzan con nivel de conocimientos dispar.

Disponer de un perfil mejor definido de los alumnos puede ser de gran ayuda si tenemos en cuenta que, en primer año, las comisiones suelen ser numerosas (3 comisiones de 100 alumnos en 2020).

1.4 Objetivos Generales

El objetivo general es el desarrollo de un sistema web que permita resolver el problema presentado anteriormente, permitiendo a los profesores de las distintas cátedras evaluar los conocimientos de los alumnos en los temas de la currícula de las asignaturas del área de Programación de la carrera de Ingeniería en Sistemas de Información.

1.5 Objetivos Específicos

Hoy en día existen aplicaciones, de escritorio y online, que evalúan conocimientos de programación e implementan correcciones automáticas de forma práctica mediante un test de caja negra sobre códigos fuente que los alumnos envían como solución a Problemas determinados, tales como las páginas de jueces online (por ej. URI Online Judge), pero con estas no se puede evaluar ni definir los temas con la precisión que se quisiera, con lo cual es necesario el desarrollo de una herramienta dedicada en la cual los profesores puedan participar activamente y definir Temas, Algoritmos asociados a estos temas y los Problemas asociados a los Algoritmos, para así tener una evaluación más profunda y consistente del nivel de los alumnos en temas de programación definidos.

Otro objetivo de esta herramienta es permitir que otros alumnos observen el progreso de sus compañeros en algunas actividades seleccionadas, para socializar el desempeño individual y grupal de las comisiones. Esta socialización favorece la competencia y la cooperación de los alumnos con respecto al aprendizaje.

Este objetivo plantea la necesidad de la asociación de las cuentas de usuario de los alumnos al desempeño de los mismos en sitios o competencias de programación (por ej. TecnoMate).

En cuanto a requerimientos específicos de desarrollo, se pretende:

- Una interfaz amigable tanto para profesores como para alumnos, que pueda utilizarse en diferentes dispositivos.
- Una herramienta con una lógica de funcionamiento bien definida e implementada.

- Una base de datos que soporte la información y que responda adecuadamente en su comunicación con la aplicación.

1.6 Alcance

Se espera que todos los profesores del área Programación y todos los alumnos de I.S.I de la UTN Santa Fe puedan utilizar el sitio web, así como aquellos que no pertenezcan a la facultad, pero quieran participar de una competición/concurso determinado de programación.

Los alumnos/usuarios podrán:

- Inscribirse en cursos/concursos
- Resolver Problemas dentro de los mismos.
- Formar equipos con otros usuarios en concursos a los que se hayan inscripto.

Los profesores podrán:

- Crear Cursos, y, dentro de los mismos, Temas, Algoritmos y Problemas.
- Aprobar o rechazar las soluciones enviadas por los alumnos.

Con respecto al alcance del proyecto en sí, las actividades a realizar serán:

- Todas las que conllevan un desarrollo clásico Iterativo (Recolección de requerimientos, Análisis, Diseño e Implementación).
- Testing de todas las funcionalidades del sistema.
- Reuniones y comunicación con el director de proyecto entre las iteraciones del desarrollo.
- Investigaciones varias para la puesta en marcha del proyecto, tales como la elección del lenguaje y el uso de este, motor de base de datos, programas de diagramado, metodología de desarrollo, etc.

1.7 Marco Teórico

Un Juez Online es un sistema que es desarrollado principalmente para poder testear las soluciones en concursos de programación, si bien hoy en día se utilizan en contextos externos a las competiciones como, por ejemplo, disponer de ejercicios de práctica para los alumnos de un curso de programación.

Lo que hace este sistema es compilar y ejecutar el código enviado y comparar su salida con datos preparados y luego mostrar el resultado. En algunos casos puede haber límites de tiempo, de uso de memoria e incluso de seguridad.

Si se encuentran errores, se considerará que el programa no es correcto y será rechazado.

Algunos ejemplos de jueces online son:

- URI Online Judge
- Sphere Online Judge
- Panda Online Judge

Debido al crecimiento del uso de estos jueces tanto en competencias como en los cursos de programación, nació la necesidad de un software como el que aquí se describe ya que todos estos jueces tienen distintos Problemas los cuales sólo pueden ser resueltos en su respectiva página/aplicación. De esta forma, un profesor puede darles a sus alumnos para que resuelvan, por ejemplo, dos Problemas hospedados en URI, uno en Sphere y otro en Panda, y poder efectivamente ver que los resolvieron a todos.

De la misma forma, si es necesario practicar para las competencias, los usuarios pueden ver los Problemas que sus compañeros resolvieron en todos los jueces online, sin necesidad de tener que estar revisando las cuentas de los mismos en cada una de ellos.

En el caso de los cursos, los problemas subidos serán relacionados a la cátedra, por ejemplo, dentro del curso AEDD 1er año, habrá un tema llamado “Arreglos”, dentro de los cuales habrá categorías de algoritmo como por ejemplo “Algoritmo burbuja”, “Algoritmo selección”, y dentro de estas subcategorías problemas que pueden ser resueltos por esos algoritmos.

Ahora bien, si bien el ejemplo de arriba es un caso común, discutiendo las bases con el Ingeniero Ambort se detectaron categorías que no siempre serán utilizadas de esta forma. Supongamos que, también dentro del curso AEDD de 1er año, existe el tema “Caracteres” o “Enteros”. Quizás aquí no haya que usar una categoría de algoritmos, sino simplemente existirá una categoría de algoritmos también llamada “Caracteres” o “Enteros”.

En el caso de los concursos, los problemas subidos serán aquellos que el profesor considere adecuados para que el alumno practique para esa competencia en particular. De esta forma, supongamos que tres alumnos forman un equipo para la edición 2021 del Tecnomate, al crearse cada uno una cuenta en el sistema y agregarse como amigos en ese concurso, podrán coordinar bien que problemas resolver, ver las fortalezas de cada uno, etc.

2. Desarrollo

En esta sección se realiza una descripción del proceso de desarrollo del software, empezando por la definición de las funcionalidades básicas del sistema y terminando con las pruebas realizadas para comprobar el correcto funcionamiento del mismo.

2.1 Requerimientos Funcionales del Sistema

-Registrar Profesor: Registrar un nuevo usuario del tipo profesor en la página, este tipo de usuario es el que podrá modificar el contenido de la página web dedicada a la evaluación del conocimiento de programación de los alumnos, agregar temas, Algoritmos, Problemas, etc.

-Registrar Alumno: Registrar un nuevo usuario del tipo alumno en la página web, este tipo de usuario es el principal objetivo en cuanto a seguimiento y evaluación. Podrá resolver Problemas de los temas disponibles y asociarse con compañeros para participar en un determinado Concurso.

-Autenticar Usuario: Que un Usuario, ya sea profesor o alumno, pueda iniciar sesión para poder utilizar el sistema.

-Añadir Curso: Añadir nuevo Curso en el que se registran los alumnos. Una vez que este sea subido, podrán registrarse los alumnos para poder resolver los Problemas dentro del mismo.

-Añadir Tema: Añadir nuevo Tema para el aprendizaje y evaluación de los alumnos dentro de un Curso o Concurso.

-Añadir Algoritmo: Añadir nuevo Algoritmo dentro de un tema.

-Añadir Problema: Añadir nuevo Problema dentro de un Algoritmo. Estos podrán ser resueltos por los alumnos en su proceso de aprendizaje.

-Eliminar Tema: Eliminar Tema para evaluar a los alumnos dentro de un curso o Concurso. Sólo se eliminará el Tema de tal forma que no más alumnos puedan acceder al mismo dentro del curso, los datos de los alumnos que resolvieron los Problemas del Tema seguirán existiendo.

-Eliminar Algoritmo: Eliminar Algoritmo dentro de un tema.

-Eliminar Problema: Eliminar Problema dentro de un Algoritmo.

-Enviar solución de Problema: Que un alumno envíe su solución a un Problema. El alumno envía una captura del juez online con el Problema resuelto, verificando que es efectivamente suya, debe verse su nombre de usuario del respectivo juez online, y, opcionalmente, el código mismo del Problema.

-Inscribirse a curso: Que un alumno se inscriba a un curso en el cual resolverá Problemas durante su aprendizaje.

-Inscribirse a concurso: Que un alumno se inscriba en un concurso de programación en el cual participará ya sea solo o junto con otros compañeros.

-Añadir a un compañero a un concurso en común: Que un alumno añada a un compañero de grupo para un concurso de programación. Se vincularán ambos usuarios con el mismo concurso y le aparecerán al usuario en su perfil los Problemas resueltos por sus compañeros.

-Aceptar Solución: Que un profesor pueda aceptar la solución de un Problema de un curso determinado enviada por uno de sus alumnos.

2.2 Metodología

La metodología elegida está basada en el método clásico de desarrollo incremental, ya que, al ser un proyecto individual, muchos aspectos de las metodologías ágiles fueron descartados. Por ejemplo, los roles no se pudieron asignar a distintas personas ya que existía un único integrante en el equipo, la programación en parejas es algo que tampoco se pudo hacer por el mismo motivo.

A pesar de esto, sí se utilizaron algunos aspectos de metodologías ágiles, como, por ejemplo, las reuniones con el cliente para mostrar los avances, la flexibilidad con respecto a la documentación, etc.

Las etapas fueron:

- **Recolección de requerimientos:** En esta etapa, se llevaron a cabo reuniones con el director del proyecto para discutir cuáles eran las funcionalidades que debía tener el sistema, hablando claramente con él y tomando notas de todas sus indicaciones. Luego estas conversaciones se desglosaron y convirtieron en requisitos para el sistema.
- **Análisis:** Luego de obtener los requisitos en la primera etapa, estos se dividieron en funcionales y no funcionales. Los funcionales fueron refinados en casos de uso.
- **Diseño:** Se obtuvo el diseño general del sistema en esa iteración, el Diagrama de Entidad de Relaciones para la base de datos y el Diagrama de Clases del sistema. No se realizaron diagramas de secuencia para tener un poco más de flexibilidad con respecto a la documentación.
- **Implementación:** Aquí se realizó la codificación del sistema propiamente dicha, se escribió el código PHP/HTML/CSS y se realizaron las tablas MySQL necesarias.

El plan consistió básicamente en definir ciertas funcionalidades para una iteración luego de hacer la recolección de requerimientos, análisis y diseño general. Se implementaron algunas funcionalidades y, luego de mostrar el progreso de desarrollo, se realizó una reunión con el cliente en la cual se refinaron adecuadamente los requisitos, lo que implicó la modificación o la adición de algunos aspectos al diseño inicial. Luego se desarrollaron las funcionalidades de esta segunda iteración y así sucesivamente hasta llegar al final.

Si bien este fue el plan inicial, sólo se requirió una etapa de refinamiento de requerimientos, y, aun así, sólo se agregaron pocos más, con lo cual no hubo tanta predominancia del desarrollo incremental.

Este tipo de metodología fue adecuada, ya que al ser este un trabajo unipersonal, se puede llegar a pensar que utilizar una metodología ágil no hubiese tenido sentido, y los métodos clásicos tienen una estructura que es un poco más fácil de seguir. Sin embargo, realizar una implementación puramente clásica tampoco hubiese sido prudente, ya que no se contaba con un equipo de desarrollo para poder reaccionar a los cambios de manera lo suficientemente rápida, siendo que ya de por sí los métodos clásicos no se adaptan muy bien a los cambios en los requerimientos. Para esto resultó útil tomar partes de ambas metodologías. Las reuniones con el director del proyecto permitieron adaptarse rápidamente si algo no estaba realizándose de la forma esperada.

2.3 Análisis

En esta etapa se obtuvieron las especificaciones de Casos de Uso. Ver anexo.

2.4 Velocidad estimada y real

En esta sección veremos cómo fue realmente el desarrollo con respecto a las proyecciones hechas en el plan de proyecto.

Cabe aclarar que nunca hubo una tercera revisión, el proyecto se consideró finalizado luego de la 2da.

Estimaciones realizadas en el plan de proyecto:

- Entrega primera iteración: abril de 2020.
- Entrega segunda iteración: mayo de 2020.
- Entrega tercera iteración: julio de 2020.

Fechas reales:

- Entrega primera iteración: 18 de agosto de 2020.
- Entrega segunda iteración: 2 de septiembre de 2020.

Como puede apreciarse, el proyecto arrancó bastante más tarde de lo estimado, a mitades del mes de julio, por lo que todas las fechas fueron desplazadas.

Por ello la primera iteración, en la cual se debió arrancar desde el principio, fue terminada el 18 de agosto. La segunda iteración fue solo dos semanas después, ya que los cambios sugeridos fueron mínimos, simplemente se agregó la opción para los profesores de calificar las entregas y la opción de agregar el código.

El plan se contempló que las actividades de investigación serían completadas en el mes de marzo, esto no fue del todo así ya que recién el mes de mayo pude completar el aprendizaje de los lenguajes necesarios (PHP / HTML5 / CSS / MYSQL) para el desarrollo del proyecto.

Adicionalmente, si bien el trabajo iba a ser de cuatro horas diarias, esto era más que nada una estimación, ya que, si bien aproximadamente se cumplió la cantidad de horas, muchos días se hacían horas de más, hasta llegar incluso a las 8 horas, y otros días de menos atento a que trataba de que no queden tareas por la mitad.

En cuanto a la duración de la implementación de los casos de uso en sí, en la siguiente tabla se puede ver el tiempo estimado y el tiempo real que tomó cada Caso de Uso para estar en funcionamiento.

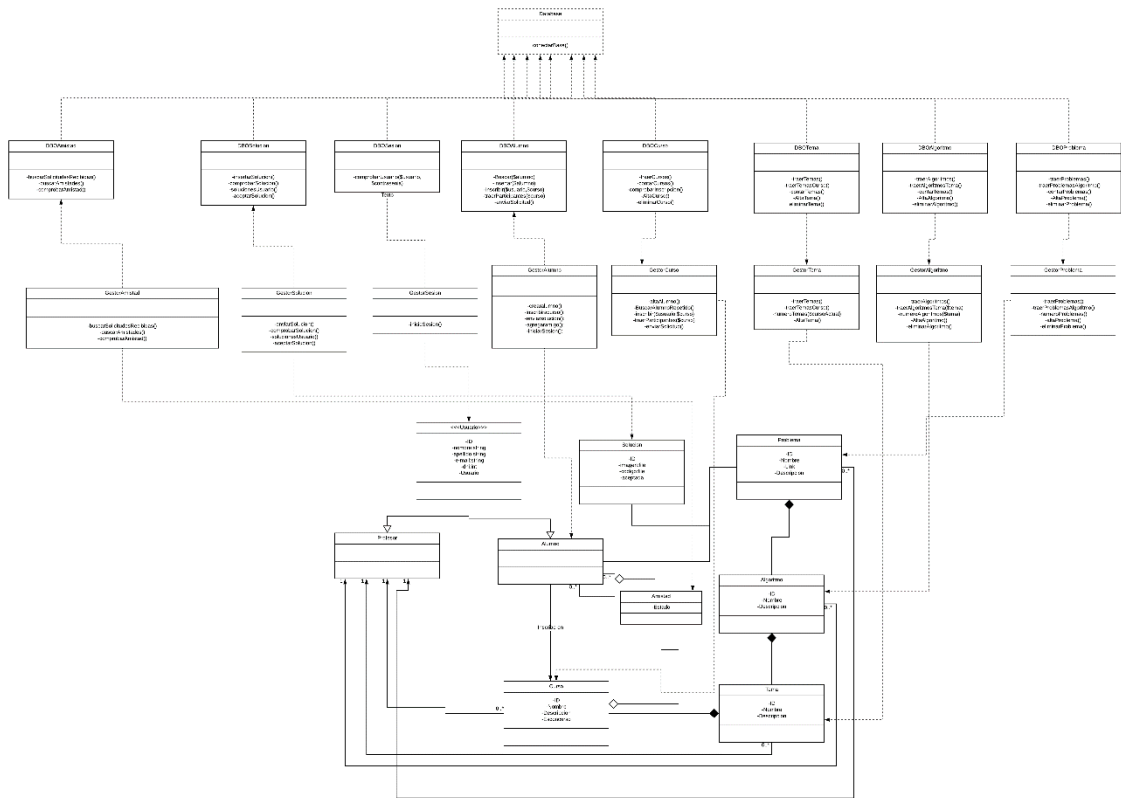
Caso de Uso	Esfuerzo Estimado	Esfuerzo Real
Registrar Profesor	10hs	0hs
Registrar Alumno	10hs	25-30hs
Autenticar Usuario	10hs	2hs
Añadir Curso	10hs	20hs
Añadir Tema	10hs	2hs
Añadir Algoritmo	10hs	2hs
Añadir Problema	20hs	5hs
Eliminar Curso	10hs	10hs
Eliminar Tema	5hs	2hs
Eliminar Algoritmo	5hs	2hs
Eliminar Problema	5hs	2hs
Enviar Solución Problema	30hs	25-30hs
Inscribirse a curso	15hs	20hs
Inscribirse a concurso	10hs	0s
Añadir Compañero a un curso en común	30hs	30hs

En la mayoría de los casos, salvo excepciones, se sobreestimó la complejidad de algunas tareas, sobre todo las altas y las bajas. Los registros de profesor se decidieron hacer por fuera del sistema por lo cual el Caso de Uso no se realizó, y como la inscripción a un concurso es la misma que a un curso, tampoco se utilizaron horas para dicho caso.

Cabe aclarar, también, que las horas de esfuerzo real de la tabla solo hacen referencia al tiempo de desarrollo, sin considerar otras actividades como, por ejemplo, el testing unitario.

2.5 Diseño

Los diagramas que se utilizaron para la implementación del sistema propiamente dicho.



1 Diagrama de clases (Parte 1)

Haciendo énfasis en distintas partes de este diagrama para mejor claridad:

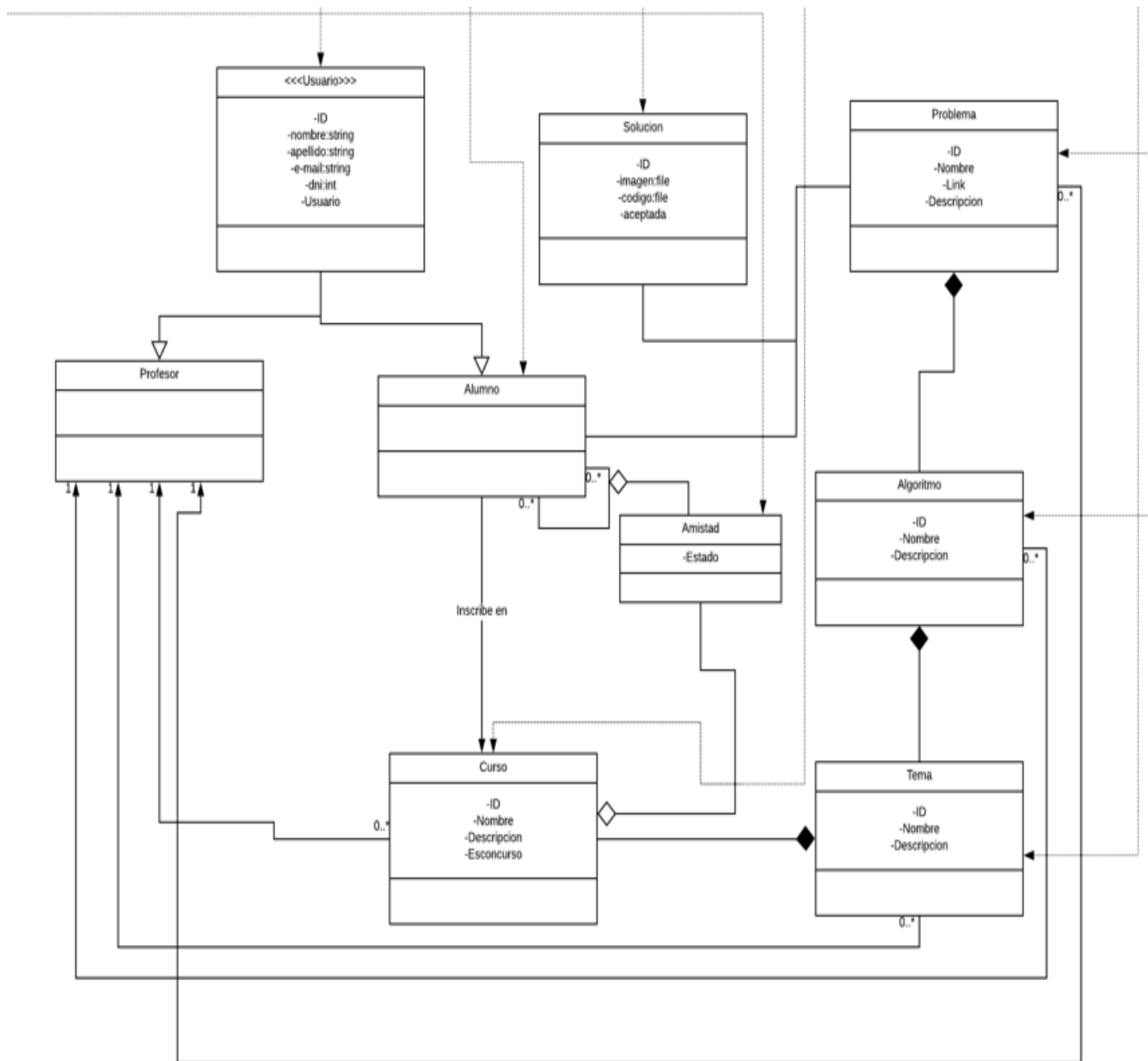
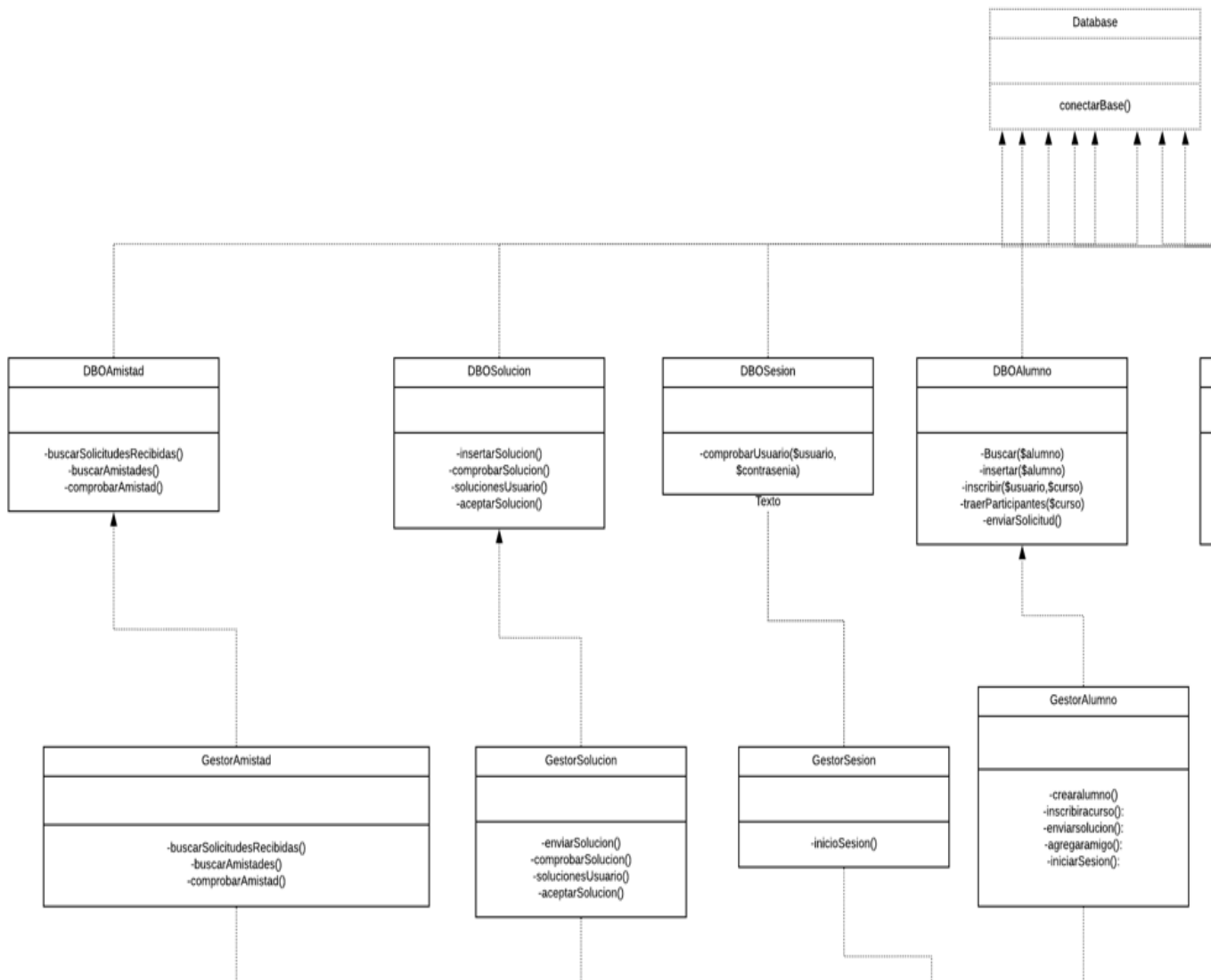


Diagrama de clases parte 1 (A)



*

Diagrama de clases Parte 1 (B)

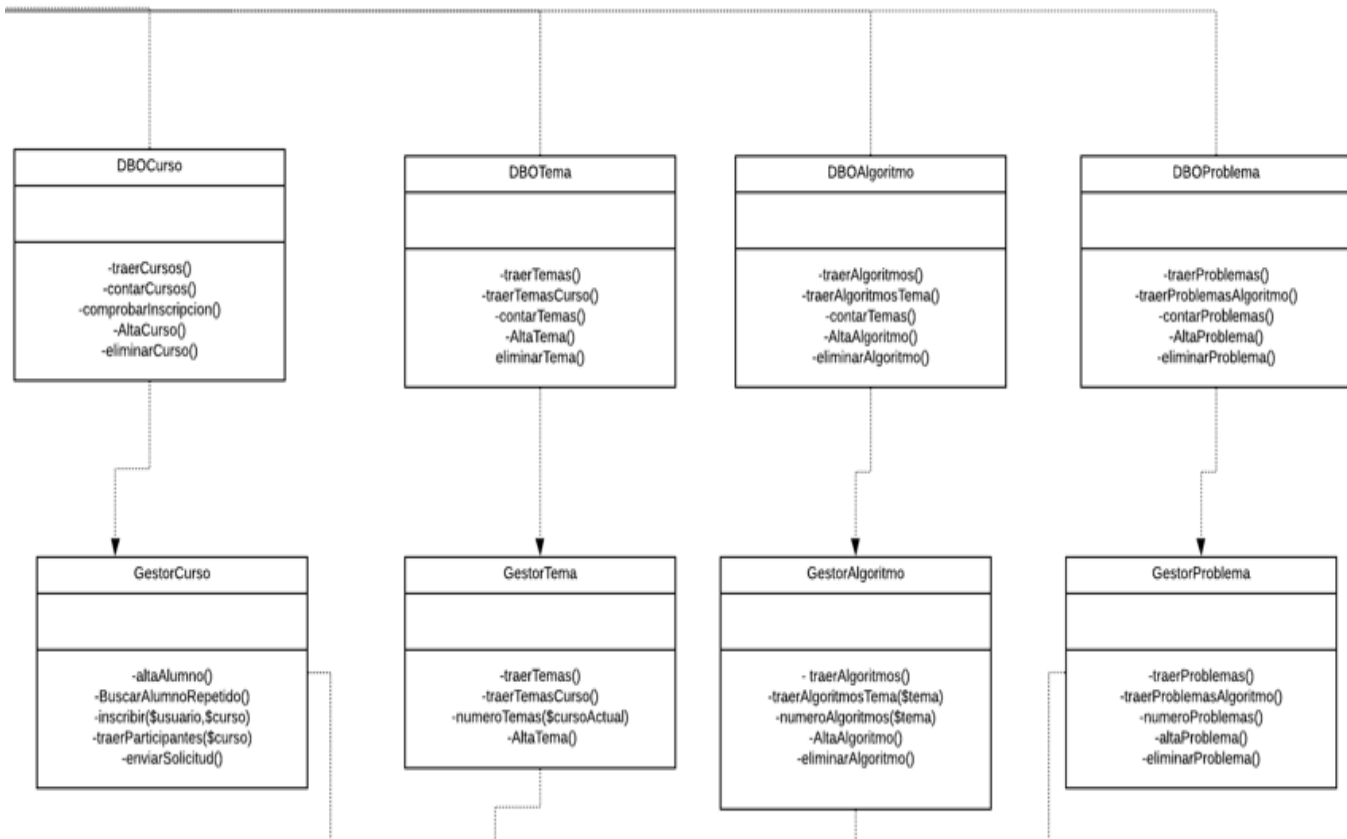
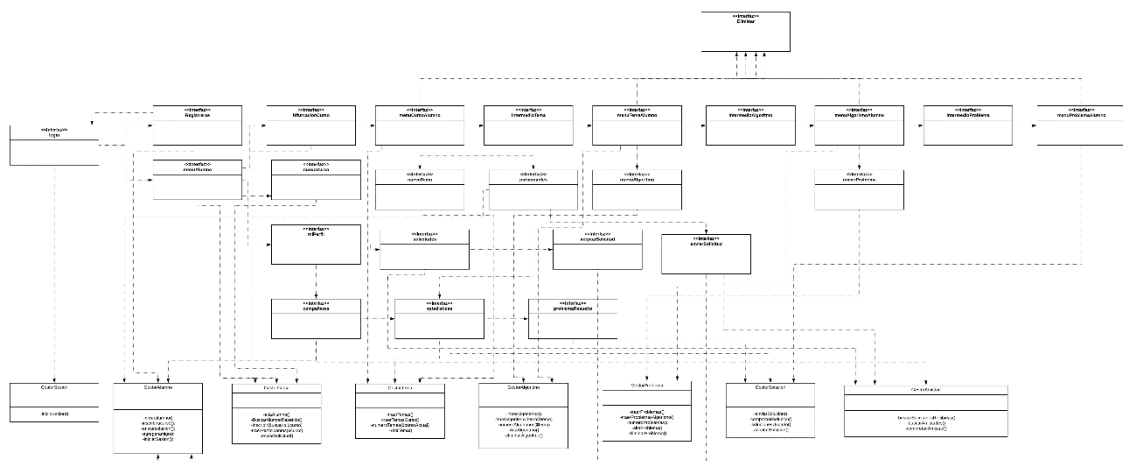


Diagrama de clases Parte 1 (C)



2 Diagrama de Clases (Parte 2)

Nuevamente, se hace énfasis en diferentes partes del diagrama para mejor claridad:

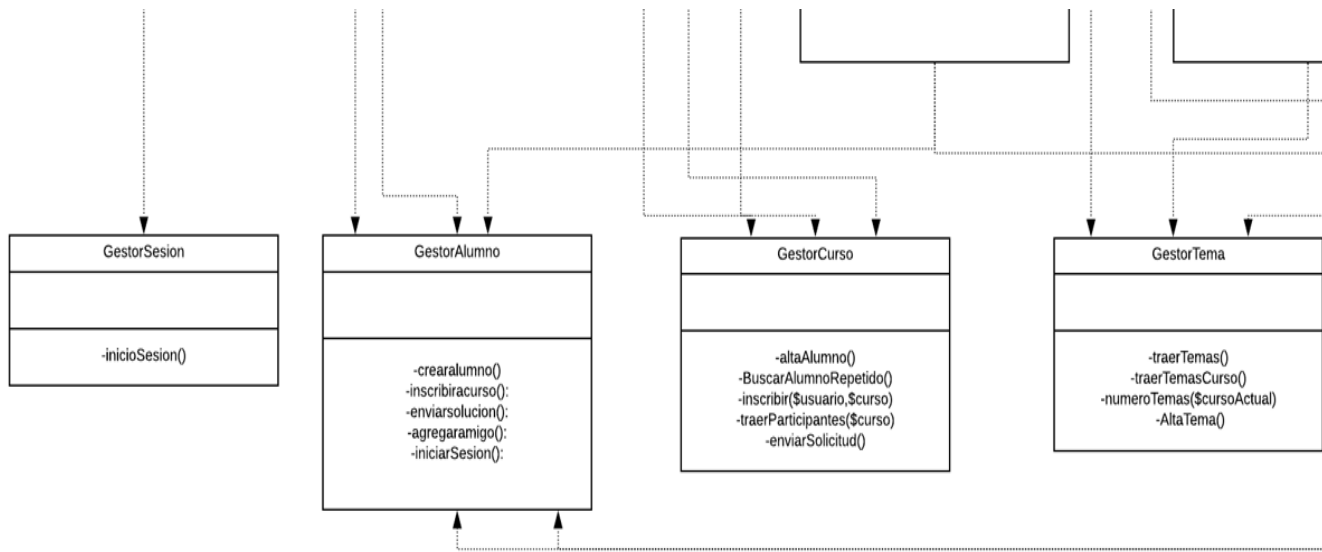


Diagrama de clases Parte 2 (A)

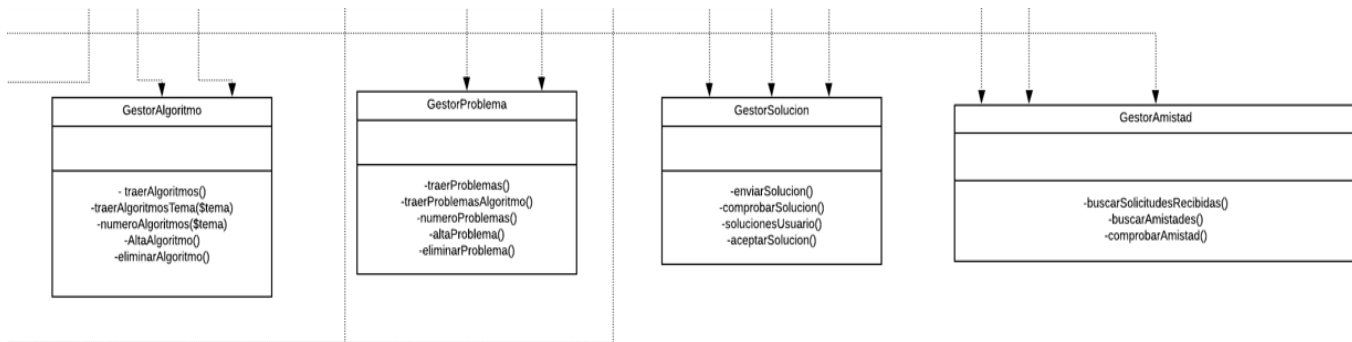


Diagrama de clases Parte 2 (B)

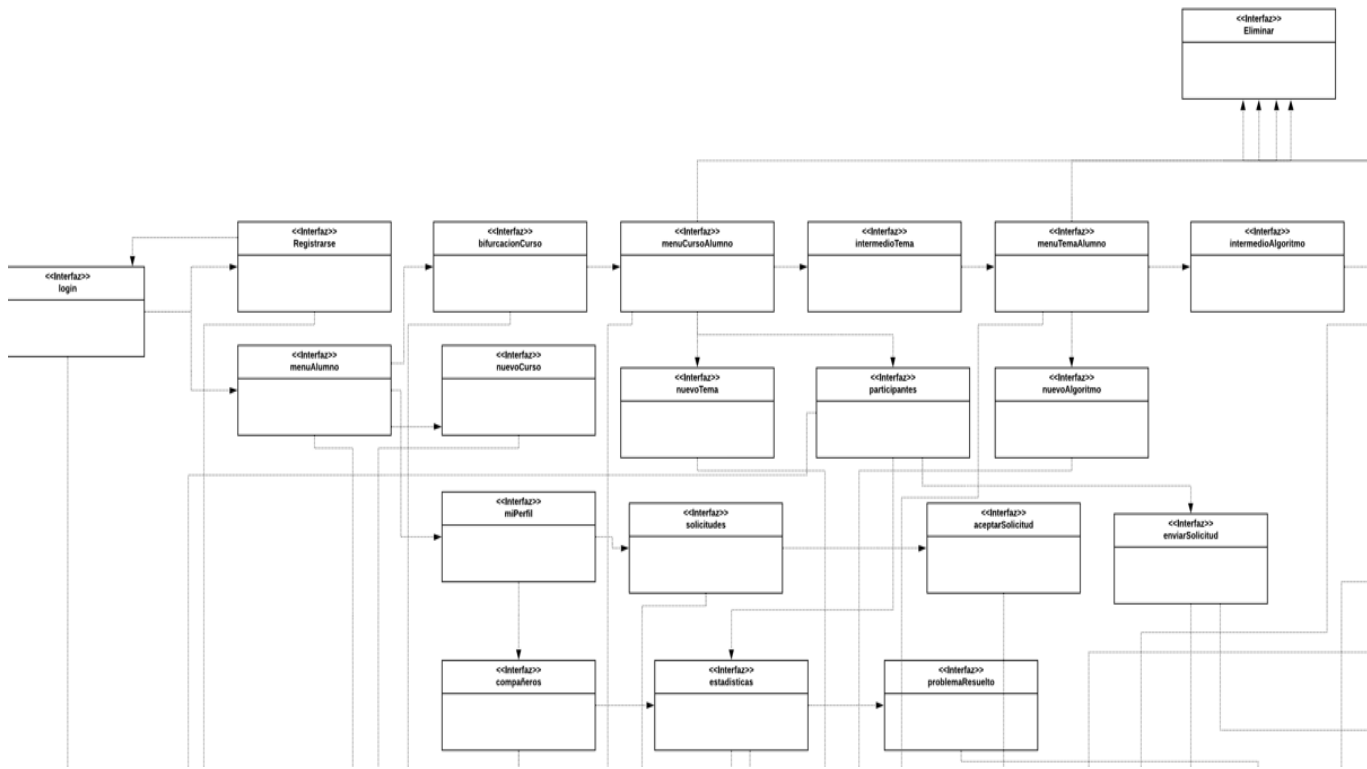


Diagrama de clases Parte 2 (C)

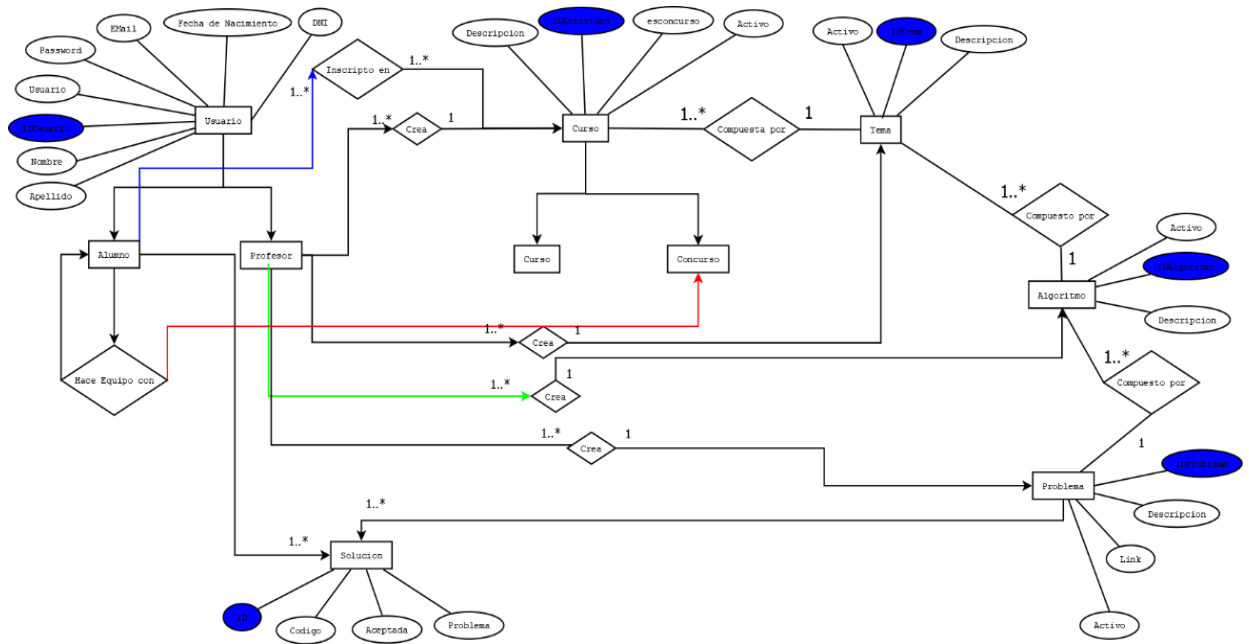


Diagrama de Clases Parte 2 (D)

Como puede verse en las imágenes, lo que se hizo fue crear cada uno de los objetos del sistema (Figura 1A). Luego, se los unió en base a sus relaciones, un curso está compuesto por varios Temas, un Tema por varios Algoritmos, etc. Posteriormente, a cada objeto relevante se le asignó un gestor propio para todas las operaciones lógicas relacionadas a ese objeto (Figuras 1B y 1C). A su vez, a este gestor se le asignó un Gestor de base de datos para todos los accesos a la base de datos relacionados con ese objeto. Todos estos últimos gestores se comunican con el objeto “Database” para poder conectarse con la base de datos.

En la segunda parte del diagrama de clases, se muestra como todos los Gestores se comunican con las distintas interfaces, así como estas se comunican entre ellas (Figuras 2A, 2B, 2C y 2D).

El siguiente es el diagrama de entidad-relaciones (DER) utilizado para modelar la base de datos mysql.



3 Diagrama Entidad-Relación

Para pasar de este DER a las tablas de las bases de datos se utilizaron las siguientes reglas:

- Para las reglas 1 a 1, en la tabla de alguno de los dos objetos se incluía la clave primaria de uno de los objetos como clave foránea en la tabla del otro.
- Para las reglas 1 a N, en la tabla de N se incluía la clave primaria del primer objeto como clave foránea.
- Para las reglas N a N, se crea otra tabla que consiste en ambas claves primarias de ambos objetos insertados como clave foránea. La combinación de estas dos claves foráneas era la clave primaria de la tabla. Si había algún atributo en esta relación, este era incluido en la tabla como una columna más.

2.6 Pruebas funcionales

A medida que se implementaba cada módulo, se realizaron pruebas del mismo. Por ejemplo, al implementar el Caso de Uso “registro de usuario”, se realizaron inmediatamente pruebas del mismo para ver si efectivamente registraba los usuarios, para ver si validaba los campos correctamente, etc. Finalmente, cuando el sistema estuvo terminado, se realizaron pruebas de todo el sistema.

A continuación, se muestra una tabla de las pruebas que se realizaron en esta instancia:

Caso de Uso	Prueba	Resultado
Autenticar Usuario	Se intentó iniciar sesión con un usuario y contraseña válidos.	El sistema entró correctamente.
Autenticar Usuario	Se intentó iniciar sesión con un usuario existente, pero con una contraseña inválida.	El sistema rechazó la entrada y dio el mensaje "Usuario o contraseña incorrecto".
Autenticar Usuario	Se intentó iniciar sesión con un usuario inexistente y contraseña incorrecta.	El sistema rechazó la entrada y dio el mensaje "Usuario o contraseña incorrecto".
Registrar Alumno	Se intentó registrar un alumno con un nombre de usuario ya existente.	El sistema rechazó la entrada y dio el mensaje "Ese nombre de Usuario no está disponible".
Registrar Alumno	Se intentó registrar un alumno y se dejaron los campos DNI e E-mail en blanco.	El sistema rechazó la entrada y dio los mensajes "Por favor complete todos los campos" y "Por favor ingrese un número de DNI válido".
Registrar Alumno	Se intentó registrar un alumno con un nombre de usuario ya existente y además se dejaron vacíos los campos Fecha de Nacimiento y Email.	El sistema rechazó la entrada y dio los mensajes "Por favor complete todos los campos" y "Ese nombre de Usuario no está disponible".
Registrar Alumno	Se intentó registrar un alumno con todos los campos correctos, pero en el campo para confirmar contraseña se puso algo distinto.	El sistema rechazó la entrada y dio los mensajes "Por favor ingrese correctamente la contraseña en ambos campos".
Registrar Alumno	Se intentó registrar un alumno con todos los campos correctos, pero la contraseña era de menos de 6 caracteres.	El sistema rechazó la entrada y dio el mensaje "La contraseña debe ser de más de 6 caracteres de largo".
Registrar Alumno	Se intentó registrar un alumno con todos los campos correctos.	El sistema Registro al alumno y volvió a la pantalla de login.
Añadir Curso	Se intentó añadir un curso dejando en blanco el campo nombre.	El sistema rechazó el alta y dio el mensaje "Por favor complete todos los campos".
Añadir Curso	Se intentó añadir un curso con un nombre de 4 caracteres.	El sistema rechazó el alta y dio el mensaje "El nombre del curso debe

		contener entre 5 y 50 caracteres”.
Añadir Curso	Se intentó añadir un curso con el mismo nombre que uno existente.	El sistema erróneamente dejó añadir otro curso. Me olvidé de realizar esta validación.
Añadir Curso	Se intentó añadir un curso con una descripción de más de 300 caracteres.	El sistema rechazó el alta y dio el mensaje “La descripción debe tener menos de 300 caracteres”.
Añadir Curso	Se intentó añadir un curso con un nombre y descripción apropiados.	El sistema registró el curso correctamente.
Añadir Tema	Se intentó añadir un Tema dejando en blanco el campo nombre.	El sistema rechazó el alta y dio el mensaje “Por favor complete todos los campos”.
Añadir Tema	Se intentó añadir un Tema con un nombre de 4 caracteres.	El sistema rechazó el alta y dio el mensaje “El nombre del Tema debe contener entre 5 y 50 caracteres”.
Añadir Tema	Se intentó añadir un Tema con una descripción de más de 300 caracteres.	El sistema rechazó el alta y dio el mensaje “La descripción debe tener menos de 300 caracteres”.
Añadir Tema	Se intentó añadir un Tema con un nombre y descripción apropiados.	El sistema registró el Tema correctamente.
Añadir Algoritmo	Se intentó añadir un Tema dejando en blanco el campo descripción.	El sistema rechazó el alta y dio el mensaje “Por favor complete todos los campos”.
Añadir Algoritmo	Se intentó añadir un Algoritmo con un nombre de 4 caracteres.	El sistema rechazó el alta y dio el mensaje “El nombre del Algoritmo debe contener entre 5 y 50 caracteres”.
Añadir Algoritmo	Se intentó añadir un Algoritmo con una descripción de más de 300 caracteres.	El sistema rechazó el alta y dio el mensaje “La descripción debe tener menos de 300 caracteres”.
Añadir Algoritmo	Se intentó añadir un Tema con un nombre y descripción apropiados.	El sistema registró el Algoritmo correctamente.
Añadir Problema	Se intentó añadir un Problema dejando en blanco todos los campos.	El sistema rechazó el alta y dio el mensaje “Por favor complete todos los campos”.

Añadir Problema	Se intentó añadir un Problema con un nombre de 4 caracteres.	El sistema rechazó el alta y dio el mensaje “El nombre del Problema debe contener entre 5 y 50 caracteres”.
Añadir Problema	Se intentó añadir un Problema con una descripción de más de 300 caracteres.	El sistema rechazó el alta y dio el mensaje “La descripción debe tener menos de 300 caracteres”.
Añadir Problema	Se intentó añadir un Problema con un nombre y descripción apropiados.	El sistema registró el Problema correctamente.
Eliminar Curso	Se intentó eliminar un curso aleatorio.	El sistema eliminó correctamente el Curso.
Eliminar Tema	Se intentó eliminar un Tema aleatorio.	El sistema eliminó correctamente el Tema.
Eliminar Algoritmo	Se intentó eliminar un Algoritmo aleatorio.	El sistema eliminó correctamente el Algoritmo.
Eliminar Problema	Se intentó eliminar un Problema aleatorio.	El sistema eliminó correctamente el Problema
Enviar Solución de Problema	Un alumno envió una solución de un Problema (imagen y código) en un curso aleatorio.	El sistema aceptó la solución, cuando se entra al Problema nuevamente aparece el mensaje “La solución enviada aún no ha sido aceptada”, lo cual es correcto ya que ningún profesor aceptó la solución aun.
Enviar Solución de Problema	Un alumno envió la solución de un Problema (imagen y código) en un concurso en el cual posee un amigo.	El sistema aceptó la solución, el amigo en la lista de Problemas resueltos por su compañero puede ver el Problema resuelto.
Inscribirse a Curso/Concurso	Un alumno se inscribe a un curso/concurso en el cual no se encuentra inscripto.	El alumno es inscripto correctamente.
Añadir compañero a un concurso en común	Un alumno envía una solicitud de amistad a otro usuario inscripto en la misma competencia y este la acepta.	El sistema registra la amistad y ahora el usuario puede ver los Problemas que resolvió el otro usuario en el concurso en cuestión.
Añadir compañero a un concurso en común	Un alumno envía una solicitud de amistad a otro usuario inscripto en la	La solicitud es rechazada y el usuario no es amigo del otro en ese concurso.

	misma competencia y este la rechaza.	
Aceptar Solución	Un profesor acepta la solución enviada por un alumno.	El Problema es aceptado, cuando el alumno entra de nuevo al Problema, se ve el mensaje “La solución fue aceptada”.
Aceptar Solución	Un profesor rechaza la solución enviada por un alumno.	El Problema es aceptado, cuando el alumno entra de nuevo al Problema, ve “La solución fue rechazada”.

Como se puede ver acá, luego de las pruebas finales se detectó que faltaba una validación en el alta de los cursos.

2.7 Gestión de Riesgos y Contingencias Enfrentadas

Ya descritas en el plan de proyecto, estas fueron las posibles contingencias que podían surgir.

En la tabla que se expone a continuación se describen los riesgos identificados, la probabilidad de ocurrencia y el impacto que podrían tener sobre el proyecto.

Tendremos dos variables, la probabilidad de ocurrencia (P) y el impacto (I), P irá de 0 a 1 e I irá de 1 a 4, representando:

- 1: Ínfima
- 2: Baja
- 3: Media
- 4: Alta

La exposición al riesgo (ER) es calculada por la fórmula $ER = P * I$, lo cual permite ver cuáles son los riesgos más importantes y en cual enfocarse. Así, los riesgos con valor de exposición menor a 1 no se tendrán en cuenta, y de aquellos que se tengan en cuenta, se priorizaron los que tengan mayor puntaje.

Clase	Descripción	P	I	P *I	Estrategia	Medidas de contingencia
RRHH	Paro por enfermedad o razones de fuerza mayor.	0,1	4	0.4	Tolerar	Ninguna, las chances de que suceda son muy pocas, y si suceden, no se puede hacer mucho, simplemente habrá que aplazar la fecha de entrega.
Desarrollo	Falta de experiencia en el desarrollo de este tipo de sistemas.	0,7	3	2.1	Reducir	-Buscar ejemplos de desarrollo de un sistema web parecido ,tal vez más simple, e ir aprendiendo de ahí -Hacer consultas a personas que hayan desarrollado sistemas similares Disparador: desconocimiento elevado de las herramientas para el proyecto.
Requerimientos	Subestimar el aprendizaje que se debe realizar para aprender el lenguaje de programación.	0,8	2	1.6	Reducir	Arrancar cuanto antes a investigar acerca del lenguaje de programación a utilizar, así como tutoriales de desarrollo web en general para poder cumplir con el calendario postulado y no perder tanto tiempo en esta etapa. Disparador: Cuesta realizar hasta las acciones más básicas.
Requerimientos	Subestimar el tiempo que puede tardar un requerimiento en ser implementado.	0.9	2	1.8	Reducir	Tal vez los números que se colocaron en este documento deban ser retocados una vez concluidas las etapas anteriores al inicio del desarrollo, e incluso a medida que se avanza durante el mismo.
Requerimientos	Sobreestimar el tiempo que puede tardar un requerimiento en ser implementado.	0.9	1	0.9	Tolerar	En este caso no se hace nada, simplemente se ajusta el tiempo y se agradece que el proyecto continúe más rápido.
Diseño	La interfaz de la página no es cómoda ni agradable a la vista para el usuario.	0.6	3	1.8	Reducir	Aprender sobre diseño de páginas web con un aspecto y un manejo intuitivo y arrancar el diseño y desarrollo desde esta perspectiva. Disparador: En la primera entrega, el cliente no está conforme con el aspecto y manejo de la página.
Requerimientos	No resolver cómo implementar la subida de soluciones de Problemas, ya sea por captura, integrando con sitios jueces, etc.	0.8	4	3.2	Reducir	Es un problema con alta probabilidad, con lo cual conviene investigarlo antes de comenzar con el desarrollo propiamente dicho. Si no es posible la integración con sitios jueces simplemente deberá usarse la subida por captura. Disparador: Prácticamente sucederá si o si.

Performance	El sistema no puede manejar una gran cantidad de usuarios al mismo tiempo.	0.4	1	0.8	Tolerar	No es probable que la cantidad de usuarios suba a algo inmanejable, siendo que es solo un sistema para esta facultad regional.
RRHH	Ausencia por un tiempo antes de volver a recomenzar el trabajo en el proyecto.	0.7	2	1.4	Tolerar	Reprogramar las fechas de entrega del sistema, agregar días/meses a la fecha estimada Disparador: Viaje, beca, etc.
Requerimientos	Cambio en los requerimientos, los cuales requieren implementación.	0.8	4	3.6	Reducir	Hablar cada cierto tiempo con el director del proyecto, de esa forma se logra anticipar los cambios antes de que estos puedan afectar de manera muy severa el desarrollo del proyecto Disparador: El cliente requiere cambios luego de ya iniciada la implementación.

Una vez terminado el proyecto, las contingencias que debieron enfrentarse fueron:

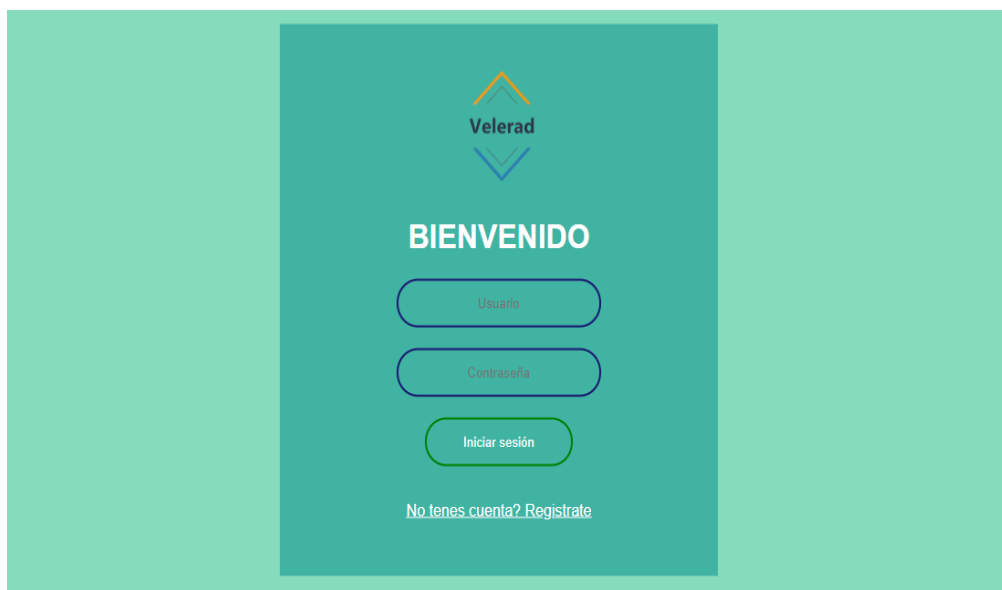
- Falta de experiencia en el desarrollo de este tipo de sistemas y subestimar el aprendizaje que se debe realizar para aprender el lenguaje de programación.
Estas eran, prácticamente, las dos principales contingencias a enfrentar. La falta de experiencia en desarrollo web generó la necesidad de aprender de cero los lenguajes (PHP/HTML5/CSS) así como la dinámica del desarrollo web en sí. Sin embargo, gracias a cursos online, así como ejemplos de desarrollo web que iban desde lo más básico a cosas más complicadas, se encontró la solución a este problema. Esta subestimación del tiempo para adquirir los conocimientos hizo que se atrase unos meses el desarrollo.
- Sobreestimar el tiempo que puede tardar un requerimiento en ser implementado.
Este se explica bastante viendo la tabla de tiempos estimados vs tiempos reales. En muchos requisitos el tiempo de implementación fue sobreestimado.
- No saber cómo implementar la subida de soluciones de Problemas.
Si bien esto sucedió, la solución sugerida por el director del proyecto fue que los usuarios suban la captura y que, opcionalmente, puedan subir el código del Problema.

2.8 Puesta en marcha

Una vez que estuvo todo terminado, lo único que faltaba hacer era subir los archivos del sistema al dominio www.tecnomate.xyz/proyecto, el cual fue provisto por la facultad, a través de cPanel, así como hacer una copia de la base de datos para copiarla en el servidor provisto por la facultad. Esta parte fue sin mucho incidente ya que no hubo que salir a buscar un servidor y un dominio.

3. Interfaz Grafica

A continuación, se ven capturas de pantalla del sistema en Google Chrome, tomadas desde una notebook con Windows 10.



1 Pantalla de Login

Pantalla de login en la cual un usuario, Profesor u Alumno, deben ingresar su nombre de usuario y contraseña para iniciar sesión. En caso de no tener usuario registrado pueden hacer click en la opción de registrarse

The registration screen, titled "BIENVENIDO", features a vertical stack of input fields on a teal background. From top to bottom, the fields are: "Nombre", "Apellido", "Usuario", "Contraseña", "Confirmar contraseña", "Email", "Fecha de nacimiento:" (with a date picker showing "dd/mm/aaaa"), and "DNI". A "Registrarse" button is positioned at the bottom of the form.

2 Pantalla de Registro

Pantalla de registro de usuario, se piden los datos necesarios para la cuenta y en caso de estar todo correcto, se registra la misma.

The main student screen, titled "CURSOS", displays a list of four course options within a teal container: "Algoritmos 2020", "Paradigmas 2020", "AEED 2019", and "Paradigmas 2019". Below the list is a pagination bar with five circular buttons labeled "1", "2", "3", "4", and "5", with "1" being the active page. A "Mi Cuenta" button is located at the bottom of the screen.

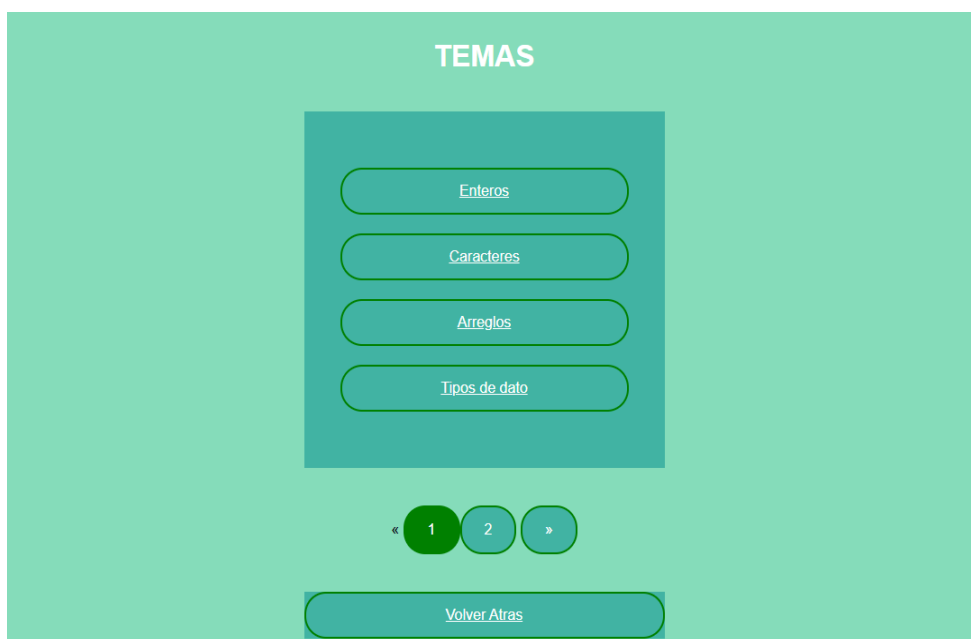
3 Pantalla principal – Estudiante

La pantalla que se le presenta a un alumno luego de iniciar sesión. Puede entrar a uno de los cursos disponibles o acceder a su cuenta.

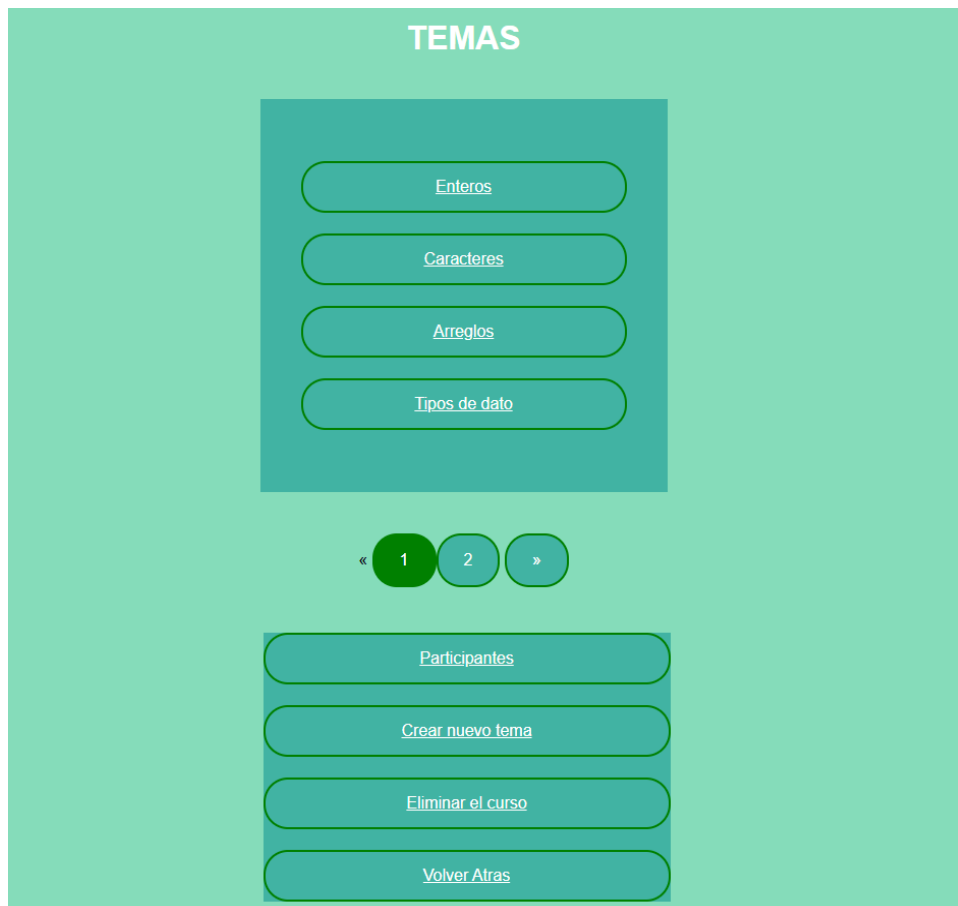


4 Perfil de Usuario

Pantalla de perfil de usuario, se muestran los datos del mismo, además de botones para acceder a las solicitudes de amistad pendientes y a los problemas que resolvieron sus compañeros de concurso (Pantalla 11).



5 Alumno dentro de un curso

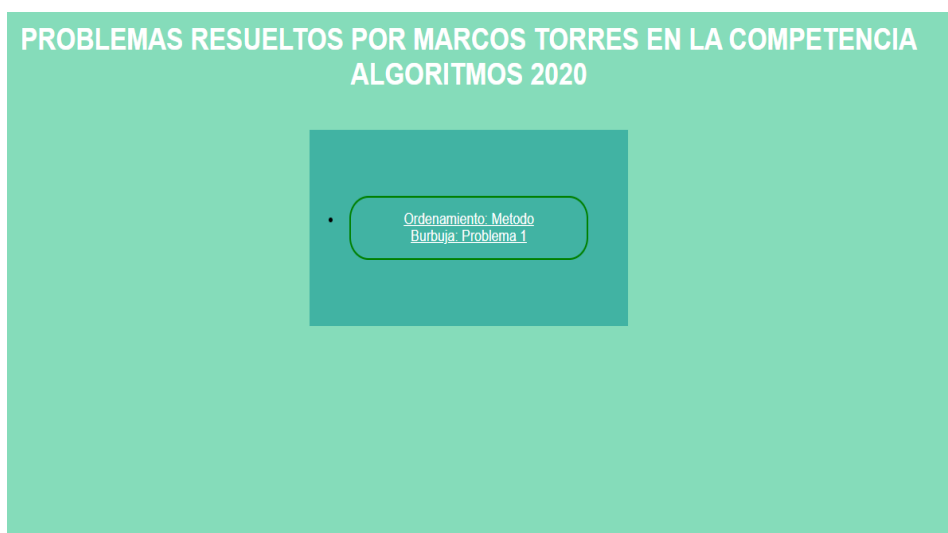


6 Profesor dentro de un curso

Las pantallas 5 y 6 son el menú que se presenta dentro de un Curso, la diferencia entre ambas es que el profesor tiene las opciones de crear un nuevo Tema y eliminar el Curso, así como también ver los participantes de este curso y los problemas que estos resolvieron para poder aprobarlos o no (Pantalla 7 y 8).



7 Participantes de un curso



8 Problemas resueltos por un alumno dentro de un curso



9 Dentro de la solución de un alumno (aceptada)

Pantalla de una solución enviada a la que puede entrar un profesor, en este caso, la solución ya fue aceptada, con lo cual solo está disponible la opción de descargar la captura del juez online y el código.



10 Dentro de la solución de un alumno (no aceptada y con código enviado)

Misma pantalla que 9, solo que en este caso la solución todavía no fue aceptada, se puede descargar la captura, descargar el código o aceptar la solución.



11 Pantalla de creación de un nuevo tema

Pantalla de creación de un nuevo Tema. Se pide simplemente el nombre del mismo y una breve descripción. Esta pantalla es igual para la creación de Cursos y algoritmos, en la de problemas solamente se solicita adicionalmente el link del mismo en la página del juez online.



12 Lista de compañeros de curso



13 Problemas resueltos por un compañero

Lista de problemas resueltos por un compañero en un concurso en el que se tiene una amistad. El usuario actualmente logeado es amigo de Marcos Torres en el concurso Tecnomate 2019 por lo que puede ver los problemas que este resolvió del mismo.

4. Herramientas utilizadas

En esta sección se describe brevemente cuáles fueron las herramientas que se utilizaron en el desarrollo del proyecto.

NetBeans



1 Logo de NetBeans

Para la escritura y compilado del código en sí, se utilizó el entorno de desarrollo NetBeans. Si bien está orientado principalmente a java, se utilizó el PHP pack, el cual soporta PHP 5.

XAMPP



2 Logo de XAMPP

Para el testeo del código escrito y la base de datos de prueba , se utilizó XAMPP. XAMPP es un paquete que contiene el sistema de gestión de base de datos mysql, un servidor web apache y el intérprete de los lenguajes PHP y Perl. Con esto, se pudo alojar la página y la base de datos en un servidor apache de una computadora personal (localhost) para poder testear y hacer modificaciones a la base de datos antes de finalmente subir la página al servidor de la facultad.

Lenguajes utilizados para el Front End

- HTML5: Quinta versión de este lenguaje utilizado para la estructuración de páginas web.
- Css: Lenguaje utilizado para definir el estilo y la apariencia de las páginas escritas con HTML o los documentos XML.



3 Logos de HTML5 y CSS3

Lenguajes utilizados para el Back-End

- PHP: Lenguaje de programación de uso general que se adapta especialmente al desarrollo web.
- MySQL: Sistema de gestión de bases de datos relacional.



4 Logos de PHP y MYSQL

Servidor



5 Logo de CPANEL

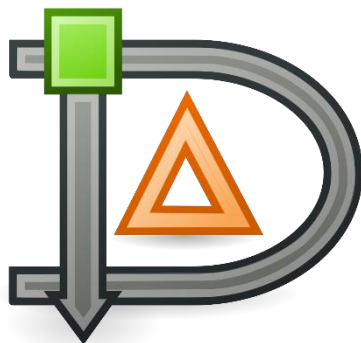
Se utilizó un servidor provisto por la facultad, cuya administración y manejo se hace mediante el software cPanel.

Soporte

- Google Drive: Herramienta en la cual se pueden subir, descargar y editar documentos de manera colaborativa. Como el desarrollo fue unipersonal, la mayor función de Google drive fue que el director del proyecto pueda ver cómo iba el desarrollo del mismo a medida que se iban actualizando los documentos.
- Microsoft Word: Herramienta de edición de documentos, en ella se escribieron el plan de proyecto, el documento de casos de usos y este mismo informe final.
- Lucidchart: Herramienta de diseño y edición de diagramas UML. Con esta herramienta se modeló el diagrama de clases.
- DIA: Otra herramienta de edición de diagramas UML. Con esta se modeló el Diagrama de Entidad-Relaciones.



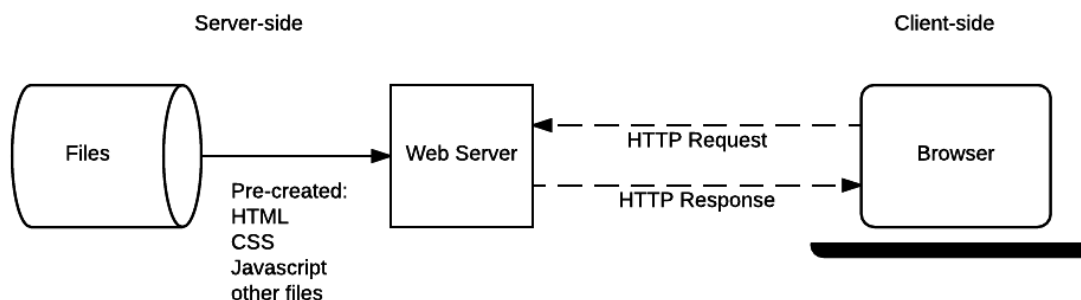
 **Lucidchart**



6 Logos de Google Drive, Microsoft Word, LucidChart y DIA

5. Arquitectura

Con respecto a la arquitectura general de la aplicación, esta puede ser explicada mediante cualquier gráfico genérico de una aplicación web comunicándose con el servidor correspondiente. Como vemos en la siguiente imagen:



1 Gráfico genérico del funcionamiento de una página web

Vemos que el usuario, el cual puede ser un profesor o un alumno, envía paquetes HTTP al servidor de la facultad. Luego de procesar estos paquetes, pudiendo o no acceder a la base de datos, el servidor le envía la respuesta al usuario.

6. Trabajos Futuros

Al definir el alcance del sistema, se consideró inicialmente que solo se iba a hacer uso del mismo por parte de los alumnos de la UTN FRSF. Es por eso que, al decidir la arquitectura del sistema, se decidió por una página web desarrollada en php con base de datos MYSQL que se comunique con la misma

Sin embargo, en caso de que la puesta en uso del sistema sea un éxito, se puede considerar la posibilidad de que la implementación de este se extienda a otras facultades de la UTN.

Esto conllevaría varios desafíos. Primero, habría que migrar la base de datos a otra mas preparada para aguantar un volumen de usuarios mas elevado. Lo ideal seria una como Microsoft Azure Enterprise, aunque quizás se deberían buscar alternativas más económicas debido a su costosa licencia.

Otra cosa que hay que considerar es la posibilidad de que sea necesario adaptar el sistema a las necesidades de cada facultad, ya que, si bien el desarrollo inicial fue hecho solo considerando la perspectiva de la FRSF, otras podrían tener distintos requisitos para un sistema de este tipo. Lo ideal seria tener un sistema principal en el que se encuentren las principales funcionalidades y extensiones específicas para cada facultad.

Por el lado de la programación, se realizaron todas las funcionalidades en php, pero para poder implementar los requisitos que puedan surgir en un futuro con mas facilidad sería ideal utilizar Javascript, para que la pagina pueda tener comportamientos que no necesiten del procesamiento del servidor y puedan ejecutarse del lado del cliente.

Por último, si bien la pagina web se visualiza bien en teléfonos móviles, no estaría fuera de las posibilidades pensar en una versión, si bien mas reducida, del sistema en forma de aplicación para celulares. Esto tendría mayor utilidad quizás para los profesores que para los alumnos, ya que el proceso de resolución de problemas y la subida de estos a los jueces online no se suele hacer por celular, pero la vista y corrección de los problemas podría hacerse sin problemas en un móvil.

7. Conclusión

Este fue mi primer proyecto de desarrollo web. De todas las ideas que me habían sido presentadas, termine decantándome por esta por dos razones: la primera fue que para poder llevar a cabo el proyecto iba a tener que aprender desarrollo web, lo cual es una habilidad que me puede ser inmensamente útil laboralmente en el futuro, y la segunda, me gusto en sí la idea de una herramienta enfocada en ayudar a los alumnos ingresantes y a aquellas personas que quieran participar de concursos de programación.

Con respecto al proyecto en sí, la parte más difícil fue el aprendizaje para ponerlo en marcha. Me tomó un par de meses ver cursos y practicar para poder sentirme con confianza con respecto a mis conocimientos. Pero si bien fue la parte más difícil, creo que fue la más satisfactoria, ya que pude aprender por mí mismo lo necesario para poder encarar el desarrollo de la herramienta.

Afortunadamente, en el desarrollo no tuve demasiados inconvenientes. Es cierto que la pandemia afectó en cierta medida la forma de comunicarme con el director del proyecto; por ejemplo, el no pudo ir probando el proyecto hasta que no lo subí al servidor de la facultad, y para esta etapa el proyecto estaba prácticamente terminado.

En la escritura de código en sí, me encontré con unos bloqueos en los cuales no sabía cómo avanzar, pero nada que me haya tomado más que un par de días. En la definición de requisitos tampoco hubo demasiados problemas, ya que incluso tuve que eliminar una iteración porque no era necesario refinar más los casos de uso. Tal vez sobreestime el esfuerzo que iban a tomar algunos casos de uso y subestime otros, pero, nuevamente, esto no fue algo que afectara desmedidamente el desarrollo.

En definitiva, creo que puedo afirmar que embarcarme en este proyecto fue una idea acertada, adquirí conocimientos que me servirán a futuro y además pude ayudar a la facultad en la que inicié la carrera de Ingeniería en Sistemas de Información. Adquirí experiencia acerca de cómo manejar los tiempos en un

proyecto, como comunicarme con un cliente para poder entender precisamente qué es lo que necesita y cómo adaptarme a los cambios en función en esas necesidades.

8. Bibliografía

-Pressman, Ingeniería de Software, Un Enfoque Práctico – 5ta edición

-Apuntes de las cátedras: Análisis de Sistemas, Diseño de Sistemas, Métodos Ágiles, Ingeniería de Software

-Tutorial PHP y MySQL completo FalconMasters:

<http://www.falconmasters.com/cursos/nuevo-curso-de-php-y-mysql/>

-Curso básico HTML FalconMasters:

https://www.youtube.com/watch?v=cqMfPS8jPys&list=PLhSj3UTs2_yVHt2DgHky_MzzRC58UHE4z&ab_channel=FalconMasters

-Curso CSS FalconMasters:

https://www.youtube.com/watch?v=24gNhTcy6pw&list=PLhSj3UTs2_yU0fGoS1bjpHqky4kCEmTbR&ab_channel=FalconMasters

-Store and retrieve images from MySQL database using PHP:

<https://www.codexworld.com/store-retrieve-image-from-database-mysql-php/>

-How to insert image in MySQL using PHP:

<https://www.studentstutorial.com/php/php-insert-image>

-Resolución a Problemas varios:

www.stackoverflow.com

-Documentación de PHP:

<https://www.php.net/>

9. Anexo

Especificaciones de Casos de Uso obtenidas en la etapa de Análisis.

Número: 1	Nombre: Registrar Profesor
Actores: Administrador web Profesor no registrado	
Objetivo: Registrar un nuevo usuario del tipo profesor en la página.	
Precondiciones: -	
Postcondiciones: <ul style="list-style-type: none"> ● Éxito: Se registró el usuario con éxito. ● Fracaso: El usuario no pudo ser registrado. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. Se realiza un Pedido al Administrador de sistema para registrar un nuevo usuario de tipo Profesor. Este pedido debe contener todos los datos relevantes del usuario y estar firmado por el jefe de cátedra. 2. Se valida el pedido. 3. El pedido es aceptado, el Administrador carga los datos del nuevo usuario en la base de datos, le asigna una contraseña al usuario, la cual, una vez que se le comunican sus datos, se le recomienda cambiar. 4. El Caso de Uso Termina. 	<ol style="list-style-type: none"> 3.El pedido es Rechazado, se le comunican las razones al solicitante y finaliza el Caso de Uso.
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales: Los datos que deben registrarse son:	

<ul style="list-style-type: none"> -Nombre -Apellido -Nombre de Usuario -Contraseña -Email -DNI -Fecha de Nacimiento
<p>Observaciones: A diferencia del registro de alumnos, el registro de profesor no se realiza por el sistema por motivos de seguridad, así se evita que algún usuario no autorizado obtenga los privilegios correspondientes a un profesor.</p>

Número: 2	Nombre: Registrar alumno
Actores: Alumno no registrado	
Objetivo: Registrar un nuevo usuario del tipo alumno en la página web.	
Precondiciones: -	
Postcondiciones: <ul style="list-style-type: none"> ● Éxito: El usuario fue Registrado Correctamente. ● Fracaso: El usuario no fue registrado correctamente. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. El alumno desea registrarse en la página. 2. El sistema presenta la pantalla de registro de usuario, solicitando los siguientes datos: <ul style="list-style-type: none"> ● Nombre ● Apellido ● Nombre de Usuario ● Contraseña ● DNI ● Fecha de Nacimiento ● E-mail 3. El usuario ingresa los datos y estos son validados. 4. Si los datos son correctos, el sistema presenta la pantalla principal. 	<ol style="list-style-type: none"> 3. Alguno de los datos es inválido. <ol style="list-style-type: none"> 3.1. El sistema muestra el mensaje de error correspondiente. 3.2. El sistema blanquea los campos incorrectos. 3.3. El Caso de Uso continúa en el paso 2.

5. El Caso de Uso finaliza.	
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales: <ul style="list-style-type: none"> • El nombre de usuario debe tener una longitud máxima de 20 caracteres, el nombre y apellido de 40 caracteres cada uno y la contraseña deberá ser de mínimo 6 caracteres y poseer al menos un dígito. • Los E-mails aceptados serán: Outlook, Gmail, Hotmail y el dominio de la facultad. 	
Observaciones: <ul style="list-style-type: none"> • Ningún usuario puede autenticarse (CU3) si no tiene un usuario. 	

Número: 3	Nombre: Autenticar Usuario
Actores: Usuario	
Objetivo: Iniciar Sesión en el sistema	
Precondiciones: El usuario debe tener ya una cuenta en el sistema	
Postcondiciones: <ul style="list-style-type: none"> • Éxito: Se muestra la pantalla principal del sistema. • Fracaso: El sistema no puede ser utilizado. 	
Flujo Principal	Flujo Alternativo

<ol style="list-style-type: none"> 1. El usuario desea utilizar la página web. 2. La página web presenta la pantalla para autenticar usuario, solicitando: <ul style="list-style-type: none"> ● Usuario ● Password 3. El sistema valida dicha información. 4. El sistema presenta la pantalla principal. 5. El CU termina. 	<ol style="list-style-type: none"> 3. El usuario selecciona la opción “No tengo cuenta-Registrarme” y se continua en el CU2. 4.1. El nombre de usuario o la contraseña son incorrectos. 4.2. Se muestra el mensaje de error correspondiente. 4.3. Se regresa al paso 2.
<p>Asociaciones de Inclusión:</p>	
<p>Asociaciones de extensiones:</p>	
<p>Requerimientos Especiales:</p> <ul style="list-style-type: none"> ● La contraseña y la clave se visualizarán con asteriscos (*). 	
<p>Observaciones:</p> <ul style="list-style-type: none"> ● Ningún usuario puede realizar ninguna tarea del sistema si no se autentica. 	

<p>Número: 4</p>	<p>Nombre: Añadir Curso</p>
<p>Actores: Profesor Autenticado</p>	
<p>Objetivo: Añadir nuevo curso en el que se registrarán los alumnos.</p>	
<p>Precondiciones: El usuario debe estar autenticado.</p>	
<p>Postcondiciones:</p> <ul style="list-style-type: none"> ● Éxito: Se agregó un nuevo curso al sistema. ● Fracaso: No se pudo agregar un nuevo curso al sistema. 	

Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. Un profesor desea agregar un nuevo curso al sistema, haciendo click en el botón “Agregar Curso”. 2. El sistema muestra una pantalla en la que se solicita que se ingrese el nombre del curso y una breve descripción del mismo. 3. Se valida el nombre del curso. 4. El curso es agregado y el Caso de Uso termina. 	<ol style="list-style-type: none"> 4.El nombre del curso ya existe o es invalido, con lo cual se envía el correspondiente mensaje de error y se vuelve al paso 2.
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales: <ul style="list-style-type: none"> ● El nombre del curso debe ser entre 10 y 50 caracteres, y la descripción del mismo no debe pasar los 300 caracteres. ● Debe registrarse cuál fue el usuario que realizó el registro. 	
Observaciones:	

Número: 5	Nombre: Añadir Tema
Actores: Profesor Autenticado	
Objetivo: Añadir nuevo Tema para el aprendizaje y evaluación de los alumnos dentro de un curso.	
Precondiciones: Debe haber al menos un curso creado.	
Postcondiciones: <ul style="list-style-type: none"> ● Éxito: Se agregó un nuevo Tema al sistema. ● Fracaso: No se pudo agregar un nuevo Tema al sistema. 	
Flujo Principal	Flujo Alternativo

<ol style="list-style-type: none"> 1. Un profesor desea agregar un nuevo Tema dentro de determinado curso, haciendo click en el botón “Agregar Tema” dentro de la página del curso. 2. El sistema muestra una pantalla en la que se solicita que se ingrese el nombre del Tema y una breve descripción del mismo. 3. Se valida el nombre del tema. 4. El Tema es agregado al curso y el Caso de Uso termina. 	<p>4.El nombre del Tema ya existe en ese curso o es invalido, con lo cual se envía el correspondiente mensaje de error y se vuelve al paso 2.</p>
<p>Asociaciones de Inclusión:</p>	
<p>Asociaciones de extensiones:</p>	
<p>Requerimientos Especiales:</p> <ul style="list-style-type: none"> ● El nombre del Tema debe ser entre 10 y 50 caracteres, y la descripción del mismo no debe pasar los 300 caracteres. ● Debe registrarse cuál fue el usuario que realizó el registro. 	
<p>Observaciones:</p>	

<p>Número: 6</p>	<p>Nombre: Añadir Algoritmo</p>
<p>Actores: Profesor Autenticado</p>	
<p>Objetivo: Añadir nuevo Algoritmo dentro de un tema.</p>	
<p>Precondiciones: Deben existir al menos un curso y un Tema dentro del mismo.</p>	
<p>Postcondiciones:</p> <ul style="list-style-type: none"> ● Éxito: Se agregó un nuevo Algoritmo al tema. ● Fracaso: No se pudo agregar un nuevo curso al tema. 	
<p>Flujo Principal</p>	<p>Flujo Alternativo</p>

<ol style="list-style-type: none"> 1. Un profesor desea agregar un nuevo Algoritmo al sistema, haciendo click en el botón “Agregar Algoritmo” dentro del tema. 2. El sistema muestra una pantalla en la que se solicita que se ingrese el nombre del Algoritmo y una breve descripción del mismo. 3. Se valida el nombre del Algoritmo. 4. El Algoritmo es agregado y el Caso de Uso termina. 	<p>4.El nombre del Algoritmo ya existe en ese Tema o es invalido, con lo cual se envía el correspondiente mensaje de error y se vuelve al paso 2.</p>
<p>Asociaciones de Inclusión:</p>	
<p>Asociaciones de extensiones:</p>	
<p>Requerimientos Especiales:</p> <ul style="list-style-type: none"> ● El nombre del Algoritmo debe ser entre 10 y 50 caracteres, y la descripción del mismo no debe pasar los 300 caracteres. ● Debe registrarse cuál fue el usuario que realizó el registro. 	
<p>Observaciones:</p>	

<p>Número: 7</p>	<p>Nombre: Añadir Problema</p>
<p>Actores: Profesor Autenticado</p>	
<p>Objetivo: Añadir nuevo Problema dentro de un Algoritmo.</p>	
<p>Precondiciones: Deben existir al menos un curso, un Tema dentro del mismo y un Algoritmo dentro del tema.</p>	
<p>Postcondiciones:</p> <ul style="list-style-type: none"> ● Éxito: Se agregó un nuevo Problema al Algoritmo. ● Fracaso: No se pudo agregar un nuevo curso al Algoritmo. 	
<p>Flujo Principal</p>	<p>Flujo Alternativo</p>

<ol style="list-style-type: none"> 1. Un profesor desea agregar un nuevo Problema al Algoritmo, haciendo click en el botón “Agregar Algoritmo” dentro del Algoritmo. 2. El sistema muestra una pantalla en la que se solicita que se ingrese el nombre del Algoritmo, una breve descripción del mismo y un link de la página de juez online en la cual debe resolverse el Problema. 3. Se valida el nombre del Problema. 4. El Algoritmo es agregado y el Caso de Uso termina. 	<p>4.El nombre del Problema ya existe en ese tema, es invalido, o el link del mismo ya se encuentra en otro Problema con lo cual se envía el correspondiente mensaje de error y se vuelve al paso 2.</p>
<p>Asociaciones de Inclusión:</p>	
<p>Asociaciones de extensiones:</p>	
<p>Requerimientos Especiales:</p> <ul style="list-style-type: none"> ● El nombre del Algoritmo debe ser entre 10 y 50 caracteres, y la descripción del mismo no debe pasar los 500 caracteres. ● Debe registrarse cuál fue el usuario que realizó el registro. 	
<p>Observaciones:</p>	

<p>Número: 9</p>	<p>Nombre: Eliminar Curso</p>
<p>Actores: Profesor Autenticado</p>	
<p>Objetivo: Eliminar un curso del sistema.</p>	
<p>Precondiciones: Debe existir al menos un curso en el sistema.</p>	
<p>Postcondiciones:</p> <ul style="list-style-type: none"> ● Éxito: El curso fue eliminado del sistema. ● Fracaso: El curso sigue en el sistema. 	
<p>Flujo Principal</p>	<p>Flujo Alternativo</p>

<ol style="list-style-type: none"> 1. El profesor desea eliminar un curso y presiona el Botón “Eliminar Curso” dentro del mismo. 2. El sistema muestra el mensaje “El curso y todos sus contenidos serán eliminados, los alumnos aun así conservarán los registros de los Problemas resueltos de el mismo” y presenta las opciones ACEPTAR-CANCELAR. 3. El profesor presiona aceptar y el sistema elimina lógicamente el curso, los temas Algoritmos y Problemas del mismo. 4. El CU termina. 	<p>3. El profesor presiona Cancelar y el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 10	Nombre: Eliminar Tema
Actores: Profesor Autenticado	
Objetivo: Eliminar un Tema de un curso.	
Precondiciones: Debe existir al menos un curso en el sistema con al menos un Tema dentro.	
Postcondiciones: <ul style="list-style-type: none"> ● Éxito: El Tema fue eliminado del curso. ● Fracaso: El Tema obsoleto aún no fue eliminado. 	
Flujo Principal	Flujo Alternativo

<ol style="list-style-type: none"> 5. El profesor desea eliminar un Tema y presiona el Botón “Eliminar Tema” dentro del mismo. 6. El sistema muestra el mensaje “El Tema y todos sus contenidos serán eliminados, los alumnos aun así conservarán los registros de los Problemas resueltos del mismo” y presenta las opciones ACEPTAR-CANCELAR. 7. El profesor presiona aceptar y el sistema elimina lógicamente el tema, los Algoritmos y Problemas del mismo. 8. El CU termina. 	<ol style="list-style-type: none"> 3. El profesor presiona Cancelar y el Caso de Uso termina.
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 11	Nombre: Eliminar Algoritmo
Actores: Profesor Autenticado	
Objetivo: Eliminar un Algoritmo de un tema.	
Precondiciones: Debe existir al menos un curso en el sistema con un Tema y a su vez un Algoritmo dentro de este.	
Postcondiciones: <ul style="list-style-type: none"> ● Éxito: El Algoritmo fue eliminado del tema. 	

<ul style="list-style-type: none"> ● Fracaso: El Algoritmo sigue dentro del tema. 	
Flujo Principal	Flujo Alternativo
<p>9. El profesor desea eliminar un Algoritmo y presiona el Botón “Eliminar Algoritmo” dentro del mismo.</p> <p>10. El sistema muestra el mensaje “El Algoritmo y todos sus contenidos serán eliminados, los alumnos aun así conservarán los registros de los Problemas resueltos de el mismo” y presenta las opciones ACEPTAR-CANCELAR.</p> <p>11. El profesor presiona aceptar y el sistema elimina lógicamente el Algoritmo y los Problemas del mismo.</p> <p>12. El CU termina.</p>	<p>3. El profesor presiona Cancelar y el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 12	Nombre: Eliminar Problema
Actores: Profesor Autenticado	
Objetivo: Eliminar un Problema de un tema.	
Precondiciones: Debe existir al menos un curso, con un tema, con un Algoritmo con un Problema dentro del mismo.	
Postcondiciones:	

<ul style="list-style-type: none"> • Éxito: El Problema fue eliminado del sistema. • Fracaso: El Problema sigue en el sistema. 	
Flujo Principal	Flujo Alternativo
<p>13. El profesor desea eliminar un curso y presiona el Botón “Eliminar Problema” dentro del mismo.</p> <p>14. El sistema muestra el mensaje “El Problema será eliminado, los alumnos aun así conservarán los registros del mismo” y presenta las opciones. ACEPTAR-CANCELAR</p> <p>15. El profesor presiona aceptar y el sistema elimina lógicamente el Algoritmo.</p> <p>16. El CU termina.</p>	<p>3. El profesor presiona Cancelar y el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 14	Nombre: Enviar Solución del Problema
Actores: Alumno autenticado	
Objetivo: El alumno envía la solución de un Problema que resolvió.	
Precondiciones: El alumno debe estar inscripto en el curso en el que está enviando el Problema, además de tener un usuario del juez online en el cual se encuentra el Problema.	
Postcondiciones:	
<ul style="list-style-type: none"> • Éxito: El alumno suma un Problema resuelto. 	

<ul style="list-style-type: none"> ● Fracaso: El Problema sigue sin resolverse. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. El alumno clickea “Enviar Solución” dentro del Problema. 2. El sistema presenta una ventana para que el Alumno seleccione la captura del Problema resuelto y el código (opcional), en la cual debe ser visible su nombre de usuario. 3. El Alumno sube la captura y clickea “enviar solución”. 4. El CU termina. 	<p>2A. El alumno ya resolvió ese Problema, con lo cual el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
<ul style="list-style-type: none"> ● La captura debe mostrar claramente el nombre de usuario del juez online, y debe ser el mismo que se utilizó para todos los otros Problemas, esto lo puede comprobar un profesor mediante una revisión rápida de las soluciones subidas. 	
Observaciones:	

Número: 15	Nombre: Inscribirse a curso o concurso
Actores: Alumno autenticado	
Objetivo: Poder inscribirse a un curso o concurso para poder realizar Problemas en el mismo.	
Precondiciones: No debe estar inscripto en el curso o concurso seleccionado.	
Postcondiciones:	
<ul style="list-style-type: none"> ● Éxito: El alumno se inscribió en el curso o concurso correctamente. ● Fracaso: El alumno no pudo inscribirse en el curso o concurso. 	

Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. El alumno clickea en el curso que desea inscribirse. 2. El sistema le mostrará el mensaje “Inscribirse en el curso - NOMBRE DEL CURSO/CONCURSO-? “y las opciones ACEPTAR – CANCELAR. 3. El usuario acepta y el sistema lo inscribe en el curso o concurso. 4. El Caso de Uso Termina. 	<ol style="list-style-type: none"> 3. El usuario no acepta y el Caso de Uso termina.
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 16	Nombre: Agregar a un compañero a un concurso en común
Actores: Alumno autenticado	
Objetivo: Se vincularán ambos usuarios con el mismo concurso, y le aparecerán al usuario en su perfil los Problemas resueltos por sus compañeros.	
Precondiciones: Ambos usuarios deben estar inscriptos en el concurso.	
Postcondiciones: <ul style="list-style-type: none"> ● Éxito: Los alumnos están asociados en base a ese curso. 	

<ul style="list-style-type: none"> ● Fracaso: Los alumnos no están asociados. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. Un alumno desea asociarse a otro en un concurso, clickea en “Participantes del concurso” en el concurso actual. 2. El sistema presenta el listado, el usuario selecciona el perfil que desee y clickea en “Enviar solicitud”. 3. El receptor acepta la solicitud y el sistema registra la asociación. 4. El Caso de Uso termina. 	<p>3A. El usuario ya ha enviado una solicitud de amistad con ese curso al otro usuario, ya es amigo en ese curso con el otro usuario o la solicitud que envió fue rechazada, el sistema muestra el mensaje correspondiente y vuelve a la pantalla anterior.</p> <p>4A. Los usuarios ya están asociados en base a ese concurso, se envía el mensaje de error correspondiente y se vuelve al paso 3.</p> <p>4B. El receptor rechaza la solicitud y el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 17	Nombre: Aceptar o rechazar solicitud de otro alumno en un concurso en común
Actores: Alumno autenticado	

Objetivo: Se vincularán ambos usuarios con el mismo concurso, y le aparecerán al usuario en su perfil los Problemas resueltos por sus compañeros o la solicitud será rechazada y no se verán más las amistades.	
Precondiciones: Ambos usuarios deben estar inscriptos en el concurso. El alumno B debe haberle enviado una solicitud de amistad al alumno A.	
Postcondiciones:	
<ul style="list-style-type: none"> • Éxito: Los alumnos están asociados en base a ese curso o simplemente la solicitud es rechazada. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. Un alumno A desea responder a la solicitud de amistad que recibió de otro alumno. 2. El alumno A hace click en “mi cuenta” y luego en “solicitudes de amistad”. 3. El alumno A clickea en la solicitud en cuestión enviada por el alumno B. 4. El alumno A acepta la solicitud, ahora es compañero del alumno B en el concurso de la solicitud y el Caso de Uso termina. 	<p>4A. El alumno A rechaza la solicitud y el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 18	Nombre: Ver Problemas resueltos por un compañero
Actores: Alumno autenticado	
Objetivo: Ver qué Problemas de un concurso determinado resolvió un compañero de concurso.	

Precondiciones: Ambos alumnos deben ser amigos en base al concurso en cuestión.	
Postcondiciones:	
<ul style="list-style-type: none"> ● Éxito: El sistema presenta la lista de Problemas resueltos. ● Fracasó: La lista no pudo ser mostrada. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. Un alumno desea ver los Problemas resueltos por su compañero en el concurso 2. Hace click en “Mi cuenta” y luego en “Compañeros de concurso”. 3. El sistema presenta una lista de los compañeros del alumno con sus respectivos concursos. 4. El alumno clickea en el compañero deseado, el sistema presenta la lista de Problemas resueltos y el Caso de Uso termina. 	
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

Número: 19	Nombre: Cerrar Sesión
Actores: Alumno autenticado	
Objetivo: El usuario puede salir del sistema una vez finalizado el trabajo en la plataforma.	

Precondiciones: El alumno debe estar autenticado.	
Postcondiciones: <ul style="list-style-type: none"> • Éxito: El alumno cerró la sesión de su cuenta. • Fracaso: El alumno no pudo salir del sistema. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. El usuario desea salir del sistema, clickea en “mi perfil”. 2. Dentro de mi perfil, el usuario clickea “Cerrar Sesión”. 3. El usuario sale del sistema y el CU termina. 	
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	

2da Iteración:

Número: 20	Nombre: Aceptar solución
Actores: Profesor	
Objetivo: El profesor acepta una solución de un Problema A enviada por un alumno B.	
Precondiciones: El profesor debe estar autenticado y el alumno A debe haber enviado una solución al Problema B.	
Postcondiciones: <ul style="list-style-type: none"> • Éxito: La solución fue aceptada. 	

<ul style="list-style-type: none"> ● Fracaso: La solución no fue aceptada. 	
Flujo Principal	Flujo Alternativo
<ol style="list-style-type: none"> 1. El profesor clickea en el curso y luego clickea en “Participantes”. 2. El sistema le presenta una lista de los usuarios participantes. 3. El profesor clickea en el alumno y el sistema le presenta una lista de los Problemas que este resolvió en el curso en cuestión. 4. El profesor clickea en una de estas soluciones, el sistema le presenta la opción de descargar la captura y el código de la solución para evaluación y la opción de aceptar la misma. 5. El profesor acepta la solución y el Caso de Uso termina. 	<p>5A El profesor presiona “Volver atrás” y el Caso de Uso termina.</p>
Asociaciones de Inclusión:	
Asociaciones de extensiones:	
Requerimientos Especiales:	
Observaciones:	