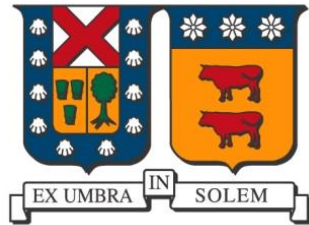


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“DISEÑO DE PLATAFORMA DE PROGRAMACIÓN DE A PARES PARA ESTUDIANTES DEL RAMO IWI-131”

VALENTÍN ANDRÉS RIVEROS CALDERÓN

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Federico Meza
Profesor Correferente: Andrea Vásquez

Diciembre - 2023

DEDICATORIA

Este documento de memoria está dedicado a todas las personas que estuvieron presentes en todos los aspectos de mi vida, muchas gracias a todos.

AGRADECIMIENTOS

Primero que todo, quiero agradecer a mi familia, la cual estuvo siempre presente tanto en mi desarrollo de esta memoria como en toda mi carrera. Mi madre Erika, mi padre Juan y mi hermano Ricardo que fueron y serán mi apoyo incondicional en todos mis proyectos de vida, y me han brindado el cariño que toda persona necesita. Ustedes siempre serán mis ejemplos para seguir y todo se los debo a ustedes.

Agradecer a mi novia Fernanda, la cual estuvo presente en todo el desarrollo de esta memoria y me ha apoyado en todo, brindándome el cariño que necesitaba y, obviamente, el amor que me has dado. Gracias por todos estos momentos maravillosos que hemos pasado durante todo el tiempo que llevamos juntos y los muchos años más que nos toca por disfrutar.

Mis compañeros de la universidad, los cuales éramos un grupo grande, pero agradeceré a cada uno de ellos. Brannndon, Álvaro, Benja Guzmán, Chelo, Exe, Jano, Marcos, Benja Pizarro, Pipe, Tomás y Josué. Todos ustedes fueron un eje principal en toda la carrera, donde pasamos momentos inolvidables y espero que la amistad entre nosotros se mantenga a lo largo de los años. Han sido grandes momentos que nunca está de más recordar. Muchas gracias a cada uno de ustedes.

Mis hermanos de otras madres, Freddy, Cristian, Matías Araya y Matías Guzmán los cuales hemos estado juntos desde muy pequeños y hasta el día de hoy mantenemos una amistad tan grande que parece una hermandad. Agradecerles a todos ustedes por ser parte de mi proceso y brindarme palabras de apoyo para poder terminar mi carrera.

También, a mi familia general, mis tías Gaby y Olga. Mi abuelita Mariana y mi abuelita Leontina y también a mí primos Carlos y Lucas, junto con Pía. Todo este esfuerzo no ha sido en vano y obtendrá los resultados esperados. Muchas gracias por el apoyo y palabras de aliento para seguir adelante.

Por último y no menos importantes, a los profesores del departamento de informática, principalmente al profesor Federico Meza, que es mi profesor guía de esta memoria, y que ha debido tener mucha paciencia conmigo, pero siempre estuvo disponible para poder continuar con el proceso y terminar esta memoria. Muchas gracias por ayudarme en cada uno de los aspectos técnicos del proceso de esta memoria y por tener los tiempos disponibles para el desarrollo completo de este proyecto.

RESUMEN

Resumen— El curso de programación que imparte la Universidad Técnica Federico Santa María busca que los estudiantes de la universidad puedan aprender los conceptos básicos de programación en Python. El curso se compone de actividades individuales relacionadas con la materia, tales como tareas, evaluaciones, actividades de ayudantía, etc. Se ha considerado que la falta de participación entre compañeros puede disminuir la motivación de los estudiantes. En este documento, se presentará una solución de programación de a pares, en donde los estudiantes podrán resolver ejercicios de programación en parejas, aumentando la participación entre estudiantes y la búsqueda de soluciones en parejas. Los estudiantes que participaron de los laboratorios con la solución propuesta consideraron la experiencia como muy buena en general, además de obtener buenos resultados en la siguiente evaluación individual del curso. Con esta plataforma, los estudiantes podrán trabajar en equipos y mantener la motivación necesaria para cursar exitosamente el ramo.

Palabras Clave— Programación de a pares; Plataforma para programar; Herramienta educativa; Metodología educativa.

ABSTRACT

Abstract— *The programming course offered by the Universidad Técnica Federico Santa María for the university students to learn the basic concepts of programming in Python. The course consists of individual activities related to the subject, such as assignments, evaluations, tutoring activities, etc. It has been considered that a lack of participation among peers can decrease student motivation. This document will present a pair programming solution, where students will be able to solve programming exercises in pairs, increasing participation among students and the search for solutions together. Students who participated in the labs with the proposed solution considered the experience to be very good overall, in addition to achieving good results in the subsequent individual evaluation of the course. With this platform, students will be able to work in teams and maintain the necessary motivation to successfully complete the course.*

Keywords— Pair Programming; Platform for Programming; Educational Tool; Educational Methodology.

GLOSARIO

API: Application Program Interface.

CSS: Cascading Style Sheets

IDE: Integrated Development Environment.

LTI: Learning Tools Interoperability

MVC: Modelo-Vista-Controlador.

ORM: Object-Relational Mapping

SMOJ: Santa María Online Judge.

UTFSM: Universidad Técnica Federico Santa María.

UVA: Unidad Virtual de Aprendizaje.

INDICE DE CONTENIDOS

RESUMEN.....	4
ABSTRACT	4
INDICE DE TABLAS.....	11
INTRODUCCIÓN	12
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	15
1.1 Situación actual.....	15
1.2 Información del problema	16
1.3 Propuesta	18
1.4 Objetivos	19
1.4.1 Objetivo General	19
1.4.2 Objetivos Específicos	19
CAPÍTULO 2: MARCO CONCEPTUAL	20
2.1 Programación	20
2.1.1 Lenguaje de Programación	20
2.1.2 Python	20
2.2 Programación de a pares.	21
2.2.1 Estudios de otros autores.	22
2.3 Plataformas de programación de a pares.....	23
2.4 SMOJ.....	27
CAPÍTULO 3: PROPUESTA DE SOLUCION	29
3.1 Desarrollo del Backend	29
3.1.1 Modelado de datos	30

3.1.2	Autenticación	31
3.1.3	Roles y permisos.....	32
3.2	Desarrollo de plataforma	32
3.2.1	Estudiantes	33
3.2.2	Profesores	36
3.2.3	Sistema	38
3.3	Desarrollo del diseño	41
3.3.1	Vista de inicio	41
3.3.2	Vista de talleres.....	42
3.3.3	Vista de ejercicios.....	43
3.3.4	Vista de ejercicio específico	44
3.3.5	Vista de código	45
3.3.6	Vista de envió	46
3.3.7	Vista de diseño de talleres	48
3.3.8	Vista de paralelos	49
3.3.9	Vista de selección de paralelos	50
CAPÍTULO 4:	VALIDACIÓN DE LA SOLUCIÓN	52
4.1	Habilitación de prototipo	52
4.2	Habilitación de plataforma antes de la segunda evaluación del curso	53
4.2.1	Resultados generales	53
4.2.2	Resultados de envíos realizados	53
4.3	Encuesta a los estudiantes.....	54
4.3.1	Resultados de la encuesta.....	55
4.4	Resultados en base a evaluación del curso	57

CAPÍTULO 5: CONCLUSIONES	59
5.1 Consideraciones y limitaciones	60
5.2 Trabajos a futuro	61
REFERENCIAS BIBLIOGRÁFICAS.....	62

INDICE DE FIGURAS

Figura 1: Metodología y modelo formativo del ramo IWI131. Fuente: Programa de asignatura IWI131.....	15
Figura 2: Árbol del problema. Fuente: Elaboración propia.	18
Figura 3: Ejemplo de Python. Fuente: Introducción a la programación.....	20
Figura 4: Motepair by Atom. Fuente: Atom.	24
Figura 5: Visual Studio Live Share. Fuente: Microsoft.....	25
Figura 6: Codeanywhere. Fuente: Codeanywhere.	26
Figura 7: CodeTogether. Fuente: CodeTogether.....	27
Figura 8: Santa María Online Judge. Fuente: SMOJ.	28
Figura 9: Modelado de datos. Fuente: Elaboración propia.....	31
Figura 10: Obtención del ejercicio. Fuente: Elaboración Propia	34
Figura 11: Parejas actuales. Fuente: Elaboración Propia	37
Figura 12: Parejas luego de la modificación. Fuente: Elaboración Propia	38
Figura 13: Metodología de programación de a pares. Fuente: Elaboración Propia	40
Figura 14: Vista de inicio (estudiante). Fuente: Elaboración Propia	42
Figura 15: Vista de inicio (administrador). Fuente: Elaboración Propia.....	42
Figura 16: Vista de talleres. Fuente: Elaboración Propia	43
Figura 17: Vista de ejercicios. Fuente: Elaboración Propia	43
Figura 18: Vista de ejercicio específico. Fuente: Elaboración Propia.....	45
Figura 19: Vista de código. Fuente: Elaboración Propia.....	46
Figura 20: Vista de probar código. Fuente: Elaboración Propia.....	46
Figura 21: Vista de envió específico. Fuente: Elaboración Propia	47
Figura 22: Vista de mis envíos. Fuente: Elaboración Propia	48

Figura 23: Vista de diseño de talleres. Fuente: Elaboración Propia.....	49
Figura 24: Vista de paralelos. Fuente: Elaboración Propia.....	49
Figura 25: Vista de parejas. Fuente: Elaboración Propia.....	50
Figura 26: Vista de edición de parejas. Fuente: Elaboración Propia	50
Figura 27: Vista de selección de paralelos. Fuente: Elaboración Propia.....	51

INDICE DE TABLAS

Tabla 1: Resultados generales. Fuente: Elaboración Propia	53
Tabla 2: Resultados de envíos realizados. Fuente: Elaboración Propia	54
Tabla 3: Resultados veredicto de Judge0. Fuente: Elaboración Propia	54
Tabla 4: Resultados de experiencia. Fuente: Elaboración Propia	55
Tabla 5: Resultados de experiencia con compañero. Fuente: Elaboración Propia	55
Tabla 6: Resultados de facilidad de uso. Fuente: Elaboración Propia.....	56
Tabla 7: Resultados de metodología. Fuente: Elaboración Propia	56
Tabla 8: Resultados de entendimiento del problema. Fuente: Elaboración Propia	56
Tabla 9: Resultados de recomendación. Fuente: Elaboración Propia.....	56
Tabla 10: Resultados de certamen 2. Fuente: Elaboración Propia	57

INTRODUCCIÓN

Una de las características fundamentales de todo estudiante es la motivación que este tiene al momento de realizar un curso específico. En mi caso específico, cuando elegí la carrera de Ingeniería Civil Informática, escogí la carrera que más se me acomodaba a mis gustos y cuáles eran los ramos que más me motivaban a estudiar.

Según (Polanco, 2005), *“la falta de motivación es una de las causas importantes que se debe valorar en el fracaso de los estudiantes”*. La motivación es relevante, es como los estudiantes van a percibir el aprendizaje, y si es que ellos quieren aprender lo relacionado con el curso propuesto.

El compromiso que tienen los estudiantes con el curso también tiene relación con la motivación que ellos tienen. Si un estudiante quiere aprender a programar, debe estar motivado en programar. Si un estudiante tiene la obligación de programar, este estudiante hará el curso por compromiso. Pero, eso no significa que el estudiante pueda fracasar en el intento.

Según (Rodríguez, 2021), el *“miedo al compromiso se produce por temor a perder la libertad o tener que sacrificar aspectos de la propia vida”*. Una persona que quiere estudiar un curso en particular, lo hará con mucho entusiasmo y motivación. En el caso contrario, una persona puede actuar basándose en el compromiso, es decir, estudiar un ramo en donde la persona no tiene nada de motivación en hacerlo.

En la UTFSM, se realizan ramos comunes para todas las carreras de Ingeniería, tales como Química, Matemática, Física, Ed. Física y, el ramo por el cual se hará estudio en esta memoria, Programación. (UTFSM, 2023).

Todas las carreras de Ingeniería en la universidad deben cursar estos ramos. Pero, muchas de estas carreras no tienen nada relacionado con la informática, por lo cual, cursar un curso de Programación para ellos es más un compromiso que por motivación, por lo tanto, es más probable que estudiantes de otras carreras reprueben el curso por falta de motivación.

Lo mismo ocurre con los otros ramos. Por ejemplo, si no soy matemático, es más probable que repruebe en comparación con un estudiante que ingresó a la carrera de Ingeniería Civil Matemática. Lo mismo ocurre con programación, es decir, si no soy informática, es más probable que repruebe el ramo en comparación con un estudiante informático. En base a esto, es el énfasis que se hace a la falta de motivación por parte de los estudiantes al momento de cursar este curso.

Al igual que este estudiante, los demás estudiantes del mismo curso, que pertenecen a la misma carrera, tampoco están lo suficientemente motivados con la realización del curso, por lo cual, nadie invita al estudiante a lograr la motivación necesaria para poder cursar el ramo sin la necesidad de que sea un compromiso.

En cuanto a la dinámica de las clases, los estudiantes interactúan directamente con el profesor, impidiendo una interacción activa entre los estudiantes del curso. En (Stefan, 2009), se define la interacción como *“participar y unirse a un diálogo para un aprendizaje activo y comprometido”*, es decir, la participación de los estudiantes es estar en constante diálogo entre pares o con el mismo profesor o ayudante, con tal de ser parte de la clase y no un espectador de la clase.

Cuando los estudiantes están des involucrados y sienten que no pertenecen a la institución, se retraen de las actividades de forma significativa. El alumnado involucrado cree que la experiencia educativa que vive es relevante para su futuro y tiene mayor disposición para el aprendizaje, no solo durante su estancia en la universidad, sino también a lo largo de todo el ciclo vital (Negrini, 2010).

Por lo tanto, para poder ayudar a los estudiantes con la participación, se buscan generar grupos de trabajo o parejas, con tal de que se mantengan en constante participación entre los miembros del curso. No solo que el profesor realice las fases teóricas del conocimiento, sino que también los estudiantes puedan interactuar entre ellos para poder mantenerse activos y practicar los elementos teóricos que se establecieron en la fase teórica anterior con el profesor. El trabajo en equipo facilita el cumplimiento de objetivos, incrementa la motivación y la creatividad, y favorece las habilidades sociales de cada uno (SENCE, 2021).

En esta memoria, se presenta el desarrollo de una plataforma que incorpora la metodología de programación de a pares. En esta plataforma se busca que los estudiantes puedan resolver ejercicios de programación en parejas, pero tendrán tiempo limitado de escritura. Es decir, un compañero escribe el código, mientras el otro compañero observa el código para encontrar incongruencias o errores de sintaxis. En esta plataforma se podrá programar desde dos computadores distintos, y los estudiantes estarán en contacto entre ellos.

Este documento se compone de 5 capítulos. En el primer capítulo se define el problema central, indicando la situación actual del curso, la información respecto al problema y los objetivos planteados.

En el siguiente capítulo, que es el marco conceptual, se describen los conceptos base para el desarrollo de la plataforma.

En el tercer capítulo, se desarrolla la propuesta de solución, en la cual se plantea como fue realizada la plataforma y las respectivas tecnologías utilizadas.

En el cuarto capítulo, se describen las pruebas realizadas por los estudiantes, donde se componen de la habilitación de un prototipo inicial, en donde se comprueba que la plataforma está bien diseñada y que los estudiantes pudieron realizar correctamente los ejercicios. Luego del prototipo inicial, se realiza la prueba principal, donde los estudiantes deben realizar un ejercicio de practica para el desarrollo del certamen del curso. Luego de que los estudiantes pudieran o no resolver el ejercicio, se realiza una encuesta para saber

qué opinión tienen los estudiantes con respecto a la plataforma y a la metodología ocupada para ella. El objetivo de estas pruebas es comprobar que la programación de a pares es una metodología útil para aprender a programar, además de que ayuda a que los estudiantes puedan interactuar entre ellos para compartir conocimientos.

Para finalizar, concluir el desarrollo de esta memoria en el último capítulo. En este se buscan los trabajos a futuro para la plataforma además de obtener respuestas en base a los resultados obtenidos.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Situación actual

Actualmente, en la Universidad Técnica Federico Santa María, se imparte el curso IWI-131, o mejor conocido por los estudiantes como Programación, el cual consiste en que los estudiantes desarrollen y apliquen el pensamiento computacional para resolver distintos problemas de ingeniería, desarrollando programas, principalmente en el lenguaje de programación *Python*. El diseño de algoritmos y programar soluciones utilizando el lenguaje de programación mencionado y distintas estructuras de datos es la competencia específica del ramo, con tal de poder resolver problemas de ingeniería utilizando estas herramientas. (Meza, 2022)

El curso pertenece a los cursos comunes de las ingenierías, es decir, que todas las carreras relacionadas con la ingeniería deben tomar este curso.

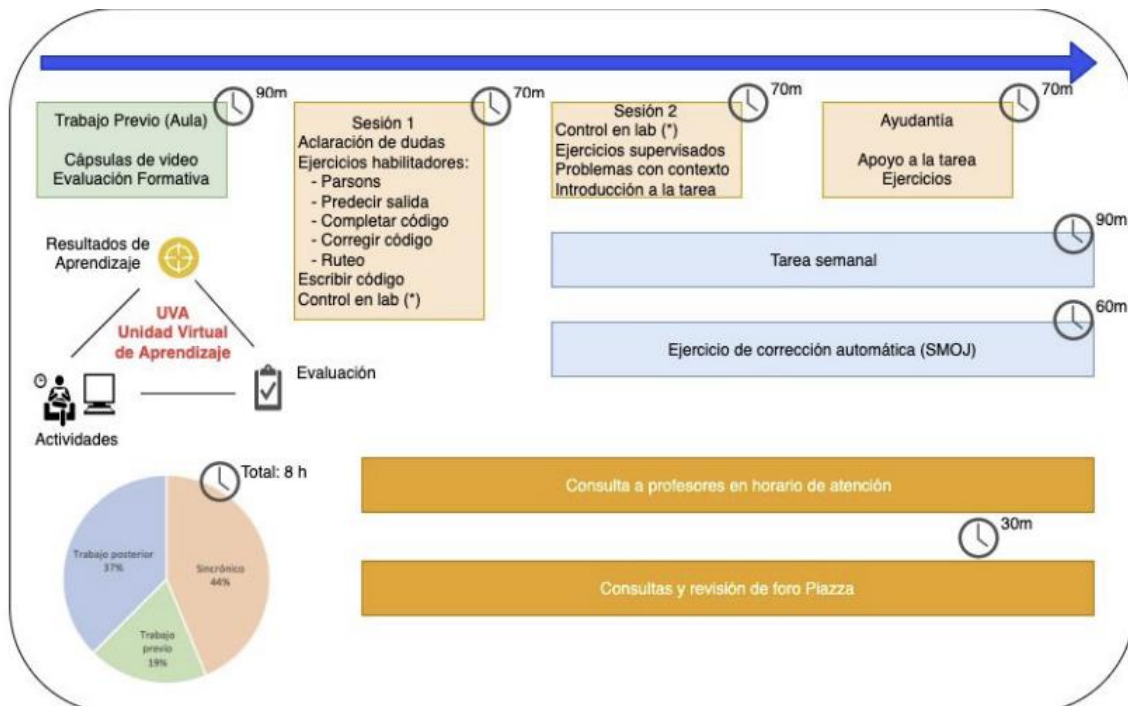


Figura 1: Metodología y modelo formativo del ramo IWI131.

Fuente: Programa de asignatura IWI131.

En la figura 1, se puede observar la metodología y el modelo formativo del curso. Este modelo formativo se basa en el concepto Unidad Virtual de Aprendizaje o UVA, el cual proporciona a que los estudiantes obtengan resultados de aprendizaje en función de actividades y evaluaciones.

Se comienza con un trabajo previo por parte de los estudiantes, donde ellos deben visualizar cápsulas de video o realizar evaluaciones formativas de forma virtual. Luego se continúa con la sesión N°1 del curso, que es la primera clase de la semana supervisada por un profesor o profesora en formato presencial, aclarando ciertas dudas por parte de los estudiantes y luego continuar con ejercicios habilitadores.

La sesión N°2 es también una sesión en contacto con el profesor, que comienza con un control de laboratorio que se hace de forma online, pero donde los estudiantes deben presentarse a la sala de clases, donde luego se hacen ejercicios supervisados. Los estudiantes deben lograr hacer estos ejercicios y aclarar ciertas dudas en relación con el ejercicio con el profesor asignado en el aula. Por último, las ayudantías, las cuales están a cargo de ayudantes, que fueron estudiantes del curso anteriormente y obtuvieron excelentes calificaciones, permitiéndoles ayudar a los estudiantes actuales del ramo en resolver dudas del curso, de certámenes, de tareas o de contenidos del curso. En estas ayudantías principalmente se hace un repaso general de lo visto en la semana para luego continuar con dudas o aclaraciones sobre la tarea que corresponde a la semana (Meza, Reglas del curso Programación, 2022).

Como elementos adicionales, los estudiantes deben realizar una tarea semanal, que corresponde a un ejercicio tipo certamen que deben resolver hasta una cierta fecha, con tal de incentivar las prácticas de ejercicios y ayudarlos a prepararse mejor para la evaluación del curso. En formato de práctica, también existe la plataforma Santa María Online Judge (SMOJ) que permite la corrección de ejercicios de manera automática (Leyton, 2022). Los estudiantes tienen a su disposición ejercicios de distintos niveles de dificultad que ellos pueden desarrollar para evaluar sus capacidades prácticas, permitiendo desarrollar soluciones para estos ejercicios y ponerse a prueba.

1.2 Información del problema

El curso de programación cuenta con múltiples etapas que ya fueron mencionadas anteriormente, donde cada una de ellas es un factor importante en el aprendizaje de los estudiantes. Cada una de estas etapas debe cumplir con los objetivos propuestos para que el aprendizaje por parte de los estudiantes tenga propósito.

Pero, en la sesión N°2 del curso, algunos de los profesores han tenido problemas en desarrollar la sesión en un cien por ciento, principalmente en la realización de los ejercicios supervisados. Esta es una sesión de laboratorio, es decir, los estudiantes se encuentran en una sala de clases que cuenta con un computador para cada uno de ellos, donde pueden realizar el o los ejercicios supervisados.

Según el profesor y jefe del curso, Federico Meza, *“los ejercicios supervisados no han tenido los resultados esperados, ya que muchos de los estudiantes que se encuentran en la sala de clases no se dedican a resolver el ejercicio sugerido”*. Muchos de estos estudiantes

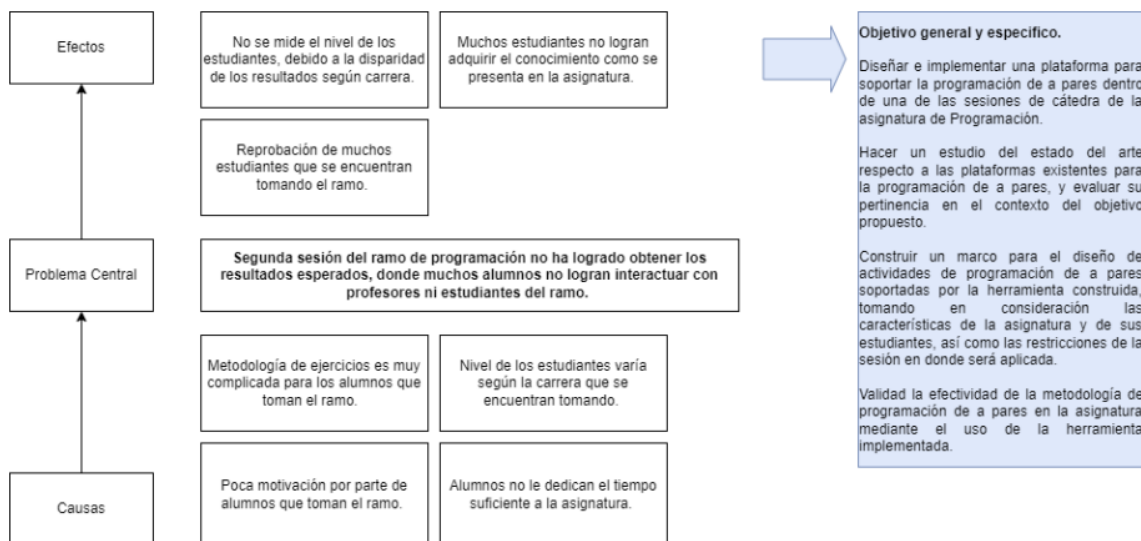
se dedican a jugar con los computadores, a hablar con sus compañeros o hacer cualquier cosa menos a practicar con el ejercicio que se sugirió en la clase. Esto impide que los estudiantes practiquen en el desarrollo del ejercicio, además de que impide que estos hagan consultas o planteen dudas al profesor en relación con el ejercicio o con la materia del curso.

Aparte de lo señalado, la realización del ejercicio es de forma individual, aunque no se les prohíbe trabajar con algún compañero para poder resolver el ejercicio junto, pero el formato es individual, donde cada uno de los estudiantes tiene un computador para realizar el ejercicio propuesto.

De acuerdo con lo señalado, esto puede tener bastantes consecuencias negativas, principalmente asociadas a la desmotivación en realizar el curso. Si un estudiante no se encuentra motivado con el curso, puede atraer consigo las distintas consecuencias negativas, como por ejemplo, reprobar el curso, desertar del curso o incluso, salirse de la universidad porque no ha podido lidiar con el curso.

La participación de los estudiantes también forma un papel fundamental al momento de realizar el curso. La interacción alumno-alumno en el curso es muy poco frecuente, ya que no existen sesiones que aumenten la participación entre ellos, sino solamente interactuar con el profesor, aclarando dudas. Esta también es otra fase del problema que aumenta la desmotivación por parte de los estudiantes del curso.

Como puede verse en el árbol del problema, las principales causas del problema son la desmotivación o poca motivación que tienen los alumnos con el curso, señalado directamente con el tiempo en que estos alumnos le dedican al curso. En el problema principal se señala que la segunda sesión del curso no ha logrado los resultados esperados, impidiendo que los alumnos tengan interacción ni con el profesor ni con sus compañeros. Los efectos que pueden ocurrir es que muchos estudiantes no logran adquirir el conocimiento necesario para aprobar el curso, incluso llegan a la reprobación del curso en muchos estudiantes que presentan desmotivación al momento de realizarlo.

**Figura 2:** Árbol del problema.

Fuente: Elaboración propia.

1.3 Propuesta

Para superar la problemática anteriormente propuesta, se propone el desarrollo de una plataforma paralela a la plataforma SMOJ que permita que los estudiantes interactúen entre ellos al hacer ejercicios de práctica.

Esta plataforma permite realizar la metodología de programación de a pares (Pair Programming), en donde dos estudiantes participan en la realización de un ejercicio en donde uno de ellos escribe el código mientras el compañero revisa el código del que escribe para, luego de un determinado tiempo, cambiar las funciones de ambos estudiantes, es decir, ahora el estudiante que escribe es el que antes revisaba el código, mientras que el estudiante que revisa es el que antes escribía. Eso se hace por un determinado tiempo de desarrollo del ejercicio y permite que los estudiantes interactúen entre ellos.

La utilización de esta plataforma sería en las sesiones N°2 de cátedra, para permitir que los estudiantes hagan los ejercicios supervisados y compartan conocimientos. Además, para situaciones de práctica de ejercicios, también estaría habilitada para ayudantías, donde los estudiantes que asisten a estas ayudantías pueden hacer los ejercicios con un compañero.

En cuanto al impacto de esta implementación, tal como se mencionó anteriormente, permite la participación de los estudiantes, no tan solo con el profesor encargado, sino que también con sus propios compañeros de curso. Además, permite alentar la motivación de los estudiantes, gracias a que estos no se sentirán solos al practicar con

ejercicios. Esta participación hará que los estudiantes puedan hacer más consultas o dudas acerca del código al profesor, para verificar en que se equivocaron o si tienen alguna duda con el ejercicio o materia en cuestión. Por otra parte, los profesores encargados de las sesiones de cátedra podrán establecer ejercicios supervisados y revisar menos códigos, ya que los ejercicios se realizarán en parejas, por lo cual, aumenta la atención hacia un código específico en vez de andar revisando muchos códigos.

1.4 Objetivos

1.4.1 Objetivo General

Diseñar e implementar una plataforma para soportar la programación de a pares dentro de una de las sesiones de cátedra de la asignatura de Programación.

1.4.2 Objetivos Específicos

- Hacer un estudio del estado del arte respecto a plataformas existentes para la programación de a pares, y evaluar su pertinencia en el contexto del objetivo propuesto.
- Construir un marco para el diseño de actividades de programación de a pares soportadas por la herramienta construida, tomando en consideración las características de la asignatura IWI-131 de la UTFSM y de sus estudiantes, así como las restricciones de la sesión en donde será aplicada.
- Validar la efectividad de la metodología de programación de a pares en la asignatura IWI-131 de la UTFSM, mediante el uso de la herramienta implementada.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Programación

En el desarrollo de software y en la ingeniería, programar pasa a ser un concepto básico y fundamental para el desarrollo de futuros ingenieros, ya que el concepto de programar busca organizar conjuntos de datos, principalmente para organizar e idear lo necesario para el futuro desarrollo de un proyecto, que puede ser informático o de otra carrera relacionada con la ingeniería.

En la informática, este concepto busca la relación entre los computadores y los usuarios que utilizan estos. A través de los lenguajes de programación, podemos hacer que nuestro computador desarrolle y ejecute las aplicaciones, programas o actividades que nosotros como usuarios queramos. (Executrain, 2022)

2.1.1 Lenguaje de Programación

Los lenguajes de programación son el idioma artificial que está prediseñado con signos, palabras o distintos símbolos que nos permiten ejecutar los programas que se realicen. Se permite que los programadores, tengan la capacidad de escribir una serie de instrucciones formando un algoritmo, controlando todos los comportamientos de un sistema informático. Con la utilización de distintas estructuras de datos, se pueden realizar y guardar datos en memoria, así se pueden utilizar estos datos y transformarlos en cosas útiles que servirán para el desarrollo de la aplicación. (Mathieu, 2014)

2.1.2 Python

Python es un lenguaje de programación diseñado principalmente para “propósitos generales”, es decir, que este lenguaje se pueda usar para resolver una variedad enorme de problemas. No solamente es un lenguaje que esté diseñado para informáticos, sino que permite su uso para otras carreras o áreas. Por ejemplo, se puede utilizar este lenguaje para generar gráficos que permitan nuevas visualizaciones de los datos, lo cual sería muy útil para ingenieros comerciales o analistas. También podremos diseñar programas para uso diario, sea cual sea el área, nos permite de todo.

```
print("Hello and welcome.")
name = input("Whats your name?")
if name == "Martin":
    print("Thats my name too!")
print("Hi " + name)
```

Figura 3: Ejemplo de Python.

Fuente: Introducción a la programación.

La figura nos permite visualizar la simbología del lenguaje, haciendo alusión a que es otro idioma nuevo, y el cual tiene distintos símbolos que cumplen con sus respectivas funciones. (Executrain, 2022)

En la Universidad Técnica Federico Santa María, se utiliza este lenguaje para las clases de programación, con tal de que todos los estudiantes que ingresen a ingeniería sepan los conceptos básicos de este lenguaje, y sean capaces de diseñar soluciones para problemas de ingeniería utilizando esta herramienta.

2.2 Programación de a pares.

La programación de a pares o *Pair Programming* es una metodología de desarrollo de software en el cual siempre existen dos o más personas que se encuentran trabajando en el mismo código y al mismo tiempo. Principalmente, una persona se encarga de escribir el código mientras que la otra persona se encuentra supervisando el trabajo de la persona que escribe y su forma de programar la solución del problema planteado. Pasado un lapso, las posiciones de los programadores cambian, es decir, la persona que se encontraba escribiendo ahora supervisa, mientras que la persona que anteriormente supervisaba ahora escribe la solución del problema planteado siguiendo las riendas que dejó la persona que escribía anteriormente. (IONOS, 2022)

Esta estrategia se ha estudiado a grandes rangos a lo largo del tiempo, arrojando resultados positivos y negativos, pero siempre ha sido una metodología que tiene una mirada en la enseñanza de los alumnos los cuales la practican, especialmente en alumnos que comienzan a programar y no entienden cómo funcionan los lenguajes de programación.

Dentro de las ventajas que tiene esta metodología al momento de implementarla en alumnos que nunca han programado y necesitan aprender es que permite la formulación y resolución de dudas entre los mismos alumnos que se encuentran aprendiendo, permitiendo la ayuda mutua entre ellos y, en casos donde los conceptos o dudas son muy complejas, el profesor puede profundizar en ellas. Esto permite minimizar riesgos a la hora de realizar el código, evitando errores de compilación. Además, permite que los alumnos puedan desarrollar su creatividad en el desarrollo del código, diseñando nuevas formas de escribir, permitiendo que la cantidad de líneas de código se minimice en cada problema solucionado.

Por el contrario, dentro de las desventajas de esta metodología se puede encontrar los tiempos de desarrollo de los ejercicios por parte de los alumnos, ya que muchos de estos pueden tener una visión de cómo hacer el código de forma personal, pero cuando deben agregar a otra persona para su colaboración, suelen ocurrir confusiones entre ellos, aumentando el tiempo de desarrollo haciendo que ambos alumnos deben conversar entre ellos para poder solucionar sus confusiones.

Otra desventaja es la formación de los distintos equipos de trabajo, ya que existen muchos alumnos que ya han programado anteriormente, mientras que otros alumnos nunca han programado en su vida, por lo cual, existe una disparidad en los conocimientos de los alumnos. Por lo tanto, debe existir una paridad a la hora de diseñar los equipos de trabajo, con tal de que los estudiantes que sepan menos sean emparejados con estudiantes que conozcan del ramo, y así permitir que los estudiantes que no tengan el conocimiento adecuado puedan sentirse ayudados por parte de los estudiantes que tengan conocimientos en la materia.

2.2.1 Estudios de otros autores.

Algunos autores y profesores del área han plasmado varios estudios sobre el concepto de programación de a pares, ya que implica una metodología muy innovadora en el área de la informática, y no es tan solo una metodología de aprendizaje, sino que también comparte tiene una gran relación con los trabajos relacionados en el área, donde dos o más programadores deben trabajar en conjunto para poder cumplir con los objetivos que tiene la empresa.

La utilización de la programación de a pares de forma distribuida, es decir, que se van cambiando los roles de los estudiantes a medida que va pasando el tiempo, tiene una tasa de aprobación bastante más amplia en estudiantes que tienen una pareja y se encuentran programando con esta metodología por sobre los estudiantes que no tienen pareja, es decir, se encuentran programando de forma individual. Además, un tema que se debe considerar es que se plantea que emparejar a los estudiantes con habilidades similares en el área de la programación logra resultados positivos y motivadores para los estudiantes, ya que participan más y realizan los ejercicios de forma muy motivada, apoyándose entre ellos. (Despina Tsompanoudi, 2016)

En temáticas relacionadas con el metaanálisis de datos, también ha sido una metodología de enseñanza con resultados positivos, pero va completamente de la mano con la correcta implementación de ella. Principalmente, al hablar de una correcta implementación de la metodología significa que, para poder implementarla, debe existir un moderador, es decir, una persona o programa que mida los tiempos de programación y supervise el trabajo de los estudiantes, con tal de que cumplan con los tiempos y respeten las reglas de la metodología. (KARTHIKEYAN UMAPATHY, 2017)

Ahora, en la actualidad, el desarrollo de software es, sin duda, la principal labor de un ingeniero informático o alguien que se especializa en el área. Esto se debe a que muchas empresas o instituciones, ya sean públicas o privadas, necesitan de un software especializado para la empresa y sus funciones, estableciendo un orden de las labores por parte de los trabajadores o ayudando a la empresa para poder darse a conocer a niveles más amplios. La programación de a pares en temáticas con desarrollo de software permite generar productos de software con una altísima calidad, ya que ambos programadores que se encuentran trabajando lo hacen de forma conjunta y dentro de una misma plataforma, donde les permite compartirse ideas entre ellos y abordando las distintas complejidades que puede tener el desarrollar el software en cuestión. Los proyectos que

se desarrollaron utilizando esta metodología tenían una enorme variedad en cuanto a los errores de compilación, generando productos de gran calidad y que funcionan extremadamente bien en las funcionalidades que necesita la empresa. En ciertas ocasiones, y por temas de costumbre o por llevarse de las distintas regalías de forma individual, muchos programadores prefieren trabajar de forma individual, donde no se sienten cómodos trabajando con otra persona en un mismo proyecto. (Kessler, 2000)

El último autor que se analizará acerca de la metodología de programación de a pares es Lewis, que hizo un estudio donde comparo firmemente las dos propuestas de metodología, trabajar con programación de a pares o de forma individual. Según lo estipulado, en los cuestionarios diarios que se hacían sobre conocimientos de programación, los estudiantes que trabajaban de forma individual obtenían más preguntas correctas en comparación con los estudiantes que tenían una pareja asociada, además de que eran mucho más rápidos para realizar estos cuestionarios. Sin embargo, los estudiantes que implementaban la programación de a pares en proyectos obtuvieron grandes resultados en alternativas que consideraban la colaboración, donde permitía que muchos estudiantes se pudieran brindar ayudas entre ellos, así poder realizar el proyecto de forma grupal y obtener mejores resultados. Por contraparte, dentro de los estudiantes que implementaron la programación de a pares, muchos de ellos sentían la frustración de trabajar en equipos, ya que muchos alumnos no cuentan con las mismas capacidades sociales y prefieren trabajar de forma individual. (Lewis, 2011)

2.3 Plataformas de programación de a pares

2.3.1 Motepair by Atom

Motepair es un paquete de editor de la plataforma Atom, la cual es una plataforma que permite programar en distintos lenguajes de programación y compilar los códigos en ella. Este paquete permite ejecutar sesiones de programación de forma remota entre dos computadores utilizando el editor de código de GitHub.

Con este paquete se pueden sincronizar dos o más editores de código Atom, donde se pueden hacer acciones como abrir o cerrar archivos, mover el ratón y obtener respuesta de los otros programadores que se encuentren en otros editores. En síntesis, es una plataforma que nos permite compartir el editor de código con otro programador, donde la otra persona puede utilizar todas las funcionalidades del editor de código, mientras el otro solo observa lo que hace el programador principal, es decir, el que maneja las riendas del proyecto. (Make it in Ukraine, 2021)

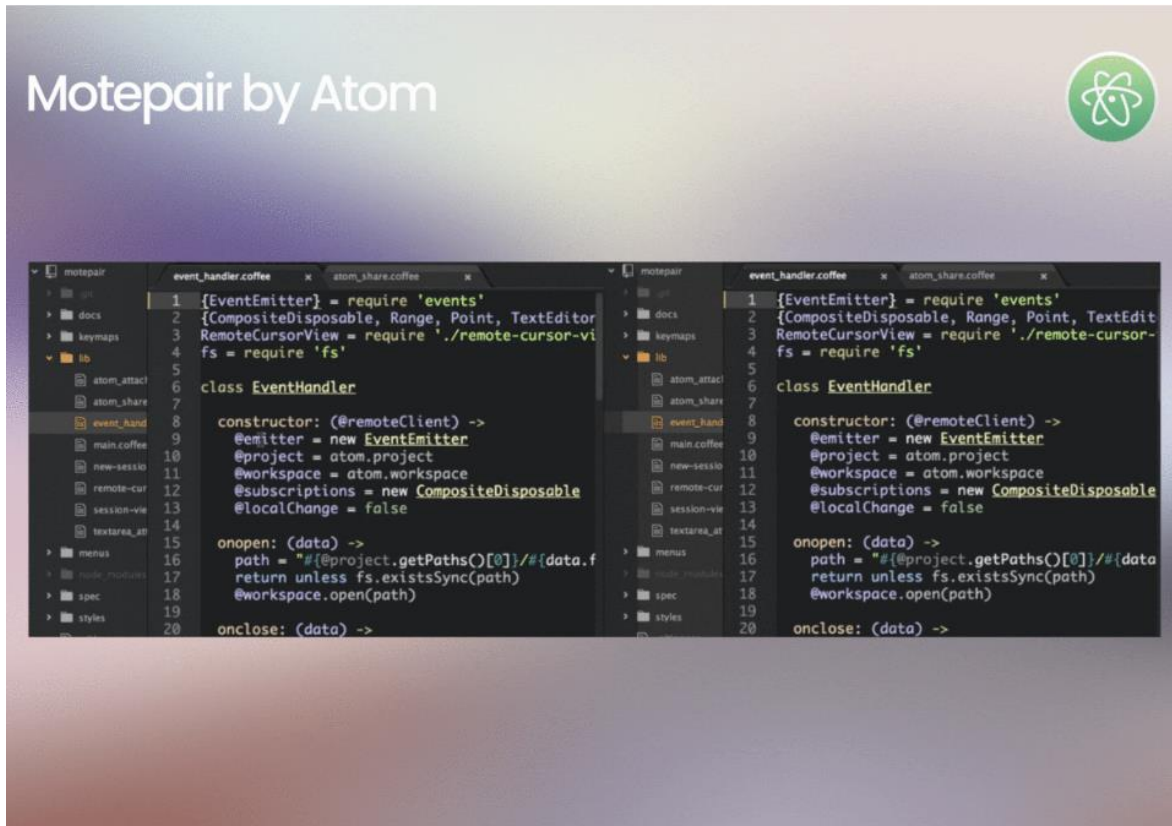


Figura 4: Motepair by Atom.

Fuente: Atom.

2.3.2 Microsoft Visual Studio Live Share

Microsoft junto con su herramienta de edición de código llamada Visual Studio Code diseñaron Live Share, que es un paquete de edición hecho exclusivamente para programadores que quieran trabajar juntos en una misma sesión. Uno de los programadores debe iniciar una sesión compartida en Visual Studio Code y, a través de un enlace, sus compañeros de equipo obtienen acceso instantáneo y seguro a las herramientas de código del programador principal. No necesitan clonar, ni copiar o hacer distintas configuraciones, solamente van a programar.

Live Share permite coeditar, depurar, tener llamadas de audio, chatear, compartir las terminales, servidores y hacer comentarios para que el resto del equipo de trabajo sepan cuáles son los pasos para seguir y poder finalizar el proyecto de forma exitosa.

Principalmente, Live Share está diseñada para equipos de trabajos formados y pertenecientes a una empresa o institución que realizan proyectos grandes. Tiene muchas personalizaciones de su menú principal, lo que hace que el trabajo a realizar sea mucho más fácil y rápido para todos los programadores que forman parte del proyecto. (Microsoft, 2022)

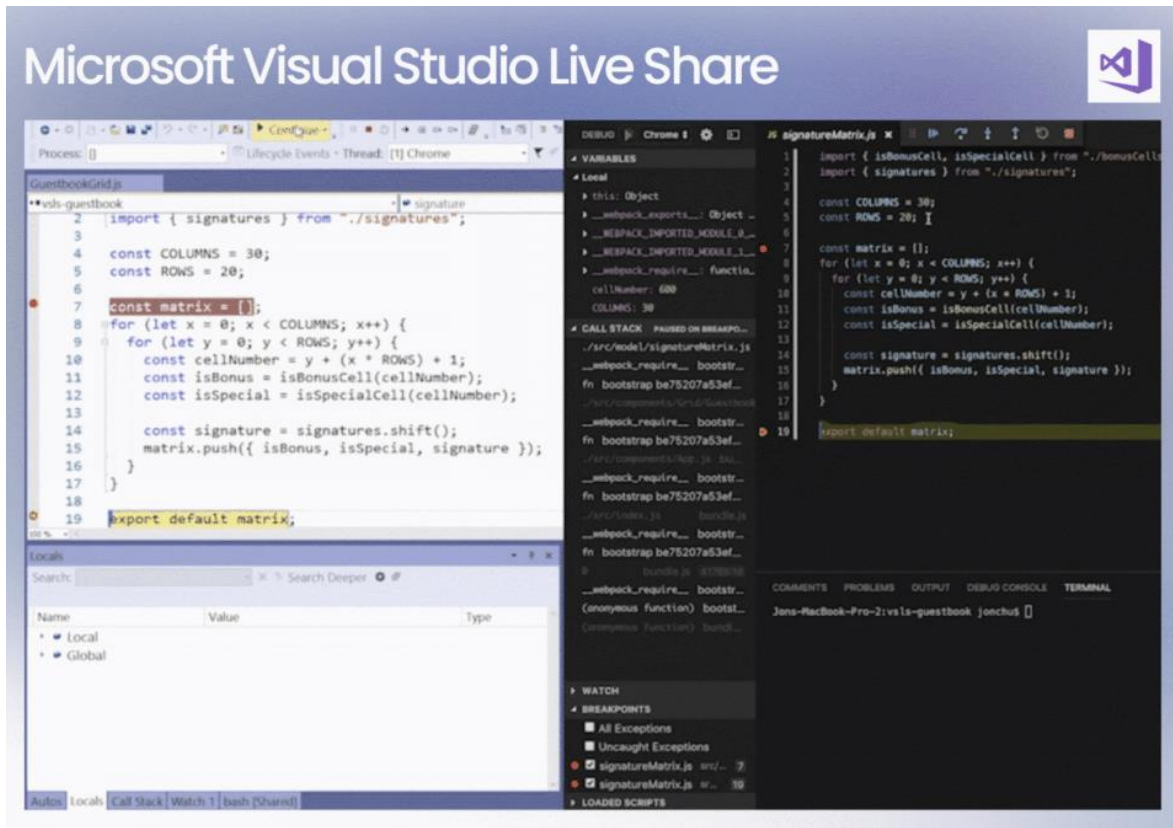


Figura 5: Visual Studio Live Share.

Fuente: Microsoft.

2.3.3 Codeanywhere

Codeanywhere es un entorno de desarrollo integrado (IDE) creado por Codeanywhere, Inc. Su principal objetivo es que los usuarios escriban, editen, colaboren y ejecuten proyectos de programación desde la web o desde un dispositivo móvil.

Relacionado con la programación de a pares, esta plataforma permite a los desarrolladores de código trabajar y colaborar entre ellos en tiempo real. Estos pueden compartir sus archivos y código desde cualquier dispositivo mientras están programando en él. Al igual que otras plataformas como GitHub o Live Share, permite hacer un seguimiento de las revisiones y de la programación de a pares en vivo con desarrolladores ilimitados, permitiendo acceder a cada parte de los desarrolladores y seguirlos solo presionando el icono y verificar su posición actual. (Codeanywhere, 2022)

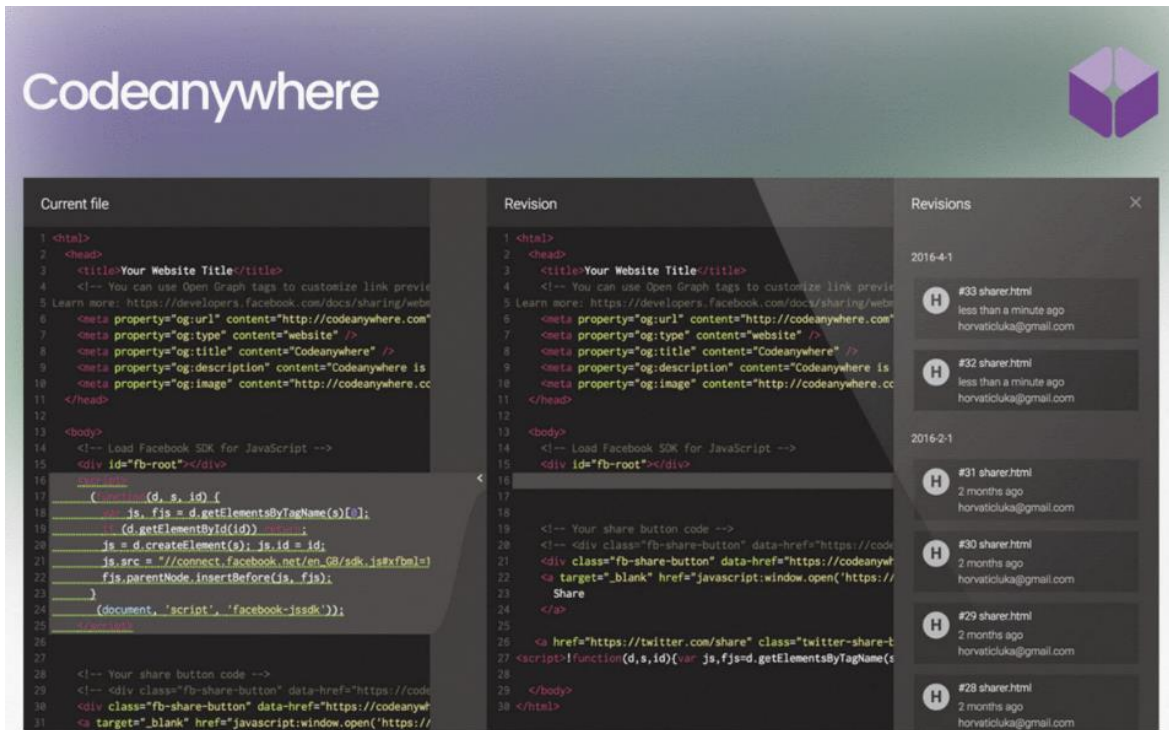


Figura 6: Codeanywhere.

Fuente: Codeanywhere.

2.3.4 CodeTogether

CodeTogether es una plataforma que permite compartir en vivo su IDE con la finalidad de que los desarrolladores puedan trabajar colaborativamente utilizando la programación de a pares en tiempo real. Es la mejor herramienta de programación de a pares y la que más se adecua a las problemáticas que enfrentan los profesores de la universidad. Permite tener múltiples desarrolladores trabajando conjuntamente, haciendo revisión de código y permite la opción de tener pantallas compartidas para los colaboradores. (Aboyi, 2021)

Esta plataforma permite compartir en vivo desde varios editores de código, como por ejemplo Eclipse, IntelliJ o VS Code. Los invitados pueden compartir desde su IDE local incluso si es diferente. Todos estos desarrolladores se pueden unir a las sesiones a través de un enlace que comparte la sesión.

Esta es la plataforma que más se adecua al problema mencionado debido a que tiene opciones de host muy avanzadas, es decir, una persona crea una sesión y puede tener los varios invitados trabajando en ella. Pero, cuando quiere que otros invitados no puedan seguir escribiendo en el código, puede pausarlos y solamente dejarlos como visualizadores de la sesión, sin la posibilidad de escribir código dentro. Esto nos permite crear las distintas sesiones de programación de a pares y poder medirlas con un rango de tiempo, y así poder parar a algunos estudiantes de la función de escribir código y que solo visualicen como avanza su compañero de equipo. (CodeTogether, 2022)

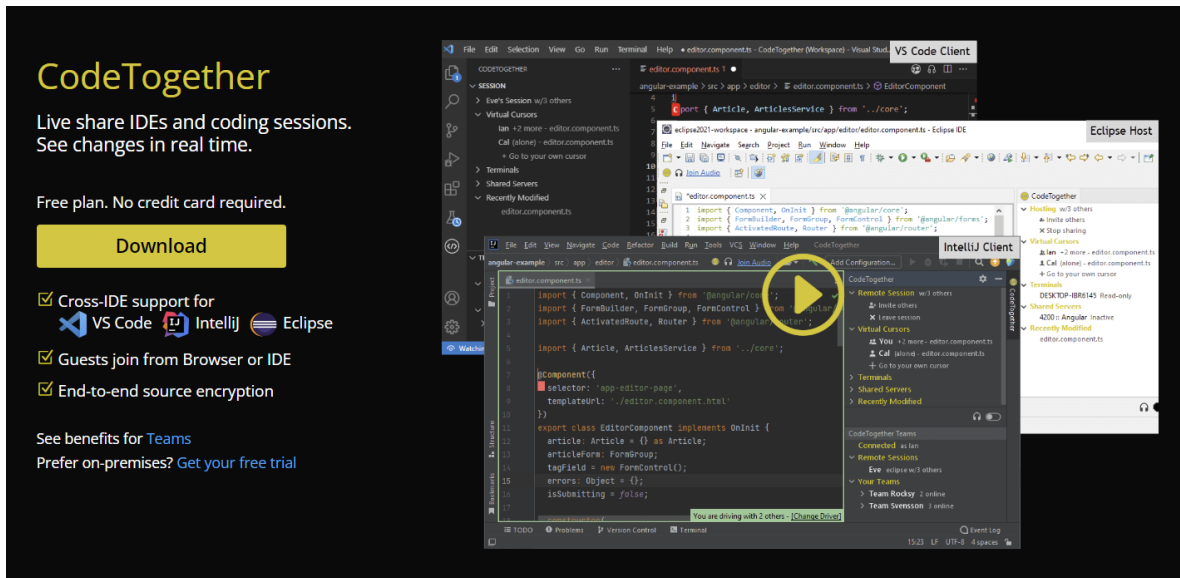


Figura 7: CodeTogether.

Fuente: CodeTogether.

2.4 SMOJ

SMOJ o Santa María Online Judge es una plataforma orientada en la resolución distintos problemas algorítmicos y completamente relacionada con la asignatura IWI-131 del Departamento de Informática de la Universidad Técnica Federico Santa María.

Esta plataforma permite resolver problemas del ramo que pueden ser tanto ejercicios propuestos como tipo certamen. Al momento de resolverlos, estos se comprobarán y medirán la solución planteada por el estudiante, comparándolos con varios casos de prueba.

La plataforma fue completamente diseñada por estudiantes de la universidad, y es una apuesta a seguir para el desarrollo de la plataforma de programación de a pares, la cual tendrá la misma base de datos de problemas sugeridos y la automatización de la corrección de código, con la finalidad de entregar el resultado a los estudiantes de forma inmediata y que puedan saber cuáles fueron los errores cometidos al momento de realizar la codificación del problema. (Hugo Leyton, 2017-2022)



Figura 8: Santa María Online Judge.

Fuente: SMOJ.

CAPÍTULO 3: PROPUESTA DE SOLUCION

Tal como se mencionó anteriormente, SMOJ es una plataforma de la UTFSM para los estudiantes del curso de programación. En esta plataforma, los estudiantes pueden practicar con ejercicios y estos serán juzgados de forma automática con la plataforma. Pero, esta plataforma no permite la participación entre estudiantes del curso, sino que incentiva a que los estudiantes practiquen ejercicios prácticos en sus casas y les entregue un resultado de forma automática, para saber si realizaron bien el ejercicio o tuvieron ciertos errores, ya sea en sintaxis o en el algoritmo implementado.

Es por lo anterior, que el objetivo de esta solución es diseñar una nueva plataforma para la universidad que permita que los estudiantes cumplan con condiciones muy similares a SMOJ, pero puedan trabajar en conjunto, principalmente siguiendo la metodología de programación de a pares.

En primeras conversaciones con el jefe del curso, Federico Meza, primero se revisó la posibilidad de integrar esta plataforma en SMOJ, ya que tendrían funciones muy similares y los problemas serían compartidos entre plataformas. Luego de conversación con uno de los estudiantes encargados de la plataforma SMOJ, Hugo Leyton, quedamos en la conclusión de que debía hacerse una plataforma paralela a SMOJ, principalmente por la utilización de tecnologías de esta plataforma.

Para efectos de este documento, describiremos cada etapa del proyecto por separado, es decir, primero verificaremos los datos, luego las funcionalidades dentro de la plataforma y luego el diseño gráfico de presentación. Para cada una de estas etapas, les asignaremos nombres para poder pasar por cada una de ellas y su correspondiente desarrollo:

1. Desarrollo de Backend: Verificación de los datos que se almacenan en la plataforma, con la respectiva tecnología utilizada para ello.
2. Desarrollo de plataforma: Funcionalidades del sistema, es decir, como se conectarán las etapas correspondientes.
3. Desarrollo del diseño: Interfaz gráfica de la plataforma, explicando las tecnologías ocupadas.

3.1 Desarrollo del Backend

Para el desarrollo del Backend, se trabajó con el framework Django, el cual está basado en el lenguaje de programación Python. Se utilizó este framework, ya que permite el desarrollo rápido de la aplicación reutilizando código que tiene como base este framework. Además, permite la arquitectura de Modelo-Vista-Controlador (MVC), donde

permite separar la organización del código, tal como se explica en este documento. Otro factor importante para señalar de este framework es la capacidad de conexión con distintas bases de datos, tales como PostgreSQL, MySQL, etc.

En función de que esta plataforma aún no se encuentra en producción, no se trabajó con ninguna base de datos, sino que simplemente se trabajó con el ORM de Django, es decir, que se interactúa directamente con la base de datos usando Python, creando una API donde se almacenan los datos necesarios.

Las tablas que se crearon son las siguientes:

- Usuario: contiene la información acerca del usuario, principalmente almacenando el nombre, el email, el paralelo del curso al cual pertenece y si es administrador de la plataforma o no lo es.
- Parejas: contiene los identificadores de los dos usuarios que conformaran parejas en la plataforma.
- Ejercicios: información correspondiente al ejercicio, donde se obtiene el nombre del ejercicio y el identificador del ejercicio en la base de datos de ejercicios del ramo de programación.
- Talleres: se crean con base en los ejercicios que se quieren en cada taller, donde contiene el nombre del taller, los ejercicios que conformaran parte del taller, la fecha de inicio y de término del taller y el tiempo de programación.
- Respuestas: son las respuestas de los estudiantes al hacer el ejercicio. Se almacenan los dos usuarios que realizaron el ejercicio, el código que ellos escribieron, el ejercicio del cual hicieron la solución, la fecha en que enviaron la respuesta, si está revisado o no, el puntaje obtenido y los resultados del juez online.

3.1.1 Modelado de datos

A continuación, se presenta un diagrama con las relaciones entre las tablas mencionadas anteriormente:

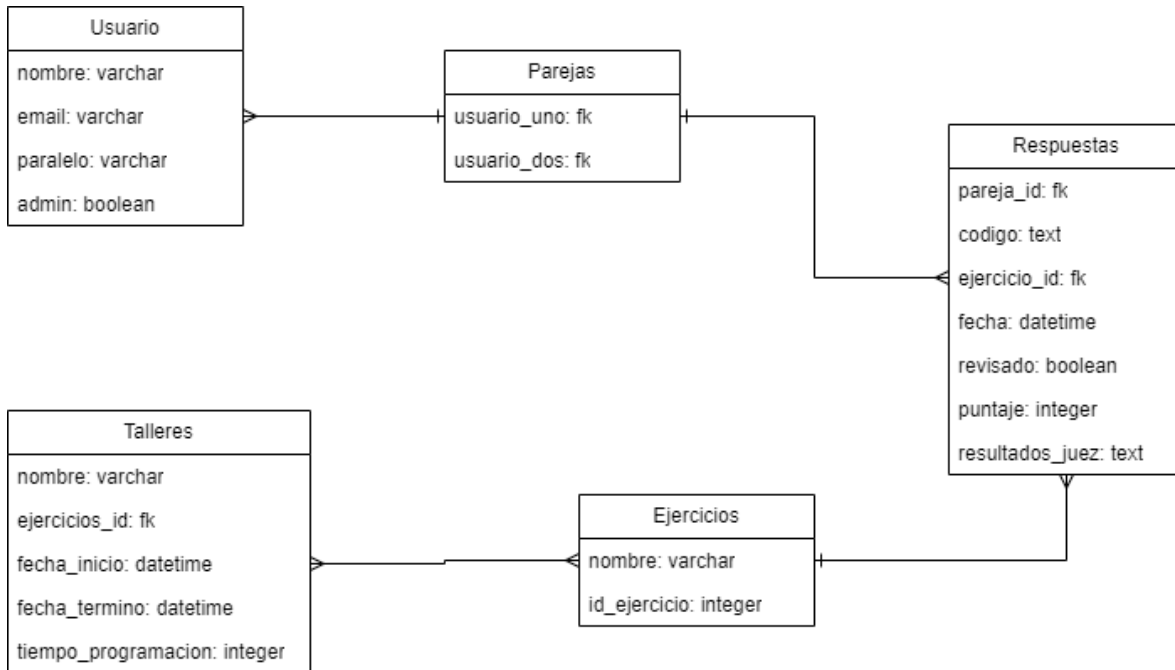


Figura 9: Modelado de datos. Fuente: Elaboración propia

3.1.2 Autenticación

Para poder ingresar a la plataforma, existe solo un método de autenticación, y es a través de la plataforma AULA de la UTFSM. Es decir, si no eres estudiante de la universidad, no puedes acceder a la plataforma.

Esta metodología de autenticación se logró gracias a LTI (Learning Tools Interoperability). Gracias al uso de herramientas externas de la plataforma AULA, que permiten a los estudiantes acceder a otros recursos de otros sitios web, se logró una autenticación completa con el uso de la plataforma.

En un paso a paso para la autenticación, un estudiante debe ingresar a la plataforma AULA a través de su cuenta institucional, luego ingresar al curso de programación inscrito, luego ingresar al enlace de la plataforma, el cual se verá en la siguiente imagen. Luego de presionar ese enlace, el estudiante accede a la plataforma ya autenticado, donde la plataforma, gracias a las herramientas LTI, permite sacar información relevante del estudiante, como el nombre y el email institucional.

Como pasos adicionales, en la plataforma debe ingresar su paralelo solo una única vez, ya que la herramienta LTI no entrega ese tipo de información con respecto al curso. Al momento de ingresar el paralelo, el usuario queda como registrado en la plataforma y puede acceder a ella.

3.1.3 Roles y permisos

En la tabla usuarios, existe un campo llamado admin, el cual es un booleano que permite saber si el usuario que está en la plataforma es un administrador o no lo es, ¿cualquier usuario puede ser administrador? Obviamente no, solo los usuarios que se ingresen con el booleano en verdadero podrán ser administradores.

Para ingresar un usuario como administrador, este no inscribirse a través de AULA, sino que se debe inscribir de forma manual a través de la API de la plataforma, con tal de ingresar el campo admin como verdadero.

Los usuarios admin tienen permisos adicionales que los estudiantes no tienen. Primero que todo, estos pueden verificar los paralelos, ver todos los estudiantes inscritos de cada paralelo y observar también las parejas. Además, permite modificar las parejas en caso de que algún estudiante quiera cambiar su pareja por distintos motivos.

También tiene disponible la pestaña de talleres, en donde puede ingresar y crear nuevos talleres con los ejercicios que quiera para el taller indicado. Por ejemplo, si quiere un taller donde se pase materia como Listas, entonces puede buscar ejercicios de listas y agregarlos al taller, ingresando los datos que pide la plataforma.

Los roles comunes, es decir, que no tienen acceso al admin, solo pueden ingresar a la pestaña de talleres en donde pueden visualizar cuáles son los talleres que se encuentran abiertos y con ejercicios propuestos. Luego de ingresar al taller, eligen el ejercicio que quieran realizar, se juntan con su pareja y pueden trabajar en el código desde dos computadores distintos. Esto se explicará más adelante en detalle.

3.2 Desarrollo de plataforma

Dentro de las tecnologías ocupadas para el desarrollo de la plataforma está Django con el lenguaje de programación Python para el backend, React con el lenguaje de programación JavaScript para el frontend y algunas herramientas adicionales como el desarrollo de sockets.

Primero, se describirán los frameworks y herramientas utilizadas para el desarrollo de la plataforma:

- Django: framework de desarrollo para Python diseñado para la creación de páginas web. Principalmente, se utiliza para el desarrollo de servidores (Tokio School, 2022).
- React: librería de JavaScript para el desarrollo de aplicaciones web. Tiene fragmentos de código de JavaScript reutilizables para crear interfaces de usuario (Deyimar, 2023).

- Sockets: canales de comunicación entre procesos para intercambio de datos localmente y entre redes. Basado principalmente en la arquitectura de cliente-servidor (IBM, 2023).

En cuanto a las entidades que interactúan con la plataforma, tenemos a los estudiantes, quienes son el centro principal y quienes realizan más actividades con la plataforma, los profesores que se encargan de la administración de la plataforma y de los ejercicios, y por último el sistema, que debe interactuar por sí solo para calcular los puntajes obtenidos por los estudiantes y el correcto uso de la metodología de programación de a pares. Se describirá a cada entidad con sus respectivas acciones a continuación:

3.2.1 Estudiantes

Autenticación

Tal como se mencionó anteriormente, el usuario puede acceder a la plataforma mediante AULA, mientras sea un estudiante matriculado en la UTFSM.

Para la metodología de autenticación, se utiliza LTI que permite conectar y reproducir una amplia variedad de herramientas externas, contenidos y actividades disponibles. Esto permite un intercambio seguro de información entre la aplicación y el LTI, que corresponde a la plataforma AULA (Verdaguer, 2021).

Para poder implementar esta autenticación, primero se debe crear la herramienta externa en la plataforma AULA, la cual pide distintos parámetros para la respectiva conexión con la plataforma, tales como el nombre de la plataforma, el endpoint, la clave pública y la clave privada.

Con eso asignado, el estudiante solo debe ingresar a la plataforma AULA, y luego ir al enlace de la plataforma.

A través de la conexión LTI, la plataforma puede obtener información del estudiante que ingreso, como el nombre completo y el email. Con esa información dentro de la plataforma, solo falta un dato específico que debe ser ingresado por el estudiante, que es el paralelo del curso. Ingresado eso, ya está autenticado en la plataforma.

Ver talleres abiertos

Al estar autenticado, el estudiante puede ingresar a los talleres que se encuentran abiertos para la realización de los ejercicios propuestos.

Todos los talleres creados están almacenados en la API de Django, por lo cual, se comprueba que el taller este dentro de la fecha de inicio y la fecha de término para que el

estudiante pueda realizar dicho taller y observar los ejercicios. De lo contrario, el taller se considera no disponible.

Ver ejercicios de un taller

Si un taller se encuentra disponible, los estudiantes que accedan al taller podrán ver los ejercicios propuestos para ese taller. Estos ejercicios son previamente asignados por el profesor que creo el taller.

Para obtener el ejercicio correspondiente, este está almacenado en la API de Django, pero este ejercicio tiene un identificador. Este identificador es el correspondiente a la base de datos de problemas de programación que está actualmente funcionando para la plataforma SMOJ. Entonces, al tener el identificador del ejercicio, podemos buscarlo en la base de datos de problemas de programación y obtener el problema. Se puede ver en el siguiente diagrama la forma en cómo se accede al ejercicio:

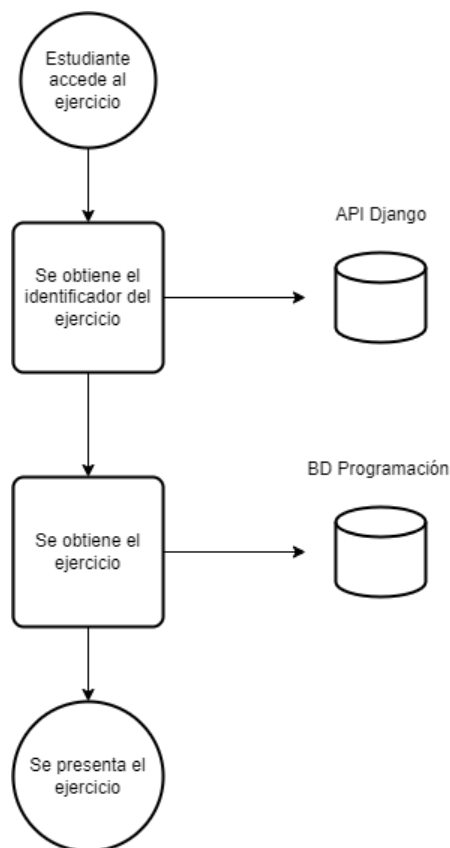


Figura 10: Obtención del ejercicio. Fuente: Elaboración Propia

Escribir código

Al ingresar a un ejercicio disponible para el taller, se observa el problema planteado, junto con distintos ejemplos de entrada y de salida del programa. Luego se encuentra el IDE, el

cual se programó con AceEditor, el cual es un editor de texto en tiempo real que se puede configurar a través de JavaScript.

El estudiante podrá escribir su código en esta sección, implementando sus conocimientos para resolver el ejercicio.

Probar código

Cuando un estudiante ya escribió su código del programa y quiere probar su código para verificar sus resultados, la plataforma permite probar el código en tiempo real.

Para la implementación de probar el código se trabajó con la API Judge0, la cual recibe el código del estudiante y lo evalúa de acuerdo con los casos de prueba. Primero se entregan datos de entrada y se comparan con los datos de salida, si los datos de salida corresponden con los datos juzgados, entonces el resultado será correcto. De lo contrario, la respuesta sale como incorrecta. Si existen errores en el código, también se señala como respuesta incorrecta y entrega el error correspondiente junto con la línea en donde ocurre el error para que el estudiante pueda corregir el error.

La implementación de la API Judge0 se logra a través de Docker, implementando un servidor exclusivo para el funcionamiento de esta. Cuando el estudiante escribe el código y le da a probar, entonces se llama directamente a la API para que juzgue el código, donde se envía el código que escribió el estudiante y un solo caso de prueba, en donde la API emitirá su respuesta en base al caso de prueba. Cuando ya obtiene el resultado, lo envía a la plataforma y muestra el resultado al estudiante.

Enviar código

Como fase final con respecto al desarrollo del ejercicio, el estudiante envía su código para que sea evaluado por la API Judge0. Primero, el estudiante al botón correspondiente de enviar el código listo, luego se guarda la respuesta en la API de Django, principalmente en la tabla de respuestas, almacenando el código escrito, la fecha, los dos usuarios que escribieron el código, el ejercicio y variables correspondientes a la revisión.

Al momento de enviar el código y ya almacenado, entonces se procede a juzgar el código, ya no con un solo caso de prueba, sino con todos los casos de prueba que tiene la base de datos de programación en relación con el ejercicio resuelto. Se comprueba la variable revisado, la cual corresponde a saber si el código fue revisado o no por la API Judge0. Si no ha sido revisado, entonces se envía el código del estudiante a la API Judge0 con todos los casos de prueba del ejercicio.

Cuando la API Judge0 ya reviso todos los casos de prueba, entonces almacena sus resultados en la variable resultados_judge0, en esta se puede ver cada caso de prueba con su resultado, si está aceptada, si está incorrecta, si tiene errores de compilación, etc. Al tener ya esto, se puede calcular el puntaje del ejercicio, si tiene todas las respuestas

correctas, entonces su resultado es la nota máxima (100 para la UTFSM), de lo contrario, es la nota mínima (0 para la UTFSM).

Luego de que se haya revisado el código, el estudiante podrá ver su calificación y saber cuáles fueron los casos de prueba en que tuvo respuestas correctas o incorrectas.

Ver envíos

En caso de que el estudiante quiera revisar sus códigos escritos anteriormente o revisar sus calificaciones, puede revisar sus envíos de código realizados en la plataforma. De acuerdo con el correo del estudiante y a su identificador único, se puede saber cuáles fueron los códigos en que el estudiante participo, mostrándole todos los envíos con su correspondiente calificación.

En caso de que el estudiante quiera ver el código específico de un envío, puede entrar al envío correspondiente que le mostrara el código escrito por el estudiante, los casos de prueba realizados correctos e incorrectos, además de su calificación.

3.2.2 Profesores

La función de los profesores en la plataforma es bastante distinta en relación con los estudiantes, ya que son los encargados de establecer los talleres, de elegir los ejercicios y de establecer las parejas de los estudiantes. A continuación, se explican a detalle cada una de las funciones de los profesores en la plataforma:

Crear talleres

Los profesores podrán crear los talleres para que los estudiantes puedan realizarlos, eligiendo los ejercicios que ellos estimen convenientes para el desarrollo de estos talleres. Al momento de la creación de los talleres, se les pide el nombre del taller (principalmente, debería corresponder al tipo de ejercicio a realizar, por ejemplo, listas, diccionarios, archivos, etc.), la fecha de inicio y de término del taller, y la parte más importante, los ejercicios.

Al momento de seleccionar los ejercicios, deben ser ejercicios que pertenecen a la base de datos de ejercicios de programación. Pero también deben estar almacenados en la API de Django con el respectivo identificador del ejercicio en la base de datos de ejercicios de programación, con tal de obtener el ejercicio con su enunciado y sus casos de prueba. Para los talleres, se pueden elegir múltiples ejercicios que los estudiantes puedan realizar. A los estudiantes se les mostrarán todos los ejercicios de los talleres y ellos pueden elegir por cuál ejercicio comenzar, pudiendo revisar cada uno de los ejercicios, ver el enunciado y elegir en qué ejercicio comenzar a desarrollar la solución.

Verificar paralelos

Los profesores tienen la opción de verificar los estudiantes pertenecientes a cada paralelo, con tal de saber cuáles son los estudiantes que se han inscrito en la plataforma ingresando a través de la plataforma AULA de la UTFSM. En esta sección pueden ver a todos los estudiantes separados por paralelos, donde pueden ver el nombre completo de los estudiantes y también el correo de cada uno de ellos.

Crear y modificar parejas

Aunque la acción anterior no tenga mucha utilidad, la opción de poder verificar cada paralelo le permite a cada profesor crear las parejas de estudiantes que trabajaran juntos.

Al momento de crear las parejas, primero aparece la opción de asignación automática, es decir, que las parejas de estudiantes se asignan automáticamente de forma aleatoria, con tal de evitar la primera selección de parejas entre estudiantes.

A medida que avance el curso, muchos de los estudiantes no se sienten cómodos con sus actuales parejas o la pareja tuvo un problema para asistir a la experiencia, por lo cual, el profesor puede modificar la pareja de un estudiante. En caso de que eso ocurra, un estudiante va con el profesor, él dice su nombre y el nombre del estudiante que será su pareja y los asigna como pareja.

Como cada uno de los estudiantes mencionados tenían una pareja anteriormente, entonces ahora ellos serán pareja, es decir, un estudiante A es pareja de un estudiante B, y existe otra pareja de un estudiante C con un estudiante D. En el caso de que el estudiante A quiera ser pareja del estudiante C, ya que no se siente cómodo con el estudiante B, entonces ahora las nuevas parejas son el estudiante A con el estudiante C, y la otra pareja es entre el estudiante B y el estudiante D. Todo lo anterior se puede ver en el siguiente diagrama de cómo se modifican las parejas para evitar de que los anteriores estudiantes se queden sin parejas:

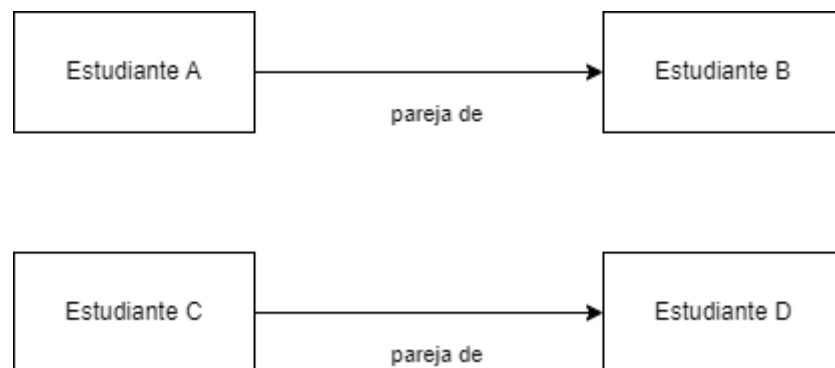


Figura 11: Parejas actuales. Fuente: Elaboración Propia

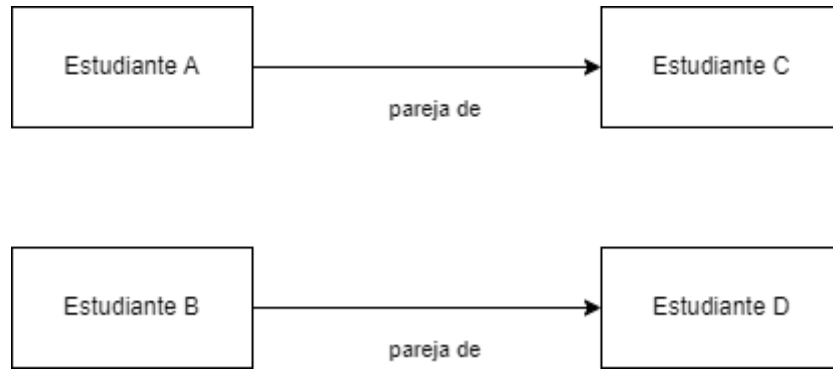


Figura 12: Parejas luego de la modificación. Fuente: Elaboración Propia

3.2.3 Sistema

El sistema también tiene tareas que son muy importantes, tales como la utilización de la metodología de programación de a pares entre los estudiantes y también la corrección de los ejercicios al momento de que los estudiantes envían sus códigos. A continuación, se detalla cada una de estas tareas realizadas por el sistema

Programación de a pares

La metodología de programación de a pares es el tema principal de este proyecto de memoria, por lo cual, implementar la utilización de esta fue el desafío más grande. El objetivo de la plataforma es que dos estudiantes puedan trabajar en conjunto en un mismo ejercicio, donde ambos puedan escribir en el código y visualizar lo que escribe el otro compañero en tiempo real, todo utilizando la metodología de programación de a pares.

Para implementarlo, se utilizaron Sockets. Esto permitieron la comunicación cliente-servidor entre los estudiantes. Primero, al estar las parejas almacenadas en la API de Django, el servidor del Socket solo permitía conexión entre estudiantes que fueran parejas. Por lo cual, al momento de ingresar al ejercicio, comprobaba si la pareja estaba conectada al ejercicio, para poder hacer el respectivo envío de lo escrito por uno de los dos estudiantes en el código.

Cuando un estudiante escribía en la zona de código, este código se enviaba al Socket, el cual es un servidor esperando respuestas de forma asíncrona. Luego, cuando recibía el código del estudiante, este Socket enviaba la respuesta a su pareja, mientras estuviera conectada al ejercicio, y se lo mostraba en la zona de código para que pudiera ver lo escrito por el otro estudiante. Así se producía la dinámica en cuanto a la programación de a pares.

No obstante, los estudiantes no podrían escribir al mismo tiempo, esto debido a la dinámica de la programación de a pares, donde uno de los estudiantes observa el código escrito mientras otro escribe el código. Para esto, se creó un piloto y un copiloto. El piloto

es quien escribe en la zona de código, mientras que el copiloto es quien observa el código escrito por el piloto, dando distintas formas de resolver el ejercicio y establecer la solución. Cuando ambos estudiantes ingresaban al ejercicio, se elegía de forma aleatoria este piloto y copiloto.

Al cabo de un tiempo almacenado en la variable `tiempo_programacion` de la tabla `talleres` de la API Django, tanto el piloto como el copiloto cambiaban sus estados. Ahora el piloto es el que antes era el copiloto y lo mismo para el copiloto actual. Esta es la dinámica de programación de a pares requerida para el respectivo funcionamiento.

Con respecto al cambio de función, también se hace a través del Socket donde, cuando pasa el tiempo señalado, se cambian las funciones entre ambos estudiantes y se habilita la zona de código para el nuevo piloto.

Todo lo relacionado con la programación de a pares se puede ver en el siguiente diagrama, con un paso a paso de como los estudiantes interactúan con la plataforma:

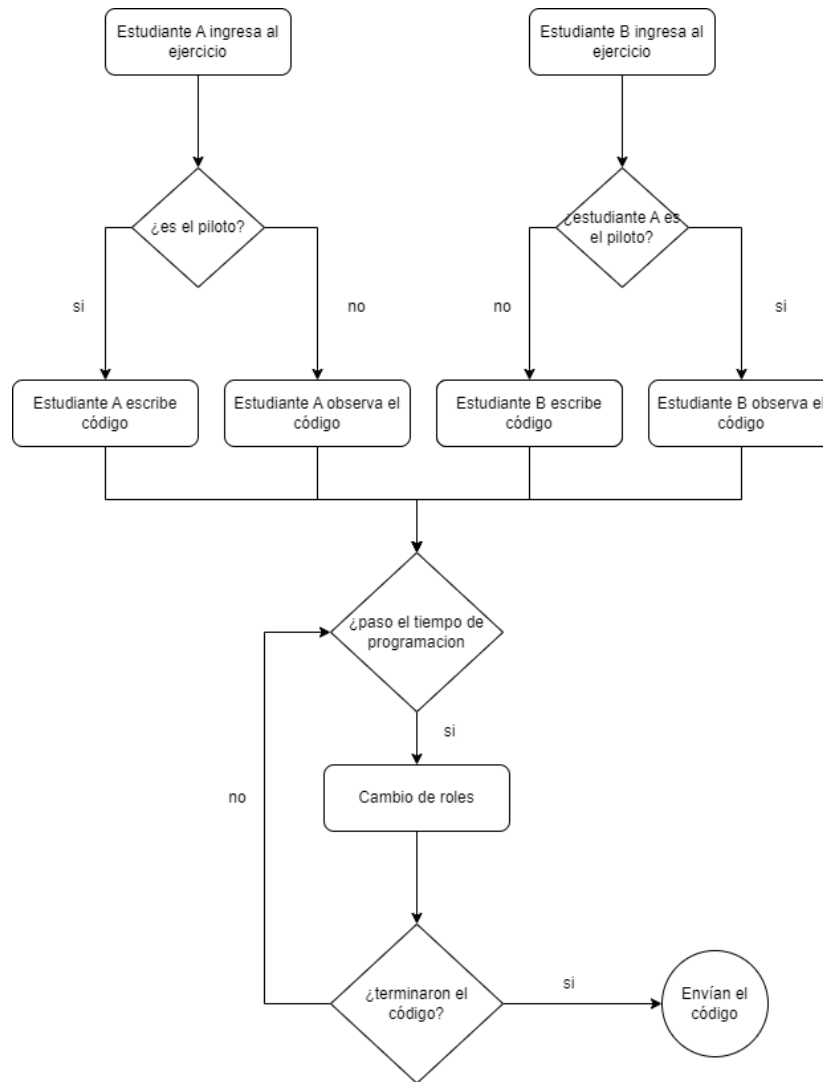


Figura 13: Metodología de programación de a pares. Fuente: Elaboración Propia

Corrección de ejercicios

El sistema también tiene una tarea que debe hacerse de forma automática y es la evaluación del código enviado por cada pareja de estudiantes. Al momento de hacer él envío, la plataforma guarda el código del estudiante en la API Django para ser evaluado por la API Judge0, la cual debe tener todos los casos de prueba del ejercicio y poder compararlos con los obtenidos por los estudiantes.

Cuando una pareja de estudiantes envía el código, entonces la API Judge0 recibe el código y comienza con las pruebas. Este proceso se hace por detrás, es decir, lo maneja la API Judge0, y cuando obtiene las respuestas de lo enviado por los estudiantes, la plataforma hace los cambios correspondientes. Primero, coloca la variable revisado en true, para que no vuelva a revisar el ejercicio, luego guarda las respuestas obtenidas por el Judge0 en una

variable `respuestas_judge0` de la API Django. Luego de tener ya las respuestas, calcula el puntaje obtenido.

Al momento de tener todo lo anterior almacenado, entonces es ahí cuando muestras los resultados obtenidos al estudiante. Cuando se encuentra revisando la respuesta, le avisa al estudiante que está en proceso de revisión.

3.3 Desarrollo del diseño

Para el desarrollo del diseño de la plataforma se ocupó la librería de JavaScript React junto con el framework Bootstrap. La utilización de estas tecnologías se debe a (Crónicas del dev, 2023):

- Facilidad de uso: sigue la sintaxis de React, siendo muy fácil de desarrollar el frontend.
- Diseño *Responsive*: las aplicaciones se adaptan al tamaño de la pantalla.
- Componentes: la utilización de componentes ya creados ahorra tiempo y esfuerzo al desarrollador.
- Personalización: los componentes mencionados pueden ser personalizados a través de CSS.

A continuación, se presentarán las vistas de la plataforma realizada con las tecnologías mencionadas anteriormente:

3.3.1 Vista de inicio

Con tal de seguir la temática de SMOJ, y siendo una plataforma paralela a SMOJ, la vista de inicio es muy similar a la vista de inicio de la plataforma SMOJ de la UTFSM. Se presenta un *Navbar* que será utilizado en todo el desarrollo de la plataforma, donde se pueden ver todas las opciones, tales como la vista de inicio, de talleres y la cuenta del usuario. En caso de que sea admin, puede tener acceso a la creación de talleres y la vista de los paralelos.



Figura 14: Vista de inicio (estudiante). Fuente: Elaboración Propia



Figura 15: Vista de inicio (administrador). Fuente: Elaboración Propia

La paleta de colores es la misma que se utiliza en SMOJ, donde el *backgroundColor del Navbar* es de color *maroon* y el fondo es de color blanco.

3.3.2 Vista de talleres

Para la vista de talleres, se utilizan componentes de tablas responsivas, almacenando todos los datos necesarios para el usuario que ve la tabla. Además, los talleres que se encuentren habilitados tienen un enlace para poder ingresar al taller además tener su estado en disponible, mientras que los talleres fuera de las fechas se encuentran finalizados y no se puede acceder a ellos

 Santa María Online Judge Inicio Talleres Admin Paralelo VALENTIN ANDRES RIVEROS CALDERON ▾			
Talleres			
Talleres por comenzar o corriendo			
Nombre	Inicio	Termino	Estado
TALLER	28/11/2023, 15:57:00	10/12/2023, 15:02:00	Disponible
Talleres finalizados			
Nombre	Inicio	Termino	Estado
Taller	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
Taller	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
Taller	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
nombre	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
UVA 1	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
UVA 9	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
UB	1/1/1970, 0:00:00	1/1/1970, 0:00:00	FINALIZADO
Taller Con Fecha	28/6/2023, 0:00:00	28/6/2023, 12:00:00	FINALIZADO

Figura 16: Vista de talleres. Fuente: Elaboración Propia

3.3.3 Vista de ejercicios

Al ingresar a un taller que se encuentre disponible, se tiene la vista de ejercicios del taller. En ella, primero se puede observar un *Nav* muy pequeñito donde se puede volver a la sección de talleres, se puede ver el taller en que uno se encuentra y también acceder a la opción de Mis envíos.

En la vista de Resumen, se pueden observar los ejercicios propuestos para el taller. En una tabla responsiva que contiene información como el nombre del ejercicio, el número de tareas a realizar en el ejercicio y el puntaje obtenido por el estudiante en caso de que ya haya hecho el ejercicio.

 Santa María Online Judge Inicio Talleres Admin Paralelo VALENTIN ANDRES RIVEROS CALDERON ▾			
Talleres	TALLER	Mis envíos	
Te recomendamos ver el video Laboratorios en SMOJ antes de intentar un laboratorio.			
Resumen			
Problema	Tarea	Puntaje Obtenido	
Cuadrados Mágicos (listas de listas)	1	0	

Figura 17: Vista de ejercicios. Fuente: Elaboración Propia

3.3.4 Vista de ejercicio específico

Cuando accedemos a un ejercicio de un taller, se puede observar el ejercicio a realizar. Se mantiene el mismo *Nav* señalado anteriormente con las mismas opciones. En primer lugar, se observa el modo del estudiante, es decir, si es piloto o copiloto. Luego se observa el tiempo de cambio entre piloto y copiloto, pasado ese tiempo, se cambian las funciones de cada uno. Luego, se observa el ejercicio, donde se tiene el título y la correspondiente descripción de este. También se señalan los datos de entrada y de salida, además de un ejemplo para que el estudiante tenga en consideración cuáles podrían ser casos de prueba.

Para la correcta visualización del ejercicio tanto en la descripción como en los casos de prueba, se utilizan *cards*, el cual es un componente de Bootstrap que permite tener la información que se requiera almacenada dentro de estas *cards*.

The screenshot displays the user interface of the Santa María Online Judge. At the top, a dark red navigation bar contains the logo and name 'Santa María Online Judge', followed by links: 'Inicio', 'Talleres', 'Admin', and 'Paralelo'. On the right side of this bar, the user's name 'VALENTIN ANDRES RIVEROS CALDERON' is shown with a dropdown arrow. Below the navigation bar, a secondary bar features three buttons: 'Talleres', 'TALLER' (highlighted in dark red), and 'Mis envíos'. The main content area has a large heading 'Piloto' in the center. Below this, a light gray box indicates 'Termina en: 00:30'. The primary content is enclosed in a white card with a title 'Cuadrados Mágicos (listas de listas)'. The text inside the card describes a magic square as a square matrix of integers where the sums of rows, columns, and main diagonals are equal, known as a magic number or constant. It provides an example of a 3x3 magic square with the constant 15:

8	3	4
1	5	9
6	7	2

Below the card, there is a dark red button labeled 'Tarea 1' and a light gray box also labeled 'Tarea 1'.

Entrada

La entrada comienza con un número entero positivo n , que representa el orden del cuadrado a evaluar.

A continuación vendrán $n \times n$ números enteros, que componen la matriz. Los primeros n números corresponden a la primera fila, los siguientes n a la segunda, y así sucesivamente.

Salida

Si la matriz ingresada es un cuadrado mágico, la salida debe ser un único número entero correspondiente a la constante mágica del cuadrado. En caso contrario, la salida debe ser la palabra NO (todo en mayúsculas).

Observaciones

Puede suponer que el valor de n será siempre un entero positivo. Asimismo, los valores ingresados para la matriz serán todos enteros. No es necesario hacer validaciones.

Ejemplo

Entrada
3
8
3
4
1
5
9
6
7
2
Salida
15

Figura 18: Vista de ejercicio específico. Fuente: Elaboración Propia

3.3.5 Vista de código

En la siguiente vista, que va junto con la vista del ejercicio anteriormente mostrado, está también el editor de código en tiempo real. Este editor se programó gracias a una librería de React llamada AceEditor. Con esta se puede tener el bloque de código para que los estudiantes puedan escribir sus códigos correspondientes. Además, se agregan dos botones, uno para poder probar el código con un caso de prueba y otro para enviarlo y terminar con el desarrollo.

En cuanto a la sección de código, esta se habilita si el usuario está en modo piloto, permitiendo la escritura en este bloque. De lo contrario, si el usuario es copiloto, solo podrá observar lo que escribe el piloto en esta misma sección, pero no podrá escribir en este bloque. Cuando se produce el cambio de roles, el nuevo usuario piloto ahora podrá escribir y el nuevo usuario copiloto solo podrá observar.

```

1 import ast
2 def mejores_jugadores(lista):
3     d = {}
4     for linea in lista:
5         nombre, pais, altura, peso, edad, club, habilidad, valor = linea.strip().split(";")
6         valor = int(valor[1:])
7         if pais not in d:
8             d[pais] = []
9         d[pais].append([valor, nombre, club])
10    l=[]
11    for pais in d:
12        d[pais].sort()
13        d[pais].reverse()
14        l.append([pais, d[pais][0][1], d[pais][0][0]])
15    l.sort()
16    return l
17
18 lista = ast.literal_eval(input())
19 print(mejores_jugadores(lista))

```

Probar Enviar

Figura 19: Vista de código. Fuente: Elaboración Propia

En caso de querer probar el ejercicio, se presiona el botón indicado y la API Judge0 comienza a evaluar el código con un caso de prueba específico. Al obtener la respuesta, se le indica al usuario si la respuesta está correcta o incorrecta. En caso de errores de compilación, estos se mostrarán en pantalla para que el estudiante sepa dónde está el error y por qué se produce el error.

Caso de prueba #1	Aceptado
Entrada	
	['Rafael da Silva;Brazil;179;73;31;Jeonbuk Hyundai;57;\$3000000\n', 'Marcel Halstenberg;Germany;188;87;31;RB Leipzig;62;\$5500000\n', 'Paulinho;Brazil;178;71;28;FC M...
Salida	
	[['Argentina', 'Tiago Casasola', 1600000], ['Brazil', 'Danilo', 4900000], ['Germany', 'Marcel Halstenberg', 5500000]]
Salida esperada	
	[['Argentina', 'Tiago Casasola', 1600000], ['Brazil', 'Danilo', 4900000], ['Germany', 'Marcel Halstenberg', 5500000]]
Feedback	

Figura 20: Vista de probar código. Fuente: Elaboración Propia

3.3.6 Vista de envío

Cuando el estudiante ya terminó el ejercicio, entonces debe enviarlo, lo cual lo mandará a la vista de envíos específicos. En ella, se puede ver el envío con su respectivo identificador, además de una tabla responsiva donde tiene el problema enviado, la fecha de cuando lo envió, el puntaje y si el código fue revisado o aún está en proceso de revisión.

En una card se puede ver el código escrito por el estudiante y luego se pueden ver cards con casos de pruebas. En estas se muestran tal como si se hubiera probado un código en la sección anterior, mostrando la entrada y la salida, además de la salida esperada para el ejercicio. Si la salida coincide con la salida esperada, entonces el resultado es aceptado. De lo contrario, será una respuesta incorrecta.

Ya que son múltiples casos de envío, entonces se pueden ver múltiples cards con varios casos de prueba. Para las siguientes cards con los casos de pruebas, estas se mostrarán bloqueadas, es decir, no saldrá ni la entrada, ni la salida, ni la salida esperada. Esto para evitar que los estudiantes sepan los casos de prueba al momento de entregar y enviar el código. Solo sabrán si el resultado fue aceptado o rechazado por la API Judge0 al revisar el ejercicio y los casos de prueba.


Santa María Online Judge
Inicio Talleres Admin Paralelo

VALENTIN ANDRES RIVEROS CALDERON ▾

Envío #28

Problema	Fecha	Puntaje	Veredicto
Medianas	14:20:04 21-09-2023	100	Aceptado

Código

```
def calcular_mediana(numeros): n = len(numeros) if n % 2 == 0: mediana = (numeros[n//2 - 1] + numeros[n//2]) / 2 else: mediana = numeros[n//2] return mediana
def main(): n = int(input()) numeros = [] for _ in range(n): numero = int(input()) numeros.append(numero) numeros.sort() mediana = calcular_mediana(numeros) resultado = int(mediana * 2)
print(resultado) if __name__ == "__main__": main()
```

Caso de prueba #1 Aceptado

Entrada

```
11
1
2
6
17
18
22
35
46
109
```

Caso de prueba #2 Aceptado

Entrada
Salida
Salida esperada
Feedback
 Verifique que su programa ordene correctamente la secuencia de valores ingresados y que distinga bien el término central de la muestra cuando la cantidad de elementos es impar.

Caso de prueba #3 Aceptado

Entrada
Salida
Salida esperada
Feedback
 Verifique que su programa ordene correctamente la secuencia de valores ingresados, que distinga bien los términos centrales de la muestra cuando la cantidad de elementos es par.

Figura 21: Vista de envío específico. Fuente: Elaboración Propia

En caso de que un estudiante quiera ver todos los envíos realizados por él, entonces podrá revisar la opción de Mis envíos. En esta vista se observa una tabla responsiva que muestra el identificador del envío, el problema realizado, la fecha de cuando fue entregado y el puntaje obtenido.

Si requiere ver un envío específico como la vista mostrada anteriormente, solo accede al enlace a través del identificador del envío.

 Santa María Online Judge Inicio Talleres Admin Paralelo VALENTIN ANDRES RIVEROS CALDERON ▾			
Talleres		Mis envíos	
#	Problema	Fecha	Puntaje
22	71	21:58:08 04-09-2023	40
23	86	12:05:00 05-09-2023	17
24	28	13:51:53 21-09-2023	0
25	28	13:54:06 21-09-2023	20
26	86	13:58:25 21-09-2023	0
27	86	14:19:02 21-09-2023	83
28	86	14:20:04 21-09-2023	100
29	28	21:41:33 24-09-2023	0
32	186	12:16:18 03-10-2023	80
34	187	23:30:51 13-11-2023	100

Figura 22: Vista de mis envíos. Fuente: Elaboración Propia

3.3.7 Vista de diseño de talleres

Para el caso de los administradores, principalmente serán los profesores o ayudantes que requieran utilizar la plataforma, ellos podrán crear los talleres para los estudiantes. En esta vista se puede observar distintos inputs, donde el administrador debe ingresar todos estos elementos para poder crear un taller.

Primero que todo, debe ponerle un nombre al taller, con el cual será identificado por los estudiantes. Luego poner una fecha de inicio y de término para enviar los resultados por parte de los estudiantes. En caso de pasarse de la fecha, este taller quedará deshabilitado. También se debe asignar un tiempo de programación, es decir, un tiempo para el intercambio de roles de los estudiantes, pasando de piloto a copiloto. Y, por último y lo más importante, la selección de ejercicios, donde se podrán seleccionar varios ejercicios que se requieran para el desarrollo del taller.

Al momento de agregar, este será almacenado en la API Django y quedará disponible en la sección de talleres.

Admin Talleres

Nombre:

Nombre

Fecha y hora de inicio:

dd-mm-aaaa --:--

Fecha y hora de término:

dd-mm-aaaa --:--

Tiempo de Programación (minutos):

5

Ejercicio:

Cuadrados Mágicos (listas de listas)

Agregar

Figura 23: Vista de diseño de talleres. Fuente: Elaboración Propia

3.3.8 Vista de paralelos

Para el caso los administradores, estos también podrán observar los paralelos de estudiantes. Donde se muestran todos los paralelos que tienen alumnos registrados, y en cada uno de ellos se utiliza una tabla responsiva donde se ve el nombre del alumno y el correo institucional que utilizó para ingresar a la plataforma.

Paralelos

Nombre	Correo
DIEGO PABLO LIZANA GUTIERREZ	dlizana@usm.cl

Paralelo 9

Nombre	Correo
MATIAS ALEJANDRO VILLALOBOS DIAZ	mvillalobosd@usm.cl

Paralelo 10

Nombre	Correo
FRANCO ANDRES APABLAZA CANO	fapablaza@usm.cl

Paralelo 14

Nombre	Correo
FERNANDA ISABEL DEL PILAR LEAL AVILA	fernanda.leal@usm.cl

Figura 24: Vista de paralelos. Fuente: Elaboración Propia

También se pueden observar las parejas de cada paralelo ingresando a la opción de parejas. En este caso solo se mostrará el paralelo seleccionado y una tabla que muestra los nombres de los estudiantes con sus respectivas parejas. En caso del estudiante no tenga una pareja asignada, no se muestra el estudiante 2.

Parejas

Paralelo 208

Estudiante 1	Estudiante 2
FELIPE ANDRE LEPIN CAMPOS	VALENTIN ANDRES RIVEROS CALDERON
HERNÁN ALCINO CABRERA MUÑOZ	NATALIA CAROLINA VILLAGRA IRRIBARRA
BENJAMIN ARTURO MARTINEZ LAYSECA	SEBASTIAN NICOLAS HERNANDEZ LARRAIN
MARTINA ZUANIC BUSTOS	SEBASTIAN RODRIGO MEDINA DONOSO
MATIAS IGNACIO CIFUENTES ZARATE	
ISIDORA CATALINA ZUÑIGA LOBOS	MARTINA JESUS CAMUS DIAZ
DANIEL ALONSO SALINAS VALDES	MARIA JOSE GOMEZ CACERES
CAMILA ISIDORA NOGUERA HAGEN	EZEQUIEL ABRAHAM ROJAS POMBET

[Editar Parejas](#)
Figura 25: Vista de parejas. Fuente: Elaboración Propia

En caso de que los estudiantes quieran cambiar de pareja o algún integrante de una pareja no asistió a la experiencia realizada con la plataforma, entonces se pueden editar las parejas. Se tienen dos inputs, uno para el estudiante 1 y otra para el estudiante 2. Al seleccionar el estudiante 1, el estudiante 2 cambia a la pareja que tiene el estudiante 1, y se puede cambiar por cualquier estudiante perteneciente al paralelo.

Al presionar el botón de cambiar parejas, se cambian las parejas con la misma lógica señalada en la sección Crear y Modificar Parejas de este documento. Estas nuevas parejas se modifican en la API Django.

Editar Parejas

Estudiante 1:

Estudiante 2:

[Cambiar Pareja](#)
Figura 26: Vista de edición de parejas. Fuente: Elaboración Propia**3.3.9 Vista de selección de paralelos**

En caso de que el estudiante no este registrado en la plataforma, entonces al momento de ingresar, se pregunta a la API Django si existe el estudiante. Si no existe, aparece esta vista de selección de paralelo donde se obtiene el nombre y el correo institucional desde la plataforma AULA (a través de LTI), y luego se agrega un *input* donde debe seleccionar el paralelo al cual pertenece el estudiante. Cuando asigna el paralelo, se envía a la API Django y el estudiante se encontrará registrado.

Si un estudiante registrado accede a la plataforma, esta lo envía directamente a la vista de inicio.

The screenshot shows a web interface for a user profile. At the top is a dark red navigation bar with the logo 'Santa María Online Judge' and links for 'Inicio', 'Talleres', 'Admin', and 'Paralelo'. On the right of the bar is a user dropdown menu showing 'VALENTIN ANDRES RIVEROS CALDERON'. Below the navigation bar, the title 'Usuario' is centered. The form contains three input fields: 'Nombre' with the value 'VALENTIN ANDRES RIVEROS CALDERON', 'Correo' with the value 'valentin.riveros@sansano.usm.cl', and 'Paralelo' which is currently empty. A blue 'Guardar' button is located at the bottom left of the form.

Usuario

Nombre:

VALENTIN ANDRES RIVEROS CALDERON

Correo:

valentin.riveros@sansano.usm.cl

Paralelo:

Paralelo

Guardar

Figura 27: Vista de selección de paralelos. Fuente: Elaboración Propia

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

Con respecto a la validación de la solución propuesta, se hicieron distintas actividades con los estudiantes del curso de programación, principalmente con el paralelo 208 del segundo semestre en el año 2023. Los estudiantes que realizaron las actividades son estudiantes con conocimientos básicos de la programación, que cursan el primer curso de programación de la universidad. Dentro de las actividades que se realizaron están las siguientes:

- Habilitación de prototipo de la plataforma
- Habilitación de plataforma antes de la segunda evaluación del curso
- Encuesta para los estudiantes luego de utilizar la plataforma

A continuación, se detallan cada una de estas actividades:

4.1 Habilitación de prototipo

La habilitación del prototipo se hizo con la intención de poder probar las funcionalidades de la plataforma en un entorno fuera del local, es decir, que otras personas pudieran utilizar la plataforma sin problemas antes de hacer las pruebas importantes.

Un mes antes de la segunda evaluación del curso, los estudiantes tuvieron acceso a la plataforma en una sesión de ayudantía. En esta sesión, los estudiantes pudieron registrarse en la plataforma y realizar un ejercicio propuesto a través de un taller. La cantidad de estudiantes presentes en la habilitación del prototipo fue de 6, en donde se encontraban desarrollando la tarea de la semana, por lo cual, centraron todo su conocimiento de la tarea en el ejercicio del prototipo.

La plataforma estaba en un 90% de desarrollo, por lo cual, los estudiantes encontraron ciertas observaciones al hacer el ejercicio. Estas fueron:

- Problemas a la hora de cambiar los roles de programación.
- No salía el error de código específico de un código en caso de error de sintaxis.

En cuanto a los aspectos positivos:

- La interfaz era muy parecida a SMOJ, lo cual permitió que los estudiantes encontraran los ejercicios de forma fácil.
- La dinámica les gusto, generando debate entre los estudiantes que realizaban los códigos.

4.2 Habilitación de plataforma antes de la segunda evaluación del curso

Una semana antes de la segunda evaluación del curso, los estudiantes tuvieron acceso a la plataforma nuevamente en una sesión de ayudantía que estaba a cargo de mi persona. En esta sesión de ayudantía, luego de revisar un código de ejercicio para el certamen, les propuse a los alumnos que ingresaran a esta plataforma, la cual pudieron acceder a través de AULA y poder realizar el ejercicio propuesta.

El ejercicio propuesto era de archivos de texto, donde los estudiantes debían analizar un archivo de texto, hacer las correspondientes separaciones y poder crear un diccionario de listas con mejores jugadores de futbol ordenados por país. El tiempo de desarrollo del ejercicio fue de aproximadamente 40 minutos para todos los estudiantes presentes en la sesión de ayudantía.

4.2.1 Resultados generales

De acuerdo con lo anterior, se obtuvieron los siguientes resultados:

Descripción	Cantidad
Estudiantes presentes	12
Envíos realizados	6
Estudiantes del curso	30
Participación en %	40%

Tabla 1: Resultados generales. Fuente: Elaboración Propia

Considerando que la cantidad de alumnos del paralelo del curso era de 30 personas, en la ayudantía estuvieron presentes 12 alumnos, de los cuales corresponde al 40% de participación del curso.

Los envíos realizados fueron 6, esto debido a que la cantidad alumnos eran 12 y era una plataforma diseñada para parejas, por lo cual, eran un total de 6 parejas.

4.2.2 Resultados de envíos realizados

En cuanto a los envíos realizados, los resultados fueron los siguientes:

Puntaje	Cantidad
Aceptado (100)	2
Intermedio (50)	1
Incorrecto (0)	3

Tabla 2: Resultados de envíos realizados. Fuente: Elaboración Propia

En estos resultados de envíos se pudo observar que tan preparados estaban los alumnos para la segunda evaluación del curso. Dos parejas obtuvieron el puntaje máximo, 1 tuvo el puntaje intermedio y 3 parejas obtuvieron resultados incorrectos.

En cuanto a las parejas que obtuvieron resultados incorrectos, los principales problemas fueron falta de tiempo y poca coordinación con su compañero. Como el tiempo fue de aproximadamente 40 minutos, a muchos de los estudiantes les tomo la mitad de ese tiempo en entender el ejercicio y la otra mitad en desarrollarlo. También, como factor adicional, en ocasiones tenían que explicarle el problema al compañero, ya que se encontraba bastante más perdido en la búsqueda de la solución.

En cuanto al veredicto de Judge0, los resultados fueron los siguientes:

Veredicto	Cantidad
<i>Accepted</i>	5
<i>Wrong Answer</i>	1
<i>Runtime Error (NZEC)</i>	6

Tabla 3: Resultados veredicto de Judge0. Fuente: Elaboración Propia

Estos resultados son en base a los casos de prueba del ejercicio, donde el ejercicio tenía 2 casos de prueba. De los estudiantes que obtuvieron la calificación máxima, salen los 4 *Accepted*, el otro sale de la pareja que obtuvo la calificación media, es decir, que tuvo una respuesta correcta y otra incorrecta.

Los otros resultados, principalmente de *Runtime Error* (error en tiempo de ejecución), se debe principalmente a que los estudiantes no alcanzaron a terminar la solución propuesta, enviando su resultado y saltando error en tiempo de ejecución.

4.3 Encuesta a los estudiantes

Luego de la realización de la experiencia mencionada, los estudiantes tuvieron que contestar una encuesta para saber si estuvieron conformes con la actividad realizada.

En la encuesta, se realizaron las siguientes preguntas a los estudiantes:

- ¿Cómo calificarías tu experiencia general con la plataforma?
- ¿Cómo estuvo la experiencia con tu compañero?
- ¿Qué tan fácil es usar la plataforma?

- ¿Crees que esta metodología es buena para aprender a programar, comparada con trabajar de forma individual?
- ¿Crees que esta metodología te ayudo a entender mejor el problema y encontrar su solución?
- ¿Recomendarías esta metodología a estudiantes que empiezan a programar?
- ¿Encontraste algún problema técnico en la aplicación?
- ¿Qué características te gustaría añadir en las próximas versiones de la aplicación?
- Comentarios adicionales

4.3.1 Resultados de la encuesta

De acuerdo con las respuestas enviadas por los estudiantes, los resultados de la encuesta son los siguientes:

- ¿Cómo calificarías tu experiencia general con la aplicación?

Experiencia	Cantidad	Porcentaje
Muy mala	0	0%
Mala	0	0%
Normal	0	0%
Buena	2	17%
Muy buena	10	83%

Tabla 4: Resultados de experiencia. Fuente: Elaboración Propia

- ¿Cómo estuvo la experiencia con tu compañero?

Experiencia	Cantidad	Porcentaje
Mala	0	0%
Normal	1	8%
Buena	11	92%

Tabla 5: Resultados de experiencia con compañero. Fuente: Elaboración Propia

- ¿Qué tan fácil es usar la plataforma?

Dificultad	Cantidad	Porcentaje
Muy fácil	4	33%
Fácil	2	17%
Medio	2	17%
Difícil	2	17%
Muy difícil	2	17%

Tabla 6: Resultados de facilidad de uso. Fuente: Elaboración Propia

- ¿Crees que esta metodología es buena para aprender a programar, comparada con trabajar de forma individual?

	Cantidad	Porcentaje
Si	12	100%
No	0	0%

Tabla 7: Resultados de metodología. Fuente: Elaboración Propia

- ¿Crees que esta metodología te ayudo a entender mejor el problema y encontrar su solución?

	Cantidad	Porcentaje
Si	12	100%
No	0	0%

Tabla 8: Resultados de entendimiento del problema. Fuente: Elaboración Propia

- ¿Recomendarías esta metodología a estudiantes que empiezan a programar?

	Cantidad	Porcentaje
Si	12	100%
No	0	0%

Tabla 9: Resultados de recomendación. Fuente: Elaboración Propia

- ¿Encontraste algún problema técnico en la aplicación?

Solo un estudiante encontró un problema técnico en la plataforma que fue relacionado con un bucle infinito de cambios de roles entre piloto y copiloto cuando cerro la ventana de la plataforma sin querer. Todos los demás estudiantes no tuvieron problemas con la plataforma.

- ¿Qué características te gustaría añadir en las próximas versiones de la aplicación?

Algunas de las respuestas de los estudiantes fueron:

- Que cada uno edite con colores diferentes.
- Un chat para poder hacer el trabajo a distancia.
- Poder escoger cuando escribir, en vez de tener un tiempo limitado.
- Poder pasar el mando de copiloto a piloto.

Muchas de estas respuestas van en contra de la metodología utilizada, ya que escoger cuando escribir puede provocar que solo uno de los alumnos escriba en el código, dejando al otro sin poder hacer nada. Por ejemplo, en el caso de una pareja donde uno de los estudiantes sabe programar y el otro no, entonces el que sabe tomará todo el código y lo hará solo.

- Comentarios adicionales

Algunos de los comentarios de los alumnos respecto a la plataforma fueron los siguientes:

- Muy buena aplicación.
- Buena la dinámica y metodología en que se desempeña la actividad.
- Muy divertida.
- Hacerlo más seguido.
- Agradable de trabajar.
- Hace más fácil aprender a programar.

4.4 Resultados en base a evaluación del curso

Respecto a los estudiantes que realizaron la actividad, los resultados que estos obtuvieron en el segundo certamen del curso fueron los siguientes:

Nota	Cantidad	Porcentaje
100-75	1	8%
75-50	9	75%
50-25	2	17%
25-0	0	0%

Tabla 10: Resultados de certamen 2. Fuente: Elaboración Propia

En base a estos resultados, la implementación de programación de a pares permitió que aproximadamente un 80% de los estudiantes pudieran aprobar la evaluación correspondiente a los alumnos que asistieron a la ayudantía en donde se realizó la actividad con la plataforma.

CAPÍTULO 5: CONCLUSIONES

En este documento de memoria, los estudiantes del curso de programación en la UTFSM pudieron practicar sus conocimientos en el curso a través de una nueva metodología, llamada programación de a pares. En esta metodología, los estudiantes podrían trabajar en parejas, donde uno de los estudiantes debe escribir el código mientras el otro observa el código que escribió el compañero con tal de detectar distintos errores en la implementación o en la sintaxis de la programación.

Con la finalidad de implementar la metodología, se implementó una plataforma la cual permite integrar la metodología de forma automática y con ejercicios ya establecidos. La plataforma se creó principalmente con *React* y *Django*, donde los lenguajes de programación utilizados fueron *JavaScript* y *Python* respectivamente.

Los datos se manejaron a través de *Django*, la cual nos permitió tener una arquitectura similar a bases de datos como *MySQL* o *PostgreSQL*, pero en forma local a través de una API que almacenaba los datos correspondientes. En cuanto a las vistas de la plataforma, esta se hizo con *React Bootstrap*, la cual es un framework que permite la reutilización de código en relación con componentes para poder utilizarse en la creación de las vistas del diseño. En relación con la metodología de programación de a pares, se utilizaron *Sockets*, los cuales nos permiten la comunicación cliente-servidor entre las distintas parejas que acceden a la plataforma y puedan escribir el código en dos computadores distintos y respetando las reglas de la metodología planteada. Además, para la implementación de la revisión de código, se utilizó una API llamada Judge0 la cual se implementó utilizando Docker para poder correr el servidor de la API. En ella, se hace la evaluación correspondiente de los códigos escritos por los estudiantes utilizando distintos casos de prueba y entregando si la respuesta fue correcta o incorrecta de acuerdo con estos casos.

Las principales características de la plataforma es la implementación de una nueva metodología para el curso de programación, la cual es la programación de a pares. En esta plataforma, los estudiantes podrán escribir código de programación y será revisado automáticamente por la API Judge0 al momento de enviar sus resultados. Los estudiantes tendrán asignados dos roles, el piloto y el copiloto, el cual se irán intercaldando pasado un tiempo de programación. El piloto es quien escribe el código mientras el copiloto es quien observa el código escrito por el piloto, pero todo en dos computadores distintos, por lo cual, el copiloto puede ver lo que escribe el piloto desde su mismo computador. Pasado el tiempo de programación, los roles se intercambian y el piloto es el antiguo copiloto y lo mismo con el copiloto. Así sucesivamente hasta terminar la solución del ejercicio y enviar su respuesta para la evaluación.

Las pruebas de la plataforma se hicieron una semana antes de la segunda evaluación del ramo, con tal de ver como estaban los estudiantes antes de rendir la evaluación, y si es que esta plataforma pudo ayudar a practicar en resolución de ejercicios a los estudiantes que tomaran la evaluación.

En cuanto a los envíos realizados por los estudiantes, el 50% de los estudiantes obtuvieron calificaciones positivas en la plataforma, mientras que el otro 50% obtuvieron calificaciones negativas en las pruebas realizadas. Muchos de los problemas fueron relacionados a la falta de tiempo que tuvieron los estudiantes para resolver el ejercicio, el cual solo disponían de 40 minutos aproximadamente para resolver el ejercicio de práctica.

Luego de la realización de los ejercicios de prueba, se realizó una encuesta a los estudiantes para ver si estaban conformes con el desarrollo de la actividad y distintos aspectos en relación con la plataforma. Dentro de los distintos resultados obtenidos en la encuesta es que la experiencia fue muy buena para los estudiantes y que la metodología implementada funcionaba para estudiantes que estuvieran aprendiendo a programar por primera vez. Otros aspectos señalados fue que la experiencia fue muy divertida, que se debería hacer más seguido y que es agradable de trabajar.

Respecto a los resultados de la segunda evaluación del curso y si es que la plataforma sirvió como una metodología de práctica, aproximadamente el 80% de los estudiantes que estuvieron presentes en la experiencia pudo aprobar su evaluación. Por lo cual, se concluye que la experiencia fue un éxito y que permitió que los estudiantes tuvieran otra metodología de práctica basada en la participación entre estudiantes.

Algunos de los beneficios que esta metodología pudo verificar es principalmente la interacción entre estudiantes del curso, donde ahora pueden compartir ese conocimiento entre ellos, y la resolución de ejercicios se hace mas compartida. Tal como lo ejemplifican los resultados obtenidos en el capítulo anterior, los estudiantes que estuvieron presente en la experiencia señalaron que la actividad permite aprender más programación, especialmente en principiantes, además de recomendar la metodología para próximas versiones del curso.

5.1 Consideraciones y limitaciones

Algunas de las consideraciones explicadas por algunos de los estudiantes que estuvieron presentes en la experiencia es algunos problemas en el desarrollo de la metodología de programación de a pares. Esto ocurre principalmente cuando existe un trabajo en curso y uno de los estudiantes se sale o cierra la ventana en donde estaba realizando el código. Al pasar eso y volver a ingresar, la pareja tiene problemas al escribir el código debido a que existe un bucle infinito en el cambio de roles.

De acuerdo con la encuesta realizada, muchos de los estudiantes tuvieron problemas de dificultad con la plataforma, es decir, que tomo mucho tiempo en buscar los distintos elementos para poder realizar el ejercicio. Un 50% de los estudiantes considera que la interfaz de la plataforma es de fácil utilización, mientras que el otro 50% tuvo problemas al encontrar el ejercicio propuesto para la actividad.

5.2 Trabajos a futuro

Para el futuro desarrollo de la plataforma, primero que todo es mejorar en distintos aspectos de la plataforma, principalmente en la implementación de la metodología, con tal de que no existan los errores vividos por los estudiantes que estuvieron presentes en la experiencia.

También, mejorar en distintos aspectos de la interfaz para que sea de fácil interpretación y los estudiantes puedan trabajar sin problemas a la hora de buscar los ejercicios.

En relación con la encuesta, los estudiantes dieron algunas características importantes a señalar para el desarrollo futuro de la plataforma. La primera de ellas es la implementación de un chat entre las parejas, con tal de que puedan comunicarse entre las parejas si tuvieran que hacer actividades a distancia, permitiendo el intercambio de ideas entre piloto y copiloto.

Para finalizar, otra futura implementación para la plataforma sería la utilización de colores diferentes para cada uno de los estudiantes, con tal de saber que escribió cada uno de los estudiantes. Esto permite saber que escribió cada uno y saber por qué escribió esto.

REFERENCIAS BIBLIOGRÁFICAS

- Aboyi, P. (2021). *A List of the Top 8 Remote Pair Programming Tools*.
- Arguedas Negrini, I., (2010). INVOLUCRAMIENTO DE LOS ESTUDIANTES EN EL PROCESO EDUCATIVO. REICE. Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación, 8(1), 63-78.
- Codeanywhere. (2022). *Codeanywhere*.
- CodeTogether. (2022). *CodeTogether*.
- Crónicas del dev. (2023). ¿Qué es React-bootstrap? ¿Vale la pena?
<https://cronicasdeldev.ar/post/react-bootstrap>
- Despina Tsompanoudi, M. S. (2016). *Evaluating the Effects of Scripted Distributed Pair Programming on Student Performance and Participation*.
- Deyimar A. (2023). ¿Qué es React? Definición, Características y Funcionamiento. Hostinger. <https://www.hostinger.es/tutoriales/que-es-react>
- Executrain. (2022). *¿Qué es programación?*
- Hrastinski, Stefan. 2009. A Theory of online learning as online participation. En *Computers & Education*. Vol. 52, no. 1, 78-82. <<https://doi.org/10.1016/j.compedu.2008.06.009>>
- Hugo Leyton, A. B. (2017-2022). *Santa María Online Judge*.
- IBM. (2023). Conceptos de Sockets. En IBM Documentación. Recuperado de <https://www.ibm.com/docs/es/aix/7.3?topic=concepts-sockets>
- IONOS. (2022). *Pair Programming: principio de los cuatro ojos en el desarrollo de software*.
- KARTHIKEYAN UMAPATHY, A. D. (2017). *A Meta-Analysis of Pair Programming in Computer Programming Courses: Implications for Educational Practice*.
- Kessler, L. A. (2000). *All I Really Need To Know About Pair Programming I Learned In Kindergarten*.
- Lewis, C. M. (2011). *Is pair programming more effective than other forms of collaboration for young students?*
- Make it in Ukraine. (2021). *14 Best Tools for Remote Pair Programming in 2021*.

Mathieu, M. J. (2014). *Introducción a la programación*. Delegación Azcapotzalco, Código Postal 02400, México, D.F.: Grupo Editorial Patria.

Meza, F. (2022). *Programa de Asignatura Programacion 2022*. Valparaiso.

Meza, F. (2022). *Reglas del curso Programación*. Valparaiso.

Microsoft. (2022). *Visual Studio Live Share*.

Polanco Hernández, A., (2005). La motivación en los estudiantes universitarios. Revista Electrónica "Actualidades Investigativas en Educación", 5(2), 1-13.

Rodríguez Cortes, R., (2021). *El miedo al compromiso: sus causas y cómo superarlo*. PsicoGlobal. <https://www.psicoglobal.com/blog/miedo-compromiso-causas-terapia>

Servicio Nacional de Capacitación y Empleo (SENCE). (2021). Trabajo en equipo - Infografía [PDF]. SENCE. https://sence.gob.cl/sites/default/files/trabajo_en_equipo_-_infografia_carta.pdf

Tokio School. (2022). ¿Qué es Django y para qué se utiliza? Tokio School. <https://www.tokioschool.com/noticias/que-es-django/>

Universidad Tecnica Federico Santa Maria. (2023). Guía del Estudiante 2023. <https://dee.usm.cl/wp-content/uploads/2023/01/Guía-de-Estudiante-2023-USM.pdf>

Verdaguer J. (2021). Qué es LTI y cómo puede mejorar su ecosistema de aprendizaje. <https://moodle.com/es/news/que-es-lti-y-como-puede-mejorar-su-ecosistema-de-aprendizaje/>