

UNIVERSIDAD DE LA REPÚBLICA

FACULTAD DE INGENIERÍA

Informe de Proyecto de Grado
de la carrera Ingeniería en Computación

EMMY

Autores:

Verónica BENTANCOR

Gastón BARRETO

Supervisores:

Sylvia DA ROSA

Marcos VIERA



**FACULTAD DE
INGENIERÍA**

Universidad de la República

20 de agosto de 2021 Montevideo, Uruguay

Resumen

Este proyecto consiste en diseñar e implementar un entorno virtual educativo para matemáticas y ciencias. Está dirigido especialmente pero no exclusivamente a profesores y estudiantes de educación media o estudiantes ingresantes a la educación terciaria.

Se aspira a proveer a estudiantes y profesores funcionalidades que contribuyan a la integración entre la matemática y la programación. Se brinda entre otras, la posibilidad de publicar ejercicios y utilizar el lenguaje de programación MateFun, diseñado por investigadores del Instituto de Computación en colaboración con el Centro Interdisciplinario en Cognición para la Enseñanza y el Aprendizaje. Este es un lenguaje funcional desarrollado especialmente para el aprendizaje matemático de estudiantes preuniversitarios.

La realización de este proyecto incluyó la especificación de requisitos, el diseño, la implementación, la validación con pruebas de usuarios y el despliegue.

En este proyecto se utiliza Ruby on Rails, además de HTML5, CSS, JavaScript, JQuery y Bootstrap. Además se implementa la conexión con un servicio externo para la ejecución y envío de código MateFun.

Se implementó la mayor parte de la plataforma, mientras que el intérprete y las gráficas fueron importados desde otro proyecto programado en Angular, cuyo código fue convertido a JavaScript. Se siguieron las convenciones y guías de diseño de Ruby on Rails para mantener un código fuente claro y legible. Para el diseño de la interfaz, se tuvieron en cuenta aspectos de usabilidad e interacción persona computadora.

Se diseñaron pruebas de usuario que se llevaron a cabo mediante entrevistas con docentes (algunos contaban con conocimiento previo de MateFun) y estudiantes. Los docentes que tenían conocimiento en MateFun realizaron las pruebas en solitario, mientras que al resto de los docentes se les realizaron pruebas guiadas por medio de videoconferencia. A los estudiantes, se les realizaron entrevistas presenciales. Al finalizarlas, se les envió a cada participante un cuestionario para recolectar datos cuantitativos y cualitativos de la plataforma.

Este proyecto culmina con futuras líneas de desarrollo que surgen no solo de la experiencia de los autores si no también aportes de los entrevistados.

Índice general

1. Resumen	1
2. Agradecimientos	6
3. Introducción	7
3.1. Motivación	7
3.2. Antecedentes	8
3.3. Objetivos	8
3.4. Etapas del proyecto	9
4. Organización del documento	10
5. Análisis de requerimientos	11
5.1. Relevamiento	11
5.1.1. NRICH	11
5.1.2. Project Euler	12
5.1.3. Rosalind	12
5.1.4. LeetCode	14
5.1.5. Moodle	14
5.2. Conclusiones del relevamiento	17
6. Evolución de la especificación	19
6.1. Definición inicial y su evolución	19
6.2. Especificación	20
7. Diseño	22
7.1. Modelo de dominio	22
7.2. Arquitectura	22
7.3. Interfaces de usuario	24
7.3.1. Página principal	24
7.3.2. Navegación por tema, grado o docente	24
7.3.3. Listar publicaciones	25
7.3.4. Perfil docente	25
7.3.5. Crear publicación	26

7.3.6.	Responder publicación	27
7.3.7.	Listar y ver soluciones	27
8.	Implementación	28
8.1.	Equipo de desarrollo	28
8.2.	Ruby on Rails	29
8.3.	Base de datos	29
8.4.	Reutilización de código y refactorización	29
8.5.	Arquitectura MVC	30
8.6.	Pruebas de código	32
8.7.	Búsqueda de publicaciones	34
8.8.	Integridad de datos, validación y seguridad	36
8.9.	Frontend	38
8.9.1.	Tecnologías y herramientas utilizadas	38
8.10.	Editor de código e interprete	39
8.11.	Gráficas	39
8.12.	Editor de texto enriquecido	40
8.13.	Ejercicios interactivos	41
8.14.	Almacenamiento de imágenes	41
8.15.	Fechas y zonas horarias	41
8.16.	Paginación de resultados	41
8.17.	Código Secreto	42
8.18.	Breadcrumbs	42
8.19.	Listas desplegadas autocompletables	42
9.	Diseño del sitio	43
9.1.	Diseño adaptable	43
9.2.	Diseño de las interfaces	44
9.2.1.	Página principal	46
9.2.2.	Estudiantes	54
9.2.3.	Usuarios	56
9.2.4.	Administradores	57
9.2.5.	Docentes	60

10. Validación	66
10.1. Pruebas con usuarios	66
10.1.1. Iteraciones	69
10.1.2. Formulario de evaluación	69
10.1.3. Resultados	69
11. Conclusiones	71
12. Futuras líneas de desarrollo	74
12.1. Nuevas funcionalidades	74
12.1.1. Disciplinas	74
12.1.2. Internacionalización	74
12.1.3. Botón de ayuda	75
12.1.4. Agregar filtros por fecha	75
12.1.5. Filtros y ordenación	75
12.1.6. Subir imágenes	75
12.1.7. Enviar respuesta otro docente	75
12.2. Sistema de usuarios	76
12.2.1. Calificaciones	76
12.3. Publicaciones	78
12.3.1. Tipos de publicaciones	78
12.3.2. Editor de texto	78
12.3.3. Campos interactivos	79
12.3.4. Respuestas	79
12.4. Editor de código MateFun e intérprete	81
12.5. Entorno de gráficas	82
12.6. Extensión con MateFun Infantil	83
Anexos	88
A. Documento de requisitos	89
B. Bocetos del sitio	96
C. Manual de usuario	104

D. Documento de pruebas de usuario	119
E. Documento de implementación	144

Agradecimientos

Se agradece a la institución educativa Liceo IEP de El Pinar; a Estela Rosano por el contacto con algunos los docentes que participaron de las pruebas de interacción y a cada uno los participantes por su tiempo y dedicación: Agustín Breccia, Eliana Pérez, Erika Poo, Daniel Grillo, María Eugenia Curi, Jeannine Maufinet, Javier Wagner, Sandra Viglione y Yenny Adinolfi; así como también a nuestros familiares que participaron de las pruebas.

Se agradece también a los tutores del proyecto por su disponibilidad, voluntad y trabajo durante todos estos meses.

Introducción

En este documento se presentará una introducción al trabajo de Proyecto de Grado, edición 2020 (segundo semestre), de la carrera Ingeniería en Computación, de la Facultad de Ingeniería, Universidad de la República.

Este proyecto consistió en desarrollar un entorno virtual integrado para publicación y resolución de ejercicios de matemática y ciencias. Se desarrolló entre los meses de septiembre 2020 y julio 2021 por los estudiantes Verónica Bentancor y Gastón Barreto.

Incluyó etapas iterativas como: relevamiento, definición de requisitos, desarrollo y validación como pruebas con docentes y estudiantes.

3.1. Motivación

Con este proyecto se pretende aplicar los conocimientos adquiridos durante la carrera Ingeniería en Computación, realizando un aporte a la enseñanza desde nuestro rol como estudiantes, abarcando etapas de la ingeniería de software, diseñándolo y desarrollándolo desde su concepción inicial, hasta su validación con usuarios.

Actualmente, el aprendizaje en línea está cobrando mucha importancia, en gran parte debido a la virtualidad a causa de la pandemia. Esto se presenta como una oportunidad para desarrollar nuevas herramientas educativas. Estas plataformas son de utilidad para los docentes, a los cuales beneficia poder compartir los contenidos educativos con sus estudiantes, y también a estos, que les permite acceder a sus contenidos.

La computación juega un rol cada vez más importante en la enseñanza de las ciencias, ya que permite acceder más fácilmente a sus contenidos, así como generar nuevos. En la educación secundaria los estudiantes se enfrentan con distintos desafíos, entre ellos el de resolver problemas. La ciencia de la computación puede ser utilizada para enseñar matemáticas, siendo una de sus utilidades el programar soluciones a los problemas.

La motivación de utilizar MateFun, se encuentra en ser un lenguaje de sintaxis muy similar a la matemática, lo cual reduce la curva de aprendizaje y se presenta como opción para un primer acercamiento a la programación.

3.2. Antecedentes

El proyecto MateFun surge desde el Centro Interdisciplinario para la Cognición en la Enseñanza y el Aprendizaje[10].

MateFun es también un lenguaje de programación puramente funcional, interpretado y fuertemente tipado, desarrollado por investigadores del Instituto de Computación (InCo) y estudiantes de la Facultad de Ingeniería[10]. Fue diseñado para apoyar el aprendizaje matemático de estudiantes preuniversitarios, por lo cual su sintaxis es mínima y muy similar al lenguaje matemático[27]. Entre sus implementaciones se incluyen un IDE web[22] y versiones infantiles web[23] y Andriod[24].

MateFun permite dibujar figuras, crear animaciones y definir funciones programadas por el usuario. Los docentes pueden compartir archivos con el grupo y los estudiantes realizar actividades. Los docentes pueden visualizar las entregas que realiza cada alumno y calificarlas.

Se ha llegado a utilizar en instancias académicas, en particular, en el curso *Matemática y programación: potencial didáctico en el aula* y también en una intervención de cuatro semanas en el Colegio Nacional José Pedro Varela con estudiantes de segundo año de ciclo básico[20].

3.3. Objetivos

Este proyecto de grado tiene como objetivo desarrollar un entorno virtual de actividades de aprendizaje de matemática con la posibilidad de extenderse a otras ciencias, donde se pueda utilizar el lenguaje MateFun. Se busca poder aplicar los conocimientos adquiridos durante la carrera.

Este entorno debe permitir a docentes y a estudiantes compartir su trabajo, ya sean los enunciados de los problemas o sus soluciones. El sitio debe contar al menos con un menú para seleccionar temas, una interfaz para crear ejercicios (con algún editor de texto y la posibilidad de subir archivos multimedia) y enlazar con el sitio de MateFun para poder programar las soluciones.

Se pretende que sea un aporte a la enseñanza, ya sea dentro de un ámbito curricular, como por fuera del mismo, buscando que sea esta plataforma una herramienta útil y atractiva para docentes, estudiantes y usuarios en general.

3.4. Etapas del proyecto

Se presenta el cronograma del proyecto:

Entre **agosto** y **septiembre** de 2020 se realizó el contacto con los tutores, la definición del proyecto y su aprobación. Se comenzó con un estudio de herramientas similares.

Desde **octubre** a **noviembre**, se hizo el relevamiento de requerimientos, diseño de los requerimientos y funcionalidades. También se realizó el estudio de aspectos relacionados a MateFun. Luego de tener definidos los requerimientos se realizaron diseños de bocetos de la plataforma.

En **diciembre** se comenzó con el estudio de tecnologías y preparación previa, la definición de nuevas funcionalidades, la implementación y validaciones con tutores, lo que llevó hasta el mes de **marzo** 2021.

En **abril** se finalizó la primera versión de la plataforma, se comenzó con la documentación, se confeccionaron los manuales de uso, y se procedió a la preparación y diseño de pruebas de usuario, incluyendo la invitación a participación de docentes del Liceo IEP.

En el mes de **mayo** se continuó con la documentación y se realizaron pruebas con docentes del liceo mencionado, así como también se realizaron invitaciones a docentes del curso MateFun y posteriormente se comenzaron las pruebas con usuarios expertos, haciendo correcciones a la plataforma de forma iterativa.

En **junio** finalizaron las pruebas de usuario y se documentaron las pruebas, los cambios propuestos y posibles extensiones al sitio.

Entre **julio** y **agosto** se finalizó el proyecto, realizando las últimas modificaciones al sitio y cerrando la documentación, el informe, creando los vídeos tutoriales y armando la presentación.

Organización del documento

El presente documento se divide en capítulos, secciones y subsecciones. Se detalla a continuación un resumen por capítulo.

- **Introducción:** Se presenta la motivación, los antecedentes, los objetivos y la organización del documento.
- **Análisis de requerimientos:** Presenta un estudio de las plataformas similares estudiadas al comienzo del proyecto.
- **Evolución de la especificación:** Se expone la transformación de la especificación desde la idea inicial hasta la especificación final de la plataforma.
- **Diseño:** Se presenta la arquitectura, los bocetos de las interfaces y el modelo de dominio.
- **Implementación:** Se detallan las características del equipo de desarrollo, la forma de trabajo y se realiza una introducción al proceso de desarrollo realizado, así como las decisiones tomadas.
- **Diseño del sitio:** Se visualizan las interfaces y se explican las decisiones de diseño tomadas, que incluyen las resoluciones de pantalla que se tuvieron en cuenta para el diseño adaptable. También se menciona la influencia de las pruebas de usuario en el sitio.
- **Validación:** Muestra la planificación de las pruebas de usuario, sus iteraciones y los resultados.
- **Conclusiones.**
- **Futuras líneas de desarrollo:** Se plantean posibles extensiones y mejoras a Emmy, surgidas de las pruebas de usuario y de los desarrolladores.
- **Glosario:** Contiene definiciones de términos y siglas utilizados.
- **Apéndice:** Incluye los documento de: requisitos, pruebas, implementación, el manual de usuario y los bocetos del sitio.

Análisis de requerimientos

En este capítulo se trabaja sobre la etapa de análisis de requerimientos, y se listan las plataformas y herramientas estudiadas.

Los requerimientos surgieron a través del intercambio con los tutores del proyecto y consultas a docentes de enseñanza secundaria. Se consultó a docentes primero por medio de los tutores en las primeras etapas del relevamiento y se contactó con nueve docentes durante las pruebas de usuario.

También se consultaron el sitio de Matefun¹, sus proyectos de grado [20] [24] [23], documentos relacionados con la plataforma [27] [21], el sitio de cursos de MateFun² y los sitios relevados en la Sección 5.1.

5.1. Relevamiento

Con la definición de la idea inicial, se comenzó a relevar plataformas de enseñanza similares, entre ellas, el sitio de matemáticas *NRICH*, la plataforma de resolución de problemas clásicos de programación *LeetCode* y sitios dedicados a enseñar programación a través la resolución de problemas de matemática, *Project Euler* y problemas de bioinformática, *Rosalind*.

5.1.1. NRICH

El sitio NRICH³ (Figura 5.1) provee recursos para el aprendizaje matemático para docentes y estudiantes, desde la edad escolar a edades más avanzadas. No solo provee ejercicios y explicaciones detalladas, si no también experimentos matemáticos para realizar, aunque no maneja lenguajes de programación. Los estudiantes pueden enviarse sus soluciones las cuáles pueden terminar siendo destacadas en el sitio.

Esta plataforma clasifica los temas según el rango de edad y los presenta de manera ordenada y categórica.

¹<https://matefun.math.psyco.edu.uy/>

²<https://cursos-matefun.moodlecloud.com/login/index.php>

³<https://nrich.maths.org/>

Figura 5.1: NRICH: Ejemplo de experimento.

5.1.2. Project Euler

Proyecto Euler⁴ (Figura 5.2) es un sitio dedicado la resolución de problemas matemáticos y de ciencias de computación utilizando la programación. Cuenta con más de un millón de usuarios y más de setecientos desafíos, donde la dificultad se va incrementando y los conocimientos de un problema sirven de preparación para los siguientes.

A diferencia de NRICH, el sitio no enseña los conocimientos matemáticos requeridos para resolver cada problema (los cuales quedan a cargo del estudiante) ni acepta el ingreso de código de programación. Solamente permite ingresar la respuesta al problema. Las letras de los ejercicios tienden a ser cortas, y a contar con un mínimo de ejemplos.

5.1.3. Rosalind

El sitio web de Rosalind⁵ (Figura 5.3) es similar al Project Euler, e intenta enseñar conceptos de programación a través de la resolución de problemas de bioinformática. No presenta un editor de texto ni es capaz de compilar código. Al igual

⁴<https://projecteuler.net/>

⁵<http://rosalind.info/problems/locations/>

Project Euler .net Logged in as ggaston095
Mon, 3 Oct 2020, 14:50
Sign Out

[About](#)
[Archives](#)
[Recent](#)
[Progress](#)
[Account](#)
[News](#)
[Friends](#)
[Statistics](#)
Cache update: 25 minutes

Problems Archives

The problems archives table shows problems 1 to 718. If you would like to tackle the 10 most recently published problems then go to Recent problems.

Click the description/title of the problem to view details and submit your answer. If you have already solved the problem it will display both the problem and the correct answer that you have already entered. In addition, once you've solved a particular problem an icon will appear in this table that gives you access to a thread in the forum dedicated to that problem.

All (Solved / Unsolved / Pinned) problems are being shown, sorted by the ID column in **Ascending** order.

View the details of all problems on one page.
Other language translation projects: Project Euler Forum

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 Go to Problem:

ID	Description / Title	Solved By	Difficulty	
1	Multiples of 3 and 5	964765	1	
2	Even Fibonacci numbers	767915	1	
3	Largest prime factor	550417	1	
4	Largest palindrome product	486223	1	
5	Smallest multiple	489821	1	
6	Sum square difference	492846	1	
7	10001st prime	421148	1	
8	Largest product in a series	351797	1	

Figura 5.2: Project Euler: Sección de ejercicios.

que éste, las explicaciones se reducen a describir el problema y los ejemplos son escasos. Ninguno de estos dos sitios busca guiar al estudiante hacia una respuesta ni le enseña los conocimientos previos que debería tener.

Rosalind SALIND About - Problems - Statistics - Glossary search Log in Register

Translating RNA into Protein solved by 18806

July 2, 2012, midnight by Rosalind Team

The Genetic Code click to expand

Problem

The 20 commonly occurring amino acids are abbreviated by using 20 letters from the English alphabet (all letters except for B, J, O, U, X, and Z). Protein strings are constructed from these 20 symbols. Henceforth, the term **genetic string** will incorporate protein strings along with DNA strings and RNA strings.

The **RNA codon table** dictates the details regarding the encoding of specific codons into the amino acid alphabet.

Given: An RNA string *a* corresponding to a strand of mRNA (of length at most 10 kbp).

Return: The protein string encoded by *a*.

Sample Dataset

```
AUGGCCAUGGCGCCCAGAACUGAGAUCAAUUGUACCCGUAUUAACGGGUGUA
```

Sample Output

```
MAAPRTEINSTRING
```

Please login to solve this problem.

Figura 5.3: Rosalind: Problema de traducir ARN en proteínas.

5.1.4. LeetCode

El sitio web de *LeetCode*⁶ (Figura 5.4) está diseñado para enseñar programación, se presenta un problema y una serie de ejemplos que sirven de casos de prueba. A diferencia de *Project Euler*, tiene una interfaz de programación embebida capaz que ejecutar código. Presenta un listado de ejercicios disponibles, que se pueden reordenar por dificultad y porcentaje de resolución (Figura 5.5).

Existen además numerosos sitios similares para aprender programación a través de la resolución de problemas, como *CodingBat*⁷, *CoderByte*⁸ y *Exercism*⁹ entre otros.

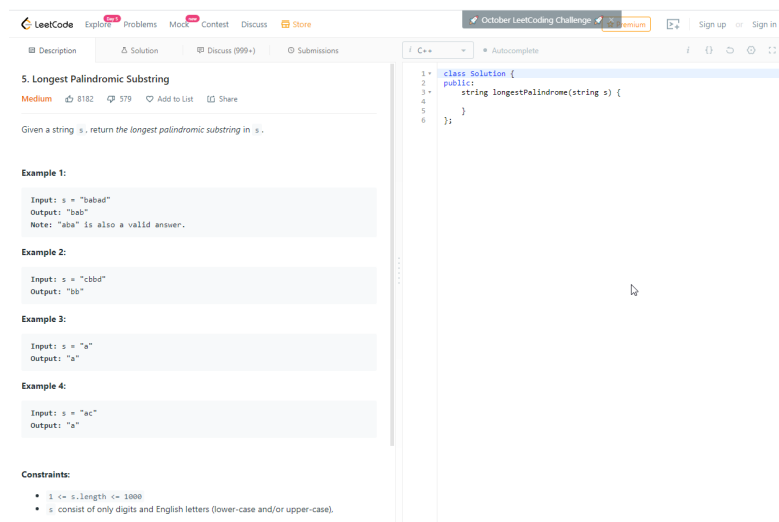


Figura 5.4: LeetCode: Resolución de un ejercicio, el problema a la izquierda y el IDE a la derecha.

5.1.5. Moodle

Se comparó la especificación inicial propuesta con una posible integración con la plataforma Moodle.

⁶<https://leetcode.com>

⁷<https://codingbat.com/java>

⁸<https://coderbyte.com/challenges>

⁹<https://exercism.io/tracks/ruby>
























#	Title	Solution	Acceptance	Difficulty	Frequency 
✓ 85	Maximal Rectangle		33.0%	Hard	
✓ 621	Task Scheduler		45.2%	Medium	
✓ 412	Fizz Buzz		59.3%	Easy	
✓ 41	First Missing Positive		28.6%	Hard	
✓ 54	Spiral Matrix		30.2%	Medium	
642	Design Search Autocomplete System 		37.1%	Hard	
✓ 344	Reverse String		63.1%	Easy	
✓ 127	Word Ladder		23.7%	Medium	
124	Binary Tree Maximum Path Sum		29.8%	Hard	
✓ 336	Palindrome Pairs		30.7%	Hard	
✓ 141	Linked List Cycle		36.5%	Easy	
193	Valid Phone Numbers		25.1%	Easy	

Figura 5.5: Listado de ejercicios de LeetCode.

Manejo de usuarios

Tanto Moodle como Matefun proveen un sistema de manejo de usuarios, permiten crear usuario por parte de un administrador, iniciar y cerrar sesión.

Asignación, entrega y calificación de tareas

En ambas plataformas se pueden realizar tareas, Matefun además provee de un editor de código en donde el alumno puede realizar la tarea. En la entrega de la tarea ambas plataformas son muy similares, el docente puede ver las entregas realizadas por los estudiantes así como calificar las mismas.

Por otro lado, Matefun no permite a los docentes subir enunciados de ejercicios ni asignarles las tareas a los estudiantes, debiendo comunicarlo por otro medio previamente.

Archivos y directorios

Matefun permite almacenar las soluciones a los ejercicios en archivos y estos en carpetas, siempre que estos se hayan generado dentro de la plataforma. Moodle expande esto permitiendo subir archivos externos.

Herramientas en Moodle

Se estudiaron como distintas herramientas disponibles en Moodle se utilizan para enseñar programación, y cómo estas mezclan la presentación de ejercicios con la resolución.

Virtual Programming Lab for Moodle

Virtual Programming Lab for Moodle¹⁰ (Figura 5.6) se muestra un plugin que permite trabajar con tareas de programación. Soporta calificaciones, grupos y roles. Permite subir archivos y ejecuta código fuente en varios lenguajes[16][17].

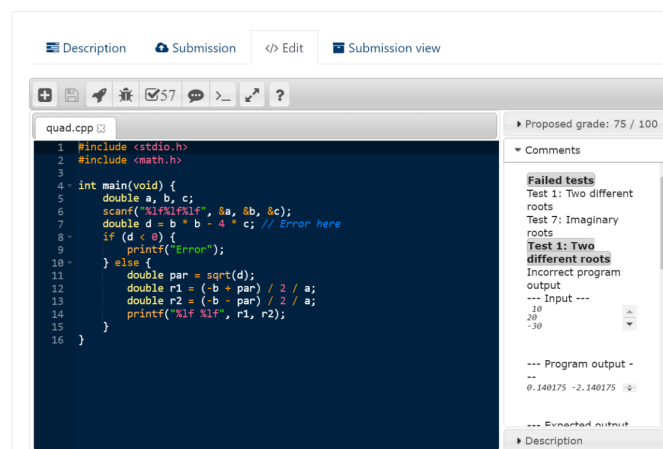


Figura 5.6: Editor de VPL.

Question types

Question types CodeRunner¹¹ (Figura 5.7) es un plugin de preguntas y respuesta de código abierto. Permite escribir código fuente y ejecutarlo, comparando las salidas con las respuestas esperadas[5].

¹⁰<https://vpl.dis.ulpgc.es/>

¹¹https://docs.moodle.org/all/es/Tipo_de_pregunta_CodeRunner

Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Write a Python3 function `sqr(n)` that returns the square of its numeric parameter `n`.

For example:

Test	Result
<code>print (sqr (-3))</code>	9
<code>print (sqr (11))</code>	121

Answer:

```

1 def sqr(n):
2     return n * n

```

Check

Test	Expected	Got	
<code>print (sqr (-3))</code>	9	9	✓
<code>print (sqr (11))</code>	121	121	✓
<code>print (sqr (-4))</code>	16	16	✓
<code>print (sqr (0))</code>	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Figura 5.7: Ejemplo de CodeRunner

5.2. Conclusiones del relevamiento

Project Euler, Rosalind y LeetCode presentan problemas que solo pueden ser resueltos utilizando programación, por lo que no es posible llegar a una respuesta sin realizar una implementación, mientras que la plataforma propuesta busca presentar ejercicios matemáticos que no necesariamente se resuelvan enteramente programando. Por otro lado NRICH no está pensando para ser utilizado junto con lenguajes de programación, ya que presenta ejercicios de otra índole, donde la respuesta surge más del ingenio y del conocimiento.

En las plataformas de Project Euler, Rosalind y Leetcode las respuestas no son supervisadas ni revisadas, son solamente corregidas de manera automática, en el caso de Project Euler, la respuesta consiste en escribir en un campo de texto plano el resultado. En el caso de NRICH, este permite enviar las respuestas las cuales pueden llegar a ser publicadas en el sitio por parte de los administradores.

De los tres sitios de programación ya mencionados, solo LeetCode trae consigo un entorno de programación para la resolución en el sitio (ya que su principal objetivo es enseñar a programar) mientras que los demás requieren el uso de editores externos y no imponen un lenguaje de programación.

Por otro lado, tanto CodeRunner como VPL de Moodle presentan también edi-

tores de código, siendo VPL el que mejor lo hace (Figura 5.6), dando comentarios acerca del código ingresado.

La plataforma Moodle presenta varias características con la definición inicial, ya que permite tener docentes y estudiantes registrados, publicar tareas, corregirlas y asignarles una nota, donde el trabajo con el entorno MateFun se debería hacer de manera externa.

En cuanto a la estructuración de contenido, NRICH es el sitio más estructurado (de los relevados), dividiendo su contenido en categorías, temas y grados, mientras que Project Euler opta por presentar el contenido como una lista de ejercicios de diversas temáticas y dificultades, sin orden aparente.

Evolución de la especificación

Este capítulo se centra en la especificación de la plataforma y cómo evolucionó desde su concepción inicial hasta su especificación final.

La tarea de la definición de requerimientos constituye la primera actividad del proceso de ingeniería de software [26]. Para esta, se aplicaron los conocimientos teóricos adquiridos en la asignatura Introducción a la Ingeniería de Software y la experiencia práctica que se obtuvo con la asignatura Proyecto de Ingeniería de Software ¹.

La especificación de requisitos se realizó a través de historias de usuario (Anexo A), buscando definir un mínimo producto viable MVP.

A partir del estudio de la idea inicial y de relevar sitios similares, se comenzó a intercambiar con los tutores acerca de la idea de la plataforma y el alcance del MVP. Se realizaron bocetos de baja fidelidad en papel, para luego crear bocetos interactivos (Sección 7.3), que fueron compartidos con los tutores, para estudiar posibles flujos de uso en la plataforma y validar el diseño.

Las pruebas con usuarios también ayudaron a validar y corregir errores luego de implementada la plataforma.

Los requerimientos se fueron refinando a lo largo de proyecto, así como también se fueron agregando otros, muchos surgidos por la interacción con la plataforma, la retroalimentación de los tutores y las pruebas de usuario.

6.1. Definición inicial y su evolución

La idea inicial del proyecto comenzó con desarrollar un entorno virtual de actividades de aprendizaje de matemática donde se pudiera utilizar el lenguaje MateFun. El alcance de la plataforma se expandió luego para abarcar no solo ejercicios de matemática si no de ciencias en general.

En la plataforma se pretendía que los profesores tuviesen acceso a ejercicios y a su vez, pudieran agregar los suyos, mientras que a los estudiantes se les presentara la posibilidad de resolverlos.

La plataforma se planteaba como abierta, ya que los estudiantes no necesaria-

¹cuya tarea involucró la definición y validación de requisitos de un producto

mente debían ser alumnos de los docentes que publicaban los ejercicios, pudiendo utilizar la plataforma para estudiar de forma independiente. A medida que se fue progresando en la plataforma, se agregaron funcionalidades para ser utilizadas en el aula, estas son identificar a la solución mediante un código y poder continuar con la misma, a través del envío de borradores intermedios.

Se debía permitir seleccionar temas y ejercicios, contar con una interfase donde los profesores pudieran subir sus propios enunciados de ejercicios de matemática (con algún editor adecuado y con la posibilidad de agregar figuras o vídeos). El sitio debía contar con un enlace externo al intérprete de MateFun que permitiera programar las soluciones (quedando almacenados en el sitio).

La idea original era una plataforma dedicada a la publicación y resolución de ejercicios, donde la interacción con MateFun se diera en un sitio externo, no involucraba la integración con el entorno de MateFun y su intérprete. Con el avance del proyecto, se decidió integrar el entorno en la plataforma, donde los autores podían incluirlo en sus publicaciones.

La plataforma lleva el nombre Emmy en honor a Emmy Noether, matemática alemana que realizó grandes contribuciones a la física teórica y álgebra abstracta. La idea surge de utilizar el nombre de una científica no tan reconocida. Por otro lado, Emmy también significa: *Ejercicios de Matemáticas en MateFun Ya!*

6.2. Especificación

Emmy es una plataforma web de acceso libre para el aprendizaje de matemática que provee un entorno de actividades de aprendizaje, pudiendo utilizar para ello el lenguaje MateFun.

Los usuarios son, docentes de matemática y/o ciencias, los cuales pueden registrarse, estudiantes y administradores.

Emmy da la posibilidad de acceder a los contenidos de manera jerárquica y estructurada, pudiendo acceder a través de los distintos temas, utilizando filtros como dificultad o grado, teniendo la posibilidad de seleccionar ejercicios o buscar por el autor de estos.

Los docentes serán capaces de redactar ejercicios, teniendo distintos estilos de texto, con la posibilidad de embeber imágenes y vídeos, espacios para completar y campos de múltiple opción, con la posibilidad de dar corrección automática,

pudiendo escribir programas que se puedan cargar en el intérprete.

Estos tendrán un área dedicada donde se exhibirán los ejercicios que han redactado, los cuáles se pueden clasificar por nivel de dificultad, grado y/o por tema. Mientras que los estudiantes podrán buscar ejercicios filtrando por docente, por grado, por tema, por dificultad o por palabra clave.

Para los estudiantes, la funcionalidad principal estará en seleccionar y resolver ejercicios pudiendo utilizar el intérprete de MateFun directamente. Pueden visualizar gráficos 2D, 3D y animaciones.

Los docentes podrán realizar un seguimiento a las tareas entregadas por los estudiantes, estos pueden enviar su solución a los profesores identificándolas por un código privado, siendo este coordinado previamente con el docente.

Se permite resolver el ejercicio y enviarlo o guardarlo como borrador. Para volver a acceder a este, ya sea para visualizarlo o continuarlo se utiliza un código secreto, el cual es único en el sistema.

Los docentes acceden tanto a las soluciones entregadas como las soluciones que están en borrador, pudiendo los estudiantes ver la corrección automática si el docente así lo desea.

Los administradores podrán crear y eliminar usuarios así como restaurar sus contraseñas. También podrán crear, eliminar temas y eliminar publicaciones. Tendrán una visión sobre cuáles son los temas más populares, la cantidad de publicaciones según su dificultad, la cantidad de respuestas y publicaciones en orden cronológico.

Diseño

Este capítulo muestra las decisiones de diseño tomadas para el desarrollo de la plataforma. Incluye la definición de la arquitectura, de la base de datos y los bocetos de interfaces de usuario.

7.1. Modelo de dominio

Basándose en la descripción de la realidad (descrita en Sección 6.2), se diseñó un modelo de dominio (como muestra la Figura 7.1. Este presenta dos tipos de usuarios; docentes y administradores. Se le asoció a las publicaciones un grado, un tema y una dificultad. Un grado puede ser un subgrado de otro, por ejemplo: *cuarto*, *quinto* y *sexto de bachillerato* son subgrados del grado *Bachillerato*. El valor de dificultad y grado sirven para ordenar. Existe un único tema por defecto, el cual se asigna a todas las publicaciones de un tema cuando se elimina.

Restricciones

- Un usuario no puede ser docente y administrador al mismo tiempo.
- Un grado no puede ser subgrado de si mismo.
- El campo *valor* de grado y dificultad es único para cada uno.
- Solo puede haber un tema por defecto.
- El campo *clave de recuperación de respuesta* es una cadena alfanumérica de seis caracteres y es única.

7.2. Arquitectura

La arquitectura de la plataforma (Figura 7.2) consiste en el modelo cliente servidor. Los usuarios accederán a ella a través de un navegador web. En la capa de negocios se implementa la lógica y la conexión con los servidores de MateFun, mientras que la capa de persistencia se encarga del almacenamiento¹.

¹Aquí también se almacena el código MateFun

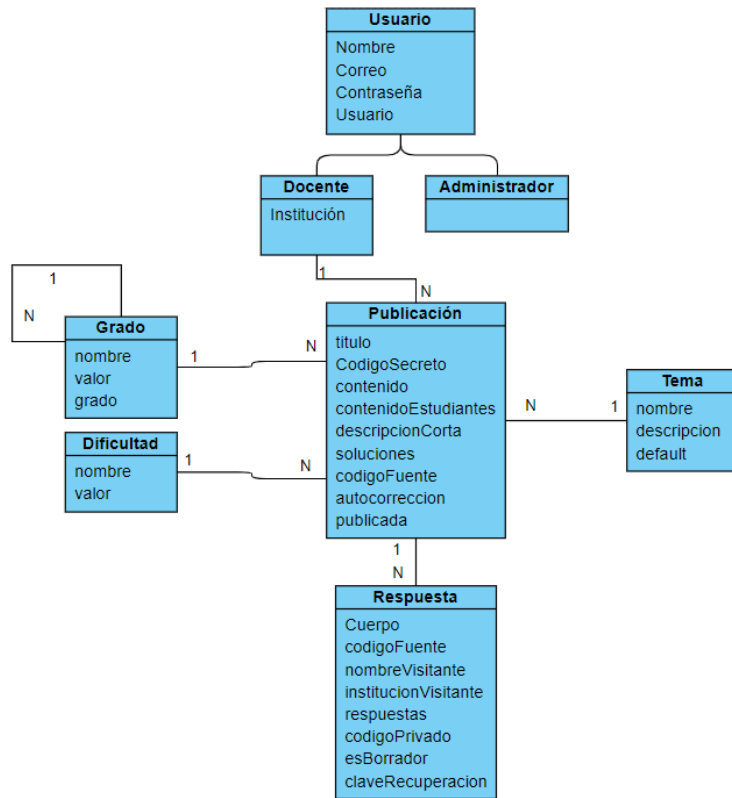


Figura 7.1: Modelo de dominio.

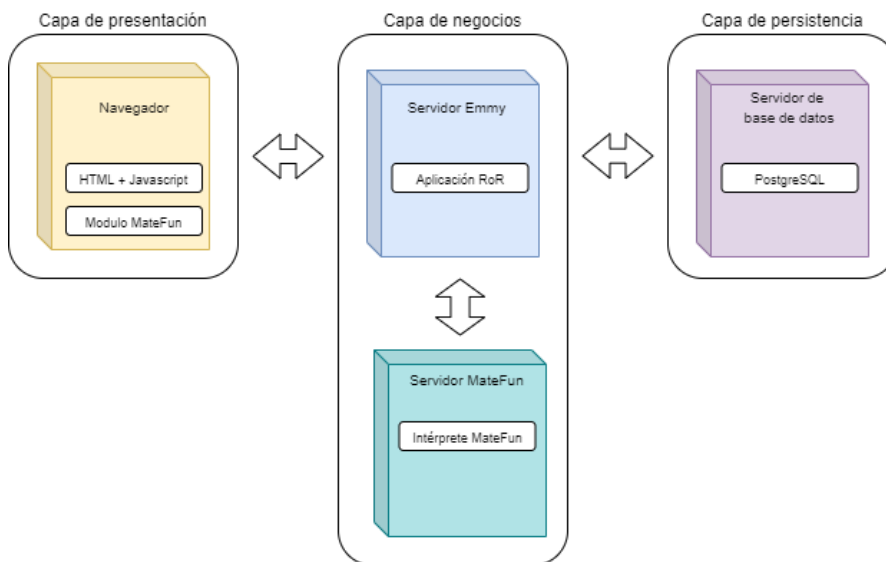


Figura 7.2: Diagrama de arquitectura.

7.3. Interfaces de usuario

Se realizaron diseños rápidos de baja fidelidad intentando abarcar las pantallas más importantes del sitio, para ello se crearon pantallas interactivas que permitían navegar a través de ellas utilizando un navegador web, para validar flujos, requerimientos y diseño. Estas pantallas fueron presentadas a los tutores para validación y fueron posteriormente la base de la primera implementación del sitio. En el Anexo B se muestran las pantallas con mayor resolución. También se encuentran disponibles para descargar ².

7.3.1. Página principal

Comenzando con la página principal, la Figura 7.3 muestra la página de bienvenida del sitio, se basó en el sitio NRICH mostrando una navegación a través de tarjetas. Se tiene una barra de navegación superior con un buscador genérico por ejercicio, usuario y tema.

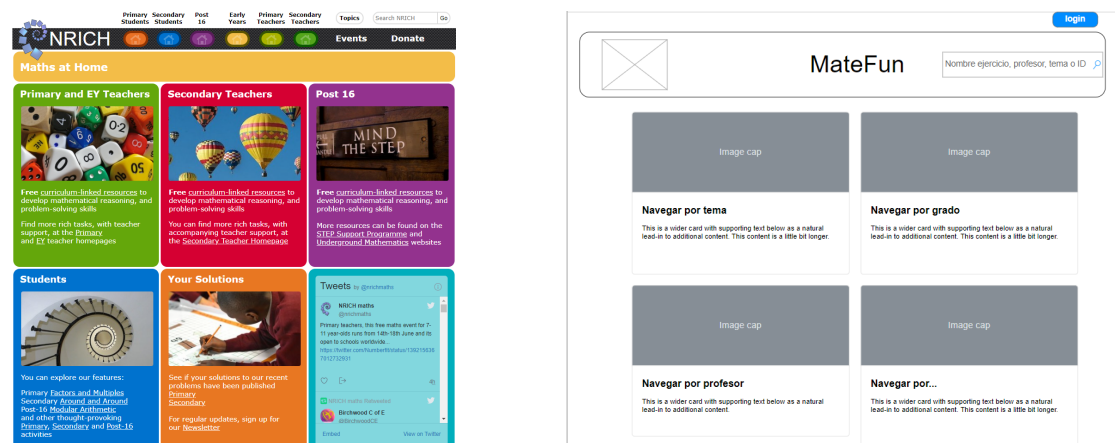


Figura 7.3: Página principal, (*NRICH -izquierda- y boceto -derecha-*).

7.3.2. Navegación por tema, grado o docente

Para la navegación se continuó utilizando un estilo de tarjetas, también basado en la navegación de NRICH. Figura 7.4 muestra la navegación por tema.

²[urlhttps://gitlab.fing.edu.uy/matefun/emmy/-/wikis/Prototipos-interactivos](https://gitlab.fing.edu.uy/matefun/emmy/-/wikis/Prototipos-interactivos)

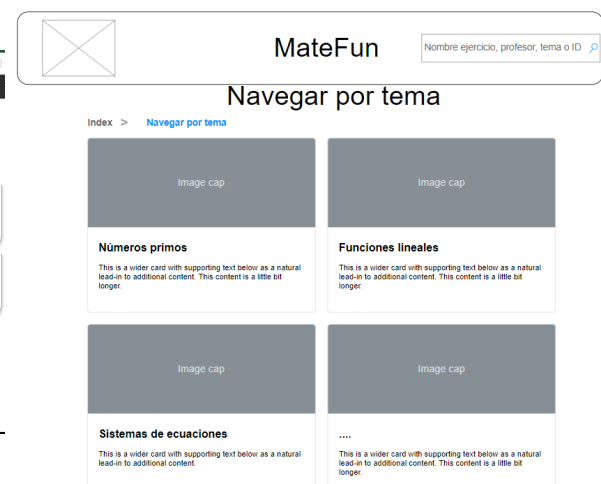
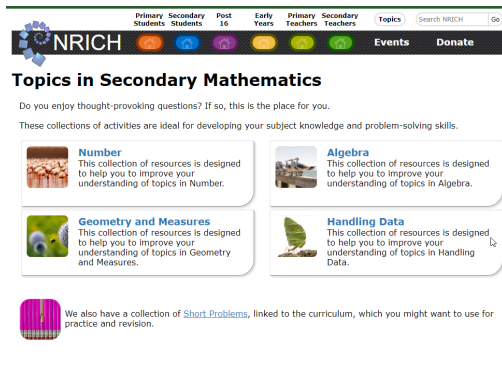


Figura 7.4: Navegación por tema, (*NRICH -izquierda- y boceto -derecha*).

7.3.3. Listar publicaciones

El listado de publicaciones (Figura 7.5) del sitio se ideó con un diseño más simple, basado en NRICH.

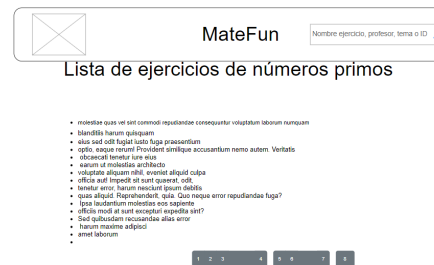
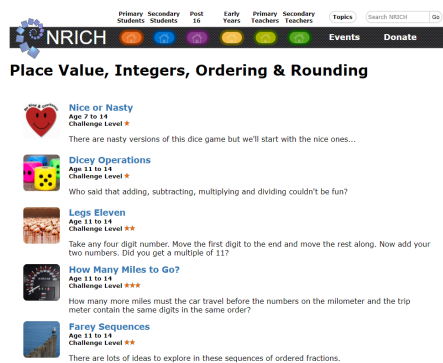


Figura 7.5: NRICH (izquierda) y boceto (derecha).

7.3.4. Perfil docente

Para el perfil docente (Figura 7.6) se dividió en dos paneles principales, a la izquierda, se listan sus publicaciones, la opción de ver respuestas y de editar, mientras que a la derecha un listado de opciones relacionadas al perfil.



Figura 7.6: Perfil docente.

7.3.5. Crear publicación

Para crear publicaciones (Figura 7.7) se pensó en un editor simple que tuviera las opciones más comunes al alcance y permitiera la posibilidad de agregar campos multimedia como imágenes y vídeos, así como opciones para ingresar campos de interactivos de completar y de múltiple opción.

También se consideró la posibilidad de embeber el intérprete, así como la posibilidad de guardar la publicación como borrador.

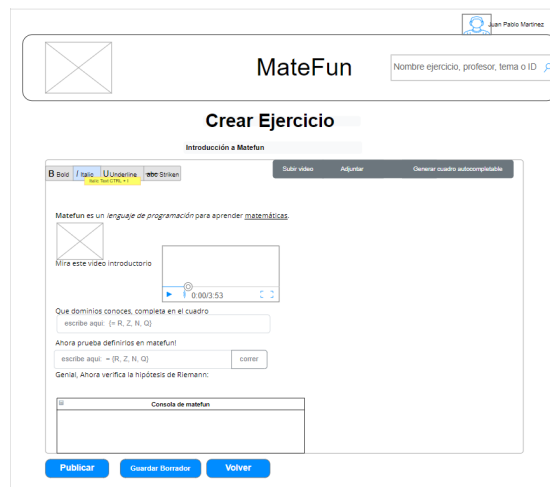


Figura 7.7: Crear publicación.

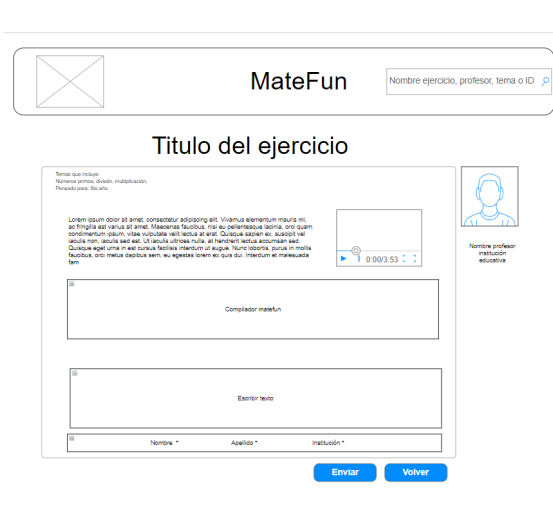


Figura 7.8: Responder publicación.

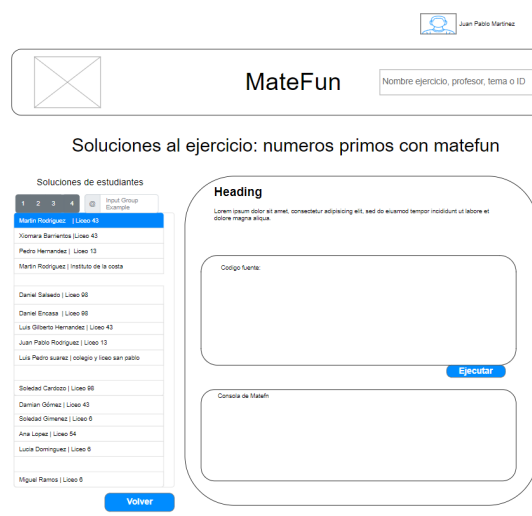


Figura 7.9: Listar y ver soluciones.

7.3.6. Responder publicación

La pantalla de responder publicaciones (Figura 7.8) se ideó en dos paneles, uno principal de mayor dimensión que contenga la publicación y el área para responder. Inspirándose en CodeRunner y LeetCode, (figuras 5.7 y 5.1.4) se pensó en embeber el intérprete junto con el área de respuesta.

Este boceto ya incluía campos para escribir texto y para utilizar el editor de texto enriquecido, por debajo, las opciones para identificarse.

En el cuadro de la derecha se pretendió mostrar la imagen y la descripción de la publicación, para que se asociara con el autor y dándole una sensación de publicación de blog.

7.3.7. Listar y ver soluciones

Para el listado y visualización de soluciones (Figura 7.9) se modeló listar a la izquierda los nombres de los estudiantes y su institución, junto con un filtro por subcadena, y a la derecha el contenido de la respuesta, todo en un diseño compacto que permitiera cambiar rápidamente de solución.

Implementación

Este capítulo detalla brevemente la implementación del sitio, la cual incluye desde las tecnologías hasta el diseño de las interfaces de usuario. Se presenta por motivos de simplicidad una leve introducción. En el Anexo E, se encuentra el documento de implementación del cual se pueden obtener información más detallada acerca del proceso de desarrollo y las características implementadas en la plataforma.

8.1. Equipo de desarrollo

Los conocimientos del equipo de desarrollo determinaron la elección de tecnologías. El equipo tenía conocimientos básicos de Ruby on Rails adquiridos en Proyecto de Ingeniería de Software, donde se implementó un Backend, específicamente, una API para una aplicación móvil.

No se tenía conocimientos sobre programación Frontend aunque se conocía los fundamentos de funcionamiento de JavaScript. En Programación 4 se había trabajado levemente Bootstrap y Java (Web services), en Fundamentos de Base de Datos se llegó a utilizar PostgreSQL, además uno de los miembros había cursado Interacción Persona Computadora, lo que facilitó el diseño de las interfaces. No se tenía experiencia laboral en desarrollo¹.

La metodología de desarrollo utilizada fue basada en prácticas Ágiles, haciendo el desarrollo de manera incremental e iterativo. Se partió de historias de usuarios en lenguaje natural, las cuales pasaron a ser *incidencias* de GitLab (Figura 8.1).

Las historias de usuario definieron el desarrollo de Emmy de manera incremental, definiendo *sprints* de duración variable (entre dos y tres semanas), intercalados con reuniones con los tutores para mostrar y validar el trabajo realizado, así como definir y priorizar las siguientes historias de usuario a implementar.

Se realizaron pruebas de integración desde el Frontend manuales durante todo el desarrollo del proyecto y se definieron pruebas unitarias y de controlador automatizadas sobre el final del desarrollo.

¹Hasta haber terminado prácticamente la etapa de implementación.










Filtrar soluciones por código de estudiante #44 · opened 3 months ago by Veronica Dahiana Bentancor Cedrez Docente Historia	CLOSED  0 updated just now
Cambiar contraseña de otro usuario siendo administrador #106 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 1 week ago
Listar administradores #105 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar publicaciones en perfil administrador #104 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar docentes en panel de administracion #103 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Ordenar docentes por substring en consola de administracion #81 · opened 2 months ago by Gaston Daniel Barreto Sugliani Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar temas en perfil administrador #102 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar respuestas de publicación #101 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Docente Historia	CLOSED  0 updated 2 weeks ago
Quitar filtro autor en perfil docente #100 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Corección	CLOSED  0 updated 2 weeks ago

Figura 8.1: Historias de usuario escritas como *incidencias* de GitLab

8.2. Ruby on Rails

Se utilizó el Framework de código abierto de Ruby on Rails, el cual utiliza el lenguaje Ruby, un lenguaje enfocado al desarrollador de sintaxis clara y simple. Se hizo esta elección como Backend de Emmy por haber trabajado con esta tecnología en Proyecto de Ingeniería de Software, además de ser un framework web.

8.3. Base de datos

Se utilizó PostgreSQL como motor de base de datos para el ambiente de desarrollo y de producción (el cual se utilizó en la asignatura Base de Datos), mientras que para tener un ambiente más ligero, se utilizó SQLite en el ambiente de pruebas.

8.4. Reutilización de código y refactorización

Se siguió la filosofía de desarrollo de Ruby on Rails de convención sobre configuración, por lo cual se optó por seguir todas las convenciones de desarrollo del

Framework, esto simplifica su configuración, mantenimiento y reduce los puntos de falla posibles, además permite a cualquier desarrollador experimentado leer, entender y continuar con el desarrollo de la plataforma de manera relativamente fácil.

La otra filosofía respetada fue DRY (Don't repeat yourself), implicó reutilizar la mayor cantidad de código posible, implementando funciones paramétricas que se adapten a varios requerimientos, (en vez de replicar código muy similar), esto ayuda a reducir errores, mantener el código limpio y mantenible, así como reducir el tamaño de métodos, clases y sus tests correspondientes.

Esta filosofía también contribuye a las decisiones de diseño del código, ya que implica por ejemplo seguir el patrón de *Fat model*, *Skinny Controller*, por lo cual toda la lógica de negocios se implementó en los modelos, mientras que los controladores se dejaron como nexo entre estos y las vistas.

Al finalizar cada funcionalidad, se le hizo una refactorización al código.

8.5. Arquitectura MVC

Modelos

Siguiendo la arquitectura MVC, los modelos contienen la lógica de negocio.

Difficulty: Modela la dificultad de los ejercicios. El orden por defecto es por nivel de dificultad, por lo cual se ordena por valor ascendente.

Grade: Modela los grados para los cuales son pensados las publicaciones. El orden por defecto es por año académico, por lo cual se ordena por valor ascendente. Las dificultades y los grados se encuentran definidos por código y solo los desarrolladores pueden modificarlos. Para hacerlo, existe un archivo de *semillas* fácilmente editable.

Topic: Modela los temas de las publicaciones, cada una puede tener asociada solo un tema. Se encuentran precargados más de cien temas, que fueron extraídos de los temarios de enseñanza primaria, secundaria y terciaria del Uruguay. Los administradores pueden agregar, quitar y modificar temas desde el menú de administración, y los desarrolladores pueden modificar el archivo de semillas para modificar los temas precargados.

Cuando se borra un tema, se le asigna automáticamente, mediante un evento

de disparo a todas sus publicaciones relacionadas el tema por defecto. El tema por defecto no puede ser modificado por los administradores, y está implementado a nivel de código también en el archivo de semillas.

Usuario: Implementado con la gema Devise. Los usuarios pueden ser docentes o administradores, esta herencia se implementó mediante STI de Rails, por lo cual todos los tipos de usuarios comparten la misma tabla de la base de datos y se diferencian por un atributo identificador.

Esto se debe a que la única diferencia a nivel de atributos de base de datos entre docentes y administradores es que los primeros pueden tener asociada una institución. Los usuarios pueden tener muchas publicaciones asociadas y por defecto se listan de a doce. Al borrar un usuario se borran de manera automática todas sus publicaciones.

Publicación: Las publicaciones tienen una dificultad, un tema, un grado y un autor, así como muchas respuestas. Borrar una publicación borra automáticamente todas sus respuestas.

Cuenta con opciones booleanas que modifican las posibilidades de respuesta, por defecto todas valen verdadero.

Aquellas publicaciones que no acepten respuestas, no podrán tener habilitada la corrección automática para estudiantes, mientras que si no aceptan el uso del intérprete, no podrán habilitar el editor de código MateFun. Estas validaciones se realizan tanto a nivel de vista como de modelo.

Si una publicación permite el envío de respuestas y luego lo deshabilita, todas las respuestas ya enviadas podrán seguir siendo accedidas tanto por los estudiantes como por los docentes. Por otro lado, si se deshabilita por ejemplo el uso del intérprete o el ingreso de texto en las respuestas, este ya no será visible, aunque su contenido se sigue almacenando en la base de datos y estará disponible la próxima vez que se habilite la opción.

Respuestas: Las respuestas son hechas por visitantes y pertenecen a una publicación. Aunque no se asocia la respuesta a usuarios registrados, se dejó la asociación hecha para poder implementar las funcionalidades, sin tener que modificar la arquitectura ni la base de datos, alcanza con modificar el controlador de respuestas. Se listan por defecto de a diez.

Datos precargados

Se definieron archivos de semillas, que son archivos diseñados para cargar datos por defecto en la base de datos, usando la sintaxis propia de Ruby on Rails. Al tener configurados datos por defecto en un ambiente de desarrollo, se facilitan las pruebas al sitio, ya que este comienza con un juego de datos completos. Se crearon publicaciones y respuestas de largo variable, así como distintas combinaciones de respuestas a campos interactivos.

Controladores

La arquitectura de la aplicación sigue el patrón de diseño MVC. Por lo tanto, las funciones de los controladores están ligadas a los modelos, y se mantuvo la filosofía de Rails de Fat model, Skinny Controller.

Contienen también toda la lógica esperable de un controlador, como la redirección de peticiones, filtro y manejo de parámetros y peticiones http, seguridad de acceso y validación. Los controladores también proveen *helpers*, que contienen lógica auxiliar para las vistas.

8.6. Pruebas de código

Durante todo el desarrollo del proyecto, desde sus primeras concepciones, se realizaron tests de integración manuales desde el Frontend. Sobre la última etapa de desarrollo, luego de haber tenido un MVP y antes de presentar una versión a producción, se implementaron las pruebas unitarias y de controlador, esto se dejó para la última etapa ya que se buscaba llegar rápido al MVP para su validación. Los tests de integración y las pruebas de Frontend (que incluyen HTML y JavaScript) se realizaron manualmente para las funcionalidades que se agregaron en cada Merge-request, con revisiones a fondo de todas las funcionalidades con cada liberación.

Se midió la cobertura de las pruebas utilizando la gema *simplecov* logrando una cobertura del 100%².

En Figura 8.2 se muestra la cobertura de código archivo por archivo, donde se tiene un promedio mayor a tres ejecuciones por línea.

²En controladores y modelos desarrollados

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/admin_controller.rb	100.00 %	76	37	37	0	1.35
app/controllers/answers_controller.rb	100.00 %	308	133	133	0	3.07
app/controllers/application_controller.rb	100.00 %	31	10	10	0	6.20
app/controllers/concerns/publications_controller_concern.rb	100.00 %	94	31	31	0	3.97
app/controllers/grades_controller.rb	100.00 %	14	6	6	0	1.17
app/controllers/pages_controller.rb	100.00 %	8	3	3	0	1.00
app/controllers/publications_controller.rb	100.00 %	256	104	104	0	2.94
app/controllers/redirect_controller.rb	100.00 %	23	11	11	0	1.64
app/controllers/topics_controller.rb	100.00 %	105	58	58	0	1.74
app/controllers/users_controller.rb	100.00 %	127	63	63	0	2.73
app/helpers/application_helper.rb	100.00 %	12	7	7	0	23.86
app/models/admin.rb	100.00 %	5	1	1	0	1.00
app/models/answer.rb	100.00 %	58	31	31	0	26.52
app/models/difficulty.rb	100.00 %	13	5	5	0	15.40
app/models/grade.rb	100.00 %	37	16	16	0	14.00
app/models/publication.rb	100.00 %	170	65	65	0	13.77
app/models/teacher.rb	100.00 %	5	1	1	0	1.00
app/models/topic.rb	100.00 %	28	13	13	0	2.31
test/controllers/answers_controller_test.rb	100.00 %	256	164	164	0	1.56
test/controllers/grades_controller_test.rb	100.00 %	21	11	11	0	1.09
test/controllers/pages_controller_test.rb	100.00 %	22	12	12	0	1.00
test/controllers/publications_controller_test.rb	100.00 %	320	167	167	0	1.87
test/controllers/redirect_controller_test.rb	100.00 %	25	17	17	0	1.00
test/controllers/topics_controller_test.rb	100.00 %	126	79	79	0	1.27
test/controllers/users_controller_test.rb	100.00 %	114	67	67	0	1.45
test/models/answer_test.rb	100.00 %	70	45	45	0	1.47
test/models/difficulty_test.rb	100.00 %	22	14	14	0	1.00
test/models/grade_test.rb	100.00 %	53	32	32	0	1.38
test/models/publication_test.rb	100.00 %	252	88	88	0	1.52
test/models/topic_test.rb	100.00 %	40	27	27	0	1.15
test/models/user_test.rb	100.00 %	56	42	42	0	1.14

Figura 8.2: Cobertura de código en modelos y controladores

Pruebas unitarias

Para las pruebas unitarias de modelos se separaron las pruebas en contextos específicos para las asociaciones, las validaciones de campos, los listados y los métodos, de esta forma, se organiza mejor el código.

Para cada método a validar, se crearon casos de prueba a partir del estudio de clases de equivalencia.

Pruebas de controladores

Para las pruebas de controlador se escribieron pruebas de rutas, se escribieron contextos basados en los perfiles administrador, usuario y visitante. En cada uno de estos contextos se escribieron pruebas que validaran las funcionalidades que estos tipos de usuarios deberían poder realizar y también se validó que no pudieran realizar acciones que no les correspondían. Para cada una de las acciones a validar que si se podían llevar a cabo, se probaron clases de equivalencia para asegurarse que estaban cubiertos todos los casos.

Calidad de código

Para controlar la calidad de código se utilizó la gema *Rubocop*³, la cual es un analizador de código estático. Se utilizó la configuración por defecto, que utiliza muchas de las reglas definidas en la guía de estilos de Ruby, y se configuraron excepciones en algunos métodos donde seguirlas, dificultaría la lectura o cohesión, por ejemplo, se desactivó el control de largo máximo de método o clase en los test, ya que implicarían dividirlo en muchos archivos, perdiendo cohesión.

8.7. Búsqueda de publicaciones

La búsqueda de publicaciones en Emmy hace uso de las funcionalidades de búsqueda de texto completo de PostgreSQL y el uso de ts vector. Este tipo de datos está relacionado con los lexemas, a lo cual se lo configuró para usar el lenguaje español como base.

³<https://github.com/rubocop/rubocop>

Con esta configuración, al realizar la búsqueda las palabras se llevan a su raíz lingüística, los plurales y las conjugaciones se transforman en su palabra raíz, mientras que palabras muy utilizadas, como por ejemplo *a*, *y*, *o*, son ignoradas, para evitar falsos positivos.

Así por ejemplo, si una persona busca “*hallé los números primos*”, van a aparecer en los resultados publicaciones que tengan esas palabras en cualquier forma de conjugación, por ejemplo “*hallar el número primo*”. Esto da una gran versatilidad y poder a las búsquedas de publicaciones, al mismo que las penalizaciones de rendimiento son mucho menores a implementaciones más simples, como las alcanzables con clausuras *LIKE/ILIKE*.

Se implementó además una búsqueda ponderada por atributo. Al momento de realizar una consulta, el mayor peso lo tienen las ocurrencias en el título, seguido por las ocurrencias de la descripción corta, y por último el contenido. Esto se hace así porque una ocurrencia de la palabra *primos* en el título, es un mejor indicador del contenido de la publicación que la misma ocurrencia en el cuerpo del mismo.



Figura 8.3: Búsqueda de números primos

La Figura 8.3 muestra la búsqueda *hallaron los numeros PRIMOS*⁴ en el sitio, por lo cual, la implementación encuentra la raíz lingüística de *hallaron* (*hallar*), transforma los plurales en singulares e ignora las mayúsculas, realizando la búsqueda por “*hallar número primo*”. Encuentra tres publicaciones, las cuales pondera su peso según dónde aparece la ocurrencia, las cuales son, en el título, en la descripción corta y en el cuerpo respectivamente.

Si esta misma búsqueda se hubiera implementado con soluciones más simples,

⁴Observar la falta del tilde y la mayúscula.

como las de *LIKE/ILIKE* ya mencionadas, no se hubieran obtenido resultados, ya que la conjugación de *hallar*, el plural de las palabras, el tilde de *números* y las mayúsculas de *PRIMOS* hubieran sido incompatibles con los campos de la publicación. Debe aclararse además, que los mismos resultados se hubieran obtenido para cualquier permutación de las palabras buscadas, tanto en la publicación como en el texto de búsqueda, algo imposible de hacer con las alternativas (ya que buscan subcadenas). Además de todo esto, tampoco se hubiera ponderado el resultado y el costo de ejecución hubiera sido mayor.

En conclusión, se implementó una búsqueda de texto completo que tiene en cuenta las conjugaciones del lenguaje, que simplifica las búsquedas por parte de los usuarios, evita falsos positivos, provee unos resultados ricos y ponderados según donde aparezcan las palabras consultadas, al mismo tiempo que ofrece grandes ventajas de rendimientos, gracias no solo a las capacidades de búsqueda en sí del motor de base de datos y a la implementación de memoria cache de esta. Si a esto se le agrega la paginación que implementa la gema *Will-paginate*, la cual solo extrae la base de datos la cantidad de elementos a mostrar en pantalla[19], se tiene una búsqueda muy completa, inteligente, rica y de muy bajo consumo frente a otras alternativas más simples.

8.8. Integridad de datos, validación y seguridad

Formularios

Para proteger contra ataques indeseados, cuando un usuario está frente a un formulario HTML, por ejemplo al crear una publicación o responder a una, Rails crea un token de autenticidad aleatorio, lo almacena en la sesión y lo embebe de manera oculta en este formulario. En el momento que el usuario envía este formulario, Rails valida el token guardado en la sesión con el enviado en el formulario HTML.

Validación de campos

Las validaciones de los campos se realizan a nivel de cliente utilizando las nuevas funcionalidades de HTML 5 y a nivel de servidor utilizando funcionalidades de Ruby on Rails.

Algunas validaciones están respaldadas por controles JavaScript, para el ingresar el nombre de usuario por ejemplo, el sitio no inserta espacios en el campo, mientras que además por HTML 5 y a nivel de modelo se agregan controles de largo.

Además de implementar validaciones en los modelos, se cuenta con verificaciones de unicidad y existencia para los campos en la base de datos. Este diseño en múltiples capas mejora la integridad de los datos y protege contra ataques externos.

Como medida de seguridad, todos los controladores, según la acción a realizar, controlan la sesión del usuario y además filtran los parámetros recibidos, además de proteger de manera nativa contra todo tipos de ataques malintencionados, como por ejemplo los de inyección de SQL.

Para lograr un sistema más robusto, todas las validaciones de unicidad se realizan tanto a nivel de base de datos, a nivel de modelos y a nivel de vistas. Las unicidades a nivel de cadena de caracteres se controlan ignorando las mayúsculas. El resto de las validaciones (como los largos de los campos o su obligatoriedad), al pertenecer la lógica de negocios y ser más susceptibles a cambios, se realizan solo a nivel de vistas y de modelo.

Sanitización de contenidos

El editor *TinyMCE* por defecto protege frente a la introducción de código malicioso, encapsulando todo el contenido del usuario en etiquetas de párrafo HTML.

Sin embargo, aunque el editor de texto filtra los contenidos, nada evita un ataque en el cual se sustituya el contenido de este por otro código malicioso (ya sea editando el código fuente HTML o realizando una petición HTTP independiente). No solo depende de la buena intención del usuario, y de los filtros a nivel cliente que el editor provee por lo cual fue necesario realizar controles de contenido del lado del servidor.

Por defecto, Rails protege contra todo tipo de ataques de inyección de código y los *sanitiza* al momento de manipularlos. Este filtro sin embargo no era útil al mostrar el contenido de las publicaciones, ya que el editor introduce código HTML para darle estilo a la publicación, y es necesario mostrar este código. Para resolver

este problema, se utilizó la gema *sanitize*⁵, la cual define por defecto una lista blanca de atributos permitidos.

Se definió la sanitización para filtrar el contenido introducido por los usuarios y que este sea luego introducido junto con el resto del código HTML del sitio, sin que este pueda ser indebidamente utilizado.

Además de la configuración por defecto, se estudiaron las etiquetas producidas por el editor de texto enriquecido y se configuró la sanitización de la gema para permitir ciertas etiquetas con ciertos atributos. No se permite el ingreso de ninguna etiqueta ni ningún atributo no permitido por estas configuraciones descritas, siendo este filtrado tanto a nivel de Frontend (por TinyMCE) como a nivel de Backend (por la gema). Mientras que para el resto de todos los campos introducidos por los usuarios en la plataforma, se utiliza la sanitización por defecto de Ruby on Rails.

Contraseñas

Basándose en los fundamentos de los requisitos de contraseña listados en NIST SP 800-63-3[7], el sitio requiere un largo entre 8 y 64 caracteres. También se pide que la contraseña contenga al menos una minúscula, una mayúscula y un símbolo. Las contraseñas se guardan encriptadas en la base de datos, utilizando OpenBSD bcrypt[25], al ser la implementación por defecto en Devise.

8.9. Frontend

El diseño del Frontend está diseñado en HTML, CSS y JavaScript utilizando JQuery *Bootstrap*. Se separó el código HTML del código JavaScript y CSS, de forma que facilite el desarrollo y lograr una aplicación modular.

8.9.1. Tecnologías y herramientas utilizadas

Bootstrap

Se utilizó Bootstrap 4 como base para el diseño del sitio. Esto tenía la ventaja de traer clases y estilos predefinidos, además de tener un soporte completo para dispositivos móviles y estar diseñado de manera *responsive*. Se consideró utilizar la

⁵<https://github.com/rgrove/sanitize>

última versión de Bootstrap (versión 5) pero se descartó por encontrarse en beta. Los iconos presentes en el sitio provienen de Bootstrap Icons ⁶ y son utilizados a través de un CDN.

reCAPTCHA

Se utilizó la gema de reCAPTCHA⁷, simplificando el uso de reCAPTCHA de Google, el cual utiliza un motor avanzado de análisis de riesgo para prevenir el uso malicioso del sitio[18]. Se configuró el sitio para utilizar la versión 2, la cual lleva consigo la resolución de un desafío y la versión 3, la cual utiliza inteligencia artificial que se basa en el comportamiento del visitante en el sitio.

JavaScript

Se utilizó JavaScript y JQuery como lenguajes de programación de Frontend, por ser el primero, el estándar en desarrollo web desde hace muchos años y el segundo por los años que tiene de desarrollo y uso.

8.10. Editor de código e interprete

Se logró reutilizar el editor de código y la consola del sitio MateFun, consta de librerías ya establecidas de JavaScript. El código se envía al servidor para ser procesado y luego se recibe para devolvérselo al usuario.

8.11. Gráficas

Se logró reutilizar el ambiente de gráficas 2D y 3D del sitio MateFun. El código se envía al servidor para ser procesado y luego se reciben los metadatos de la figura para transformarlos en imágenes 2D o 3D al usuario con librerías JavaScript.

⁶<https://icons.getbootstrap.com/>

⁷<https://github.com/ambethia/recaptcha>

8.12. Editor de texto enriquecido

Como editor de texto enriquecido se consideraron *Action Text* ⁸, nativo de Ruby on Rails, Quill ⁹ y TinyMce ¹⁰.

Se eligió TinyMce por la cantidad de funcionalidades disponibles por defecto (a diferencia de Action Text y Quill), buena documentación (que incluía varias configuraciones por defecto del editor) y uso extendido (más que *Action Text*, que está disponible por defecto solo para Ruby On Rails).

TinyMce es un editor de texto enriquecido que genera código HTML de manera transparente al usuario a partir del contenido ingresado en un editor amigable que recuerda a las interfaces antiguas de Microsoft Word y sus alternativas como OpenOffice y LibreOffice (ver Figura 8.4), lo cual se consideró como favorable para reducir la curva de aprendizaje de la plataforma, sobre todo para los docentes.

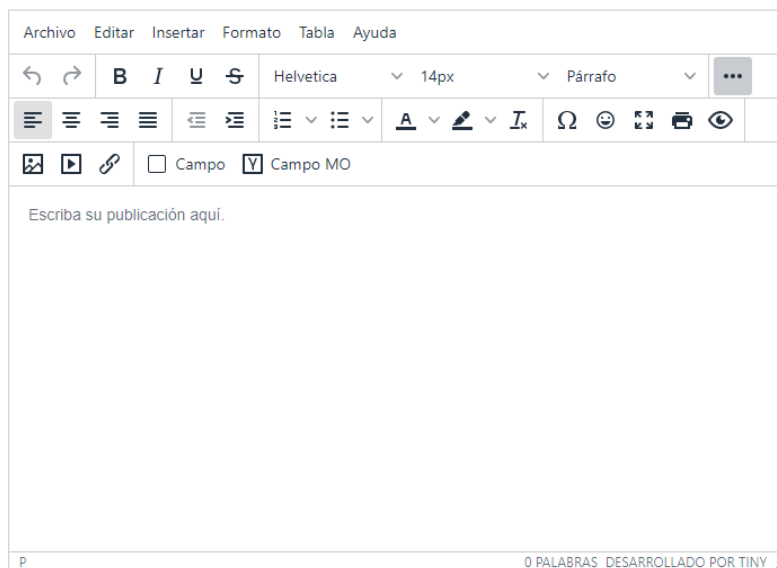


Figura 8.4: Vista del editor con los controles desarrolladores específicamente para Emmy.

Al utilizar HTML y no una sintaxis propia, se puede reemplazar este editor de texto por otro que trabaje de manera similar y mantener compatibilidad.

⁸https://edgeguides.rubyonrails.org/action_text_overview.html

⁹<https://quilljs.com/>

¹⁰<https://www.tiny.cloud/>

8.13. Ejercicios interactivos

El sitio cuenta con dos tipos de campos interactivos para la publicación de ejercicios. Estos son el campo de múltiple opción y el campo de completar, ambos se gestionan enteramente del lado del cliente, sin intervención del servidor.

Para su implementación se partió de ejemplos simples disponibles en la documentación y sitios de Internet, ya que no se encontró ninguna implementación previa de este tipo de campos para el editor de texto.

NOTA: Se recomienda consultar esta sección en el Anexo E.

8.14. Almacenamiento de imágenes

Se optó por almacenar las imágenes de las publicaciones en formato *Base64*. Esto es hecho automáticamente por *TinyMce*. A los usuarios registrados se les permite subir imágenes al sitio, mientras que los visitantes pueden adjuntar imágenes a través de un enlace (debiendo estar almacenadas en otro sitio), se diseña así para evitar un uso malintencionado del almacenamiento del sitio.

8.15. Fechas y zonas horarias

La gema de *ActiveRecord* de Rails almacena las fechas bajo la zona horaria UTC[2]. Para separar la presentación del almacenamiento de los datos, se mantuvo al configuración por defecto y mediante el uso de las librerías se convierten a la zona horaria del usuario.

8.16. Paginación de resultados

Al presentar resultados de búsqueda y listados, estos se pagan con la gema *Will-Paginate*¹¹, la cual ofrece soporte para estilos *Bootstrap*. Esta gema está basada en los comandos *LIMIT* y *OFFSET* de SQL[19].

De esta forma, cuando se solicita información solo se consulta a la base de datos por la cantidad de registros necesarias para mostrarle al usuario los resultados en

¹¹https://github.com/mislav/will_paginate

la página actual, evitando consultar registros innecesarios.

8.17. Código Secreto

El sitio no está diseñado para tener estudiantes registrados, todas las respuestas a publicaciones tienen un código secreto asociado de seis dígitos, único en la plataforma, el cual sirve para recuperar y continuar (en caso de haberla guardado como borrador) con la respuesta, todas las veces que se quiera, hasta el momento de enviarla. Una vez hecho esto, la respuesta se puede seguir consultando, pero sin opción a modificar, este código se genera por única vez y permanece invariable.

Se eligió un largo de seis ya que mantiene una buena relación entre cantidad de combinaciones y facilidad para ser manejada por los usuarios, (resultaría poco práctico por ejemplo, dar un código de 32 caracteres), además de que la memoria a corto plazo de los seres humanos puede recordar hasta siete elementos[15].

Con este largo elegido se tienen $2,176782336 \times 10^9$ combinaciones, las cuales se acabarían luego de 60 años ¹².

8.18. Breadcrumbs

Se utilizó la gema Gretel¹³ para definir los breadcrumbs del sitio, estos son del tipo *localización*, el cual además se comporta como uno dinámico[11] en la mayoría de los casos.

8.19. Listas desplegables autocompletables

Para los filtros, se definieron listas desplegables autocompletables, las cuales ayudan a seleccionar la opción deseada.

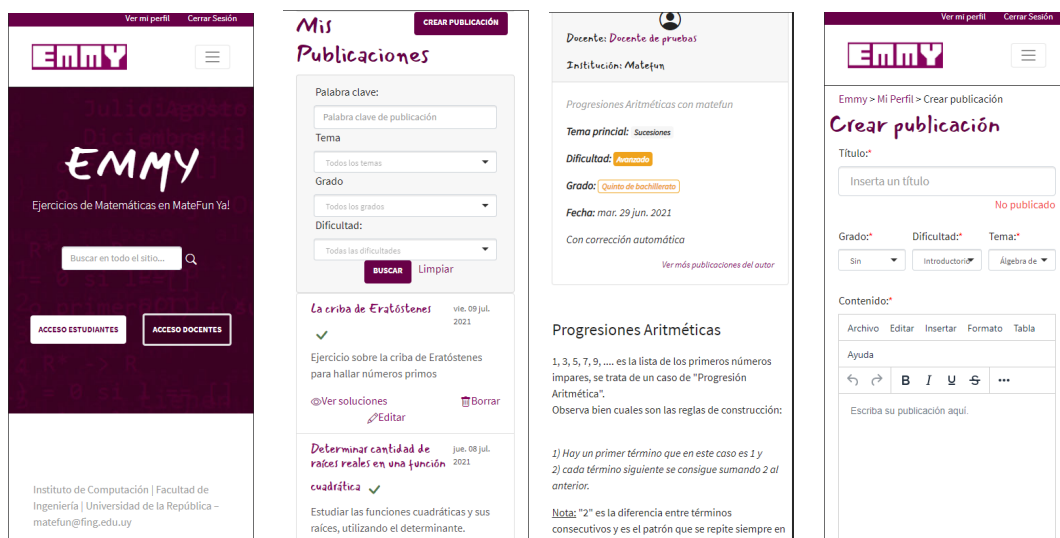
¹²Si se supone 100 respuestas por día (2176782336/100/365) sin haber borrado ninguna en ese periodo.

¹³<https://github.com/kzkn/gretel>

Diseño del sitio

En este capítulo se hace un recorrido por las pantallas del sitio, se explican las decisiones de diseño tomadas y se menciona la influencia de las pruebas con usuario en la plataforma. Se habla del diseño adaptable y se mencionan brevemente las funcionalidades del sitio. Junto con cada figura se adjunta un enlace al sitio.

9.1. Diseño adaptable



(a) Página principal (b) Perfil del docente (c) Vista de publicación (d) Crear publicación

Figura 9.1: Vista del sitio en resolución móvil(360x780)

Se utilizó el sistema de grilla disponible de Bootstrap 4, el cual está diseñado para abarcar todas las configuraciones, formas y tipos de pantalla [8].

Utilizando las clases *col* del tamaño *sm*, *md*, *lg* y *xl* se definieron todas las presentaciones de elementos, de forma que estos fueran legibles y mantuvieran la apariencia sin importar el dispositivo donde se estuvieran viendo.

Aunque resultaría poco práctico escribir publicaciones o programar en lenguaje MateFun (ya que son tareas mucho más difíciles) en un dispositivo móvil, se prestó especial atención y se hicieron recorridos por todas las páginas del sitio controlando

la correcta visualización de elementos para las resoluciones de escritorio 1366×768 (incluye a los equipos Sirio de Ceibal [9] utilizados en educación secundaria) y 1920×1080 , ya que son las más comunes del Uruguay (con un 40% y 17% de utilización respectivamente)[6], como muestra la Figura 9.2.

En el ámbito móvil, se probaron resoluciones de 360×640 y 360×780 , las cuales representan un porcentaje de utilización de 17% y 6% respectivamente en el Uruguay [12] (Figura 9.2).

Finalmente, Figura 9.1 muestra cómo se adapta el sitio a una resolución 360×780 .

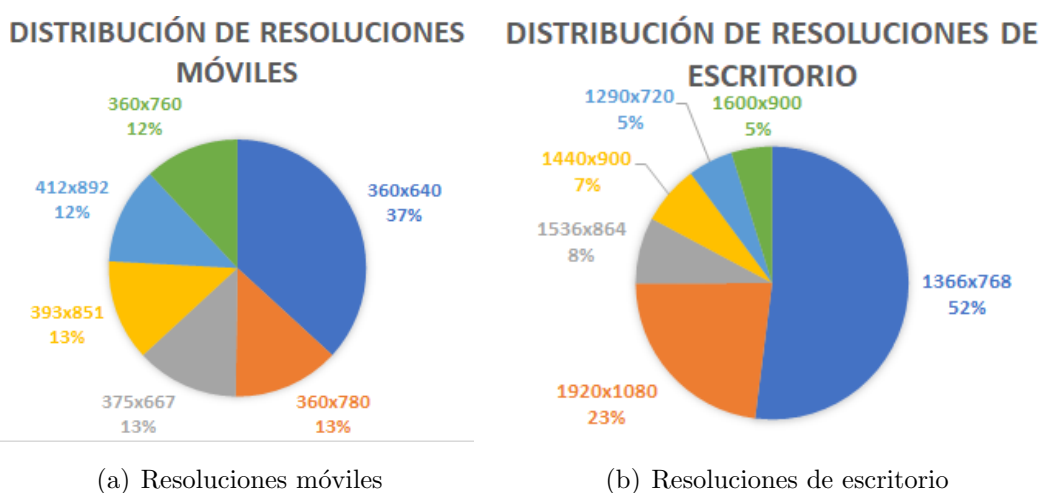


Figura 9.2: Distribución de resoluciones en Abril 2021, Uruguay (StatCounter).

9.2. Diseño de las interfaces

El sitio se diseñó al principio utilizando Bootstrap y un estilo de paleta de colores en celeste, como muestra la Figura 9.3, el cual utiliza una barra de navegación del mismo color y una pantalla de bienvenida que utiliza un elemento *Jumbotron* en el centro. Luego de tener este primer diseño se introdujeron requerimientos de estilo diferentes.

Se sugirió utilizar un estilo similar al sitio de MateFun ¹ mostrado en la Figura 9.4. Al surgir este requerimiento luego de tener implementada la estructura

¹<http://d11.com.uy/web/matefun/dev03/>

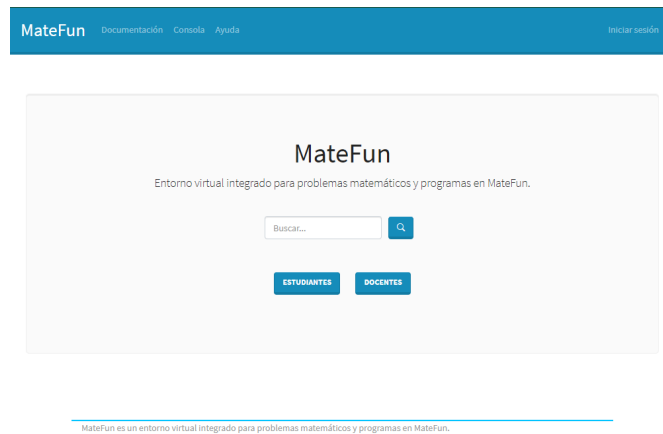


Figura 9.3: Diseño inicial del sitio

y como el sitio a replicar tenía un diseño cerrado, se realizó una ingeniería reversa inspeccionando el código, extrayendo la paleta de colores, estructuras básicas, dimensiones, estilos de letra e imágenes de los principales componentes.

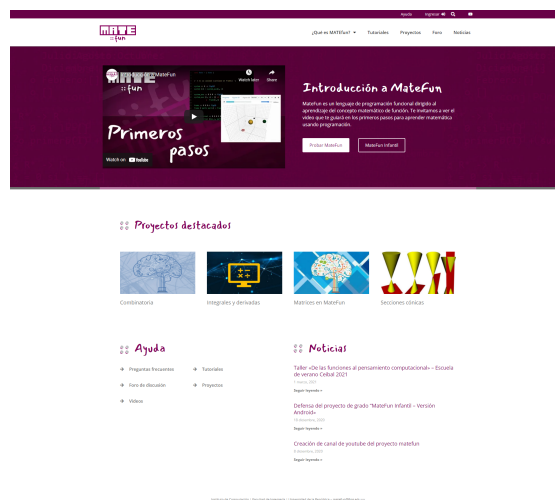


Figura 9.4: Base del nuevo diseño

Luego de aplicar los estilos de diseño (Figura 9.5), se continuó con el desarrollo, basando las decisiones en los conocimientos adquiridos en la asignatura Interacción Persona Computadora, (el cual basa sus fundamentos en los estudios de usabilidad del autor Donald Norman). Posteriormente se validó y se corrigieron los detalles con la retroalimentación obtenida por parte de tutores, docentes y estudiantes.

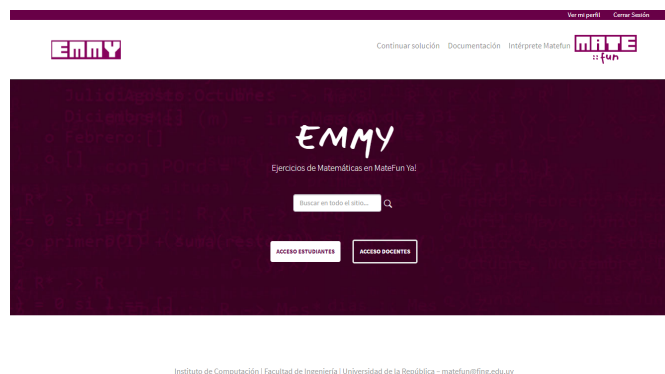


Figura 9.5: Nuevo diseño de Emmy ²

9.2.1. Página principal

La entrada al sitio se hizo similar al sitio de MateFun (Figura 9.5). Se muestra en el centro el *acceso estudiantes*, el *acceso docentes* y un buscador que tiene un mensaje *Buscar en todo el sitio* esto se agregó ya que algunos docentes en las pruebas de usuario lo utilizaron para buscar las publicaciones propias.

El botón de *acceso docentes* en un principio se llamó *docentes* y se diseño como un acceso rápido al listado de docentes, sin embargo los usuarios de prueba usaban este botón para iniciar sesión, por lo que se terminó redefiniendo su funcionalidad.

Barra de navegación

También se muestran otros enlaces, como el de continuar solución, la documentación del sitio y acceso a la web principal de MateFun, que se consideraron útiles. Por último, existe un enlace permanente al iniciar sesión. En el resto de las pantallas, la barra de navegación muestra el buscador global.

Pie de página

El pie de página es simple y copió el implementado en el sitio de MateFun. Cuenta con un color de letra gris para no llamar la atención del usuario.

²<https://emmy-fing.herokuapp.com/>

Breadcrumb

Se implementaron breadcrumbs basándose en las recomendaciones del grupo consultor de interfaz y experiencia de usuario *Nielsen Norman Group*. Este breadcrumb está posicionado en la parte superior de la pantalla, justo debajo de la barra de navegación y es del tipo jerárquico, que es el modelo recomendado [3]. Estudios realizados por esta consultora, han demostrado que este tipo de ayuda de navegación es realmente útil y ofrece beneficios sin ninguna desventaja [3].

En las pruebas de usuarios se comprobó empíricamente su utilidad, habiendo un usuario de prueba basado toda su navegación en este instrumento.

Para el diseño, se siguieron las guías recomendadas por esta institución [4], y sus detalles de implementación están descritos en la Sección 8.18, Figura 9.6 muestra un ejemplo.



Figura 9.6: Breadcrumbs ³

Acceso estudiantes

Basándose en la navegación del sitio *NRICH* de la Sección 5.1.1, se muestran tres tarjetas de navegación: por tema, grado y docente (Figura 9.7), se consideraron como los tres principales filtros. No es posible navegar por dificultad ya que es relativa al grado (si alguien busca dificultad intermedia por ejemplo, le aparecerían tanto ejercicios intermedios de secundaria como universitarios, por lo cual el filtro no es útil por sí solo).

Inicialmente el título era *Estudiantes* pero con ese título los docentes entrevistados esperaban ver sus alumnos, así que se cambió por *Navegación*.

³<https://emmy-fing.herokuapp.com/publications/28/answers/new>

⁴<https://emmy-fing.herokuapp.com/students>

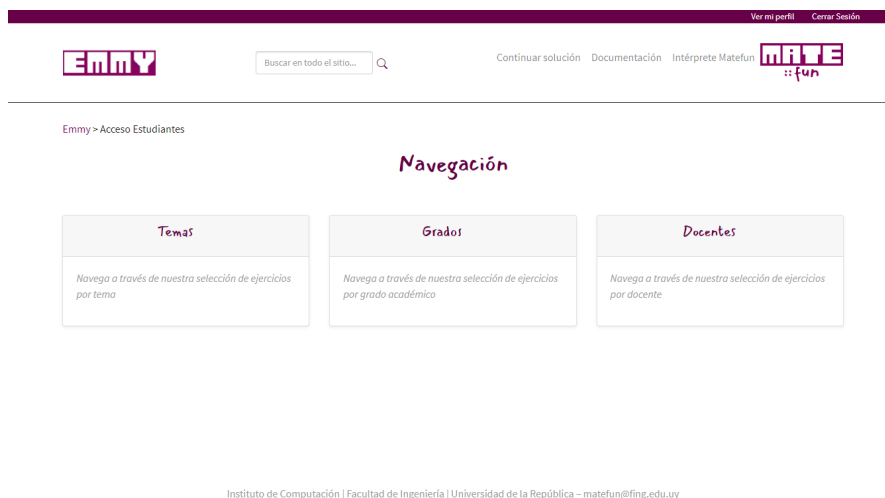


Figura 9.7: Navegación ⁴

Listar temas, grados y docentes

Las pantallas de temas, grados y docentes son similares, para los primeros, se muestra el listado de temas, con paginación de doce elementos (ya que es un número manejable que no requiere desplazar de pantalla en resoluciones de escritorio), ordenadas por cantidad de publicaciones (en estado publicado) de manera descendente. De esta forma, los estudiantes tienen acceso a los temas con mayor contenido.

Para los grados (Figura 9.8), se muestran tarjetas solamente con los grados que tengan publicaciones en estado publicado, ordenadas académicamente de menor a mayor, junto con la categoría *Sin especificar*. Se utiliza un esquema de colores similar al de dificultades, para que el visitante cree agrupaciones visuales rápidas, agrupando los grados escolares en azul, ciclo básico en verde, bachillerato en naranja y los universitarios en rojo.

Para los docentes (Figura 9.9), se muestran tarjetas solamente de aquellos que tengan publicaciones en estado publicado, ordenados por cantidad de publicaciones de manera descendente, junto con una descripción que indica las instituciones del docente, esto les permitirá a los estudiantes encontrarlos rápidamente.

⁵<https://emmy-fing.herokuapp.com/grades>

⁶<https://emmy-fing.herokuapp.com/users>

Ver mi perfil Cerrar Sesión

EMMY Continuar solución Documentación Intérprete Matefun **matefun**

Emmy > Acceso Estudiantes > Grados

Navegar por grado

Sin especificar

Tercer año escolar

Ciclo básico

Primer de ciclo básico

Bachillerato

Quinto de bachillerato

Universitario

Instituto de Computación | Facultad de Ingeniería | Universidad de la República – matefun@fing.edu.uy

Figura 9.8: Listado de grados ⁵

Ver mi perfil Cerrar Sesión

EMMY Continuar solución Documentación Intérprete Matefun **matefun**

Emmy > Acceso Estudiantes > Docentes

Navegar por docente

Docente de pruebas

Docente en Matefun

MateFun

Docente en FING

Gastón B.

Docente en FING

Verónica B.

Docente en FING

Instituto de Computación | Facultad de Ingeniería | Universidad de la República – matefun@fing.edu.uy

Figura 9.9: Listado de docentes ⁶

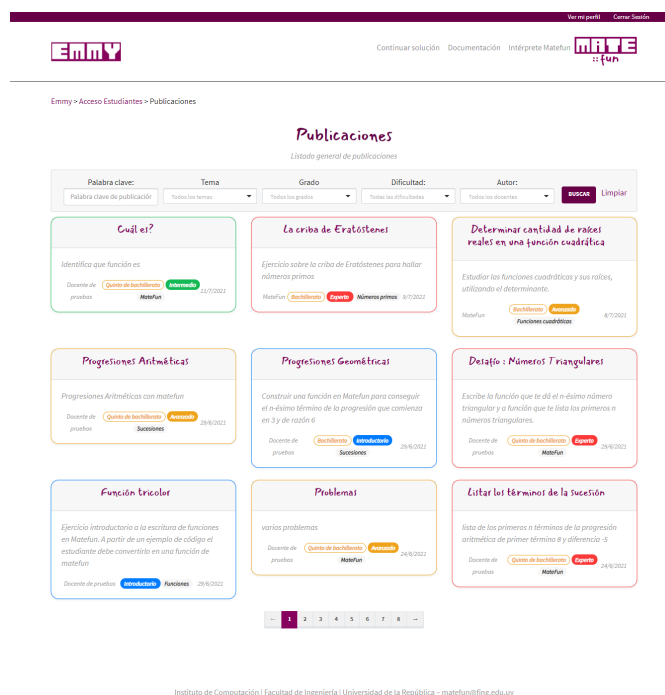


Figura 9.10: Listado de publicaciones ⁷

Buscar publicaciones, listar un tema, un grado o un docente

La pantalla de búsqueda de publicaciones (Figura 9.10) incluye los filtros en la parte superior, estos filtros son por palabra clave, el cual busca las ocurrencias de la palabras ingresadas (a partir del estudio de las raíces lingüísticas de cada una) en el título, en la descripción y en el contenido (con peso ponderado en ese orden), se encuentra detallado su funcionamiento en la Sección 8.7.

Se incluyen también filtros por temas (ordenados alfabéticamente), grados, dificultades (ordenadas de más fácil a más difícil) y autores (ordenados alfabéticamente). Se puede seleccionar el filtro rápidamente escribiendo sobre estos, lo cual es práctico cuando existe una gran cantidad de elementos que dificulta la selección. Se muestra hasta cuatro resultados relacionados y no se muestran en los filtros correspondientes, aquellos temas y autores que no tengan publicaciones en estado publicado (para no mostrar resultados de búsqueda vacío).

Algunos filtros son omitidos dependiendo de cómo se accedió a la búsqueda

⁷<https://emmy-fing.herokuapp.com/publications>

publicaciones, si se accedió a través del listado de temas, se omite el filtro correspondiente (análogamente si se accede por grado o por autor). Tiene la ventaja de no mostrar filtros redundantes, como muestra la Figura 9.11.



The image shows a search interface for the topic "Números primos". It features a horizontal layout of filters: "Palabra clave:" with a text input field containing "Palabra clave de publicaci"; "Grado:" with a dropdown menu showing "Todos los grados"; "Dificultad:" with a dropdown menu showing "Todas las dificultades"; and "Autor:" with a dropdown menu showing "Todos los docentes". To the right of these filters are two buttons: a purple "BUSCAR" button and a grey "Limpiar" button.

Figura 9.11: Acceso a la búsqueda de publicaciones a través de tema ⁸

En los dispositivos móviles, los filtros se muestran por columna (Figura 9.12).



The image shows a search interface on a mobile device. The filters are stacked vertically: "Palabra clave:" with a text input field containing "Palabra clave de publicación"; "Tema:" with a dropdown menu showing "Todos los temas"; "Grado:" with a dropdown menu showing "Todos los grados"; "Dificultad:" with a dropdown menu showing "Todas las dificultades"; and "Autor:" with a dropdown menu showing "Todos los docentes". At the bottom are two buttons: a purple "BUSCAR" button and a grey "Limpiar" button.

Figura 9.12: Filtros apilados en una resolución de dispositivo móvil.

Tarjetas de publicación

El listado de publicaciones se compone de tarjetas (Figura 9.13), comienza con un título con estilo y color en la parte superior, seguido de la descripción corta de la misma en un tono de gris, se busca que la descripción pase a segundo plano y no se quite la atención al título, que es la forma de identificar una publicación.

En último lugar, aparecen también en color gris el nombre del autor, la fecha (día/mes/año) y hasta tres píldoras informativas, estas son: dificultad, grado y tema. Las dificultades se muestran como una píldora con fondo sólido, siguiendo un esquema de colores (azul, verde, naranja y rojo), lo que permite una identificación intuitiva, al mismo tiempo que el borde de la tarjeta tiene el mismo color que la dificultad, mientras que los grados se muestran con la misma configuración de

⁸<https://emmy-fing.herokuapp.com/topics/62/publications>

colores, pero con fondo blanco. Los temas se especifican como una píldora de fondo gris y letras negras.

Al ver una tarjeta de publicación, lo primero que se espera que se visualice es el título seguido del grado, dificultad y tema, como se muestra a continuación.

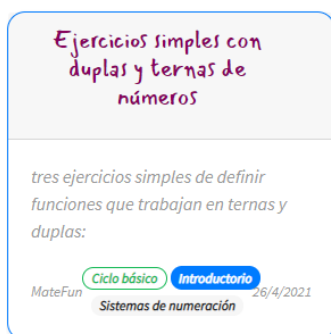


Figura 9.13: Tarjeta de publicación.

Este diseño de píldoras está inspirado en el diseño de *LeetCode* descrito en el Relevamiento (Sección 5.1.4).

Ver publicación

La visualización de una publicación se compone de título en la parte superior, una tarjeta de publicación extendida y los botones de responder, editar (solo si se es el autor) y volver.

Tarjeta de publicación extendida

La tarjeta de publicación (Figura 9.15) extendida se compone de un nombre de autor y su institución. En segundo lugar se encuentra su información descriptiva, esta incluye la descripción corta en un tono gris (logrando una menor atención), tema, dificultad, grado, fecha de publicación (incluye el día de la semana, ya que en entrevistas con usuario se ha resaltado su utilidad) y si cuenta con corrección automática. Por último, se muestra en un tipo de letra más pequeño un enlace para ver más publicaciones del autor (lo cual también se puede hacer accediendo a través del nombre del autor) y ver las soluciones (si es el autor). Esta tarjeta se mueve hacia arriba de la publicación si la resolución es muy pequeña.

⁹<https://emmy-fing.herokuapp.com/publications/6>

Emmy > Acceso Estudiantes > Docentes > MateFun > La criba de Eratóstenes

La criba de Eratóstenes

Eratóstenes



Eratóstenes fue un matemático, astrónomo y geógrafo griego, que desarrolló la Criba de Eratóstenes.

La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

¿Cuál de estos números no es primo? Seleccionar una opción



Algoritmo:



Primero se forma una tabla con todos los números naturales comprendidos entre 2 y n y se van tachando los números que no son primos de la siguiente manera:
Comenzando por el 2, se tachan todos sus múltiplos; comenzando de nuevo, cuando se encuentra un número que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente.

El proceso termina cuando el cuadrado del siguiente número confirmado como primo es mayor que n .

Se pide implementar la criba de Eratóstenes para un N arbitrario y completar el siguiente cuadro (separados con coma, sin espacios entre medio):

Valor de n	Números primos
20	cuadro de completar

RESPONDER

EDITAR

[Volver a publicaciones](#)

Figura 9.14: Vista de una publicación ⁹

Las píldoras de tema, dificultad y grado coinciden en formato con la tarjeta de publicación.

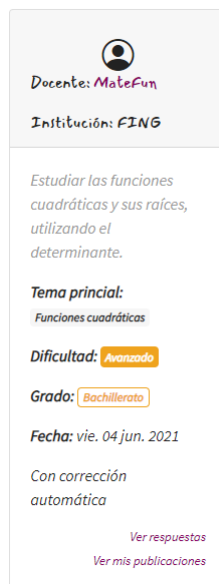


Figura 9.15: Tarjeta de publicación extendida.

9.2.2. Estudiantes

Responder publicación

Al momento de acceder a una publicación, se permite comenzar a escribir la respuesta, esto no era así al principio, donde solamente se mostraba el contenido y se debía tocar el botón resolver. En las pruebas de usuarios, se detectó que todos intentaban responder la publicación sin antes tocar este botón, por lo que se terminó modificando el flujo.

Los estudiantes pueden responder aquellas publicaciones que tengan el envío de respuesta habilitado (Figura 9.16). De los campos identificativos, es obligatorio ingresar un nombre y apellido. El código privado es un código opcional que se coordina entre docentes y estudiantes.

En todo momento, los estudiantes seguirán viendo la tarjeta de publicación extendida en el mismo lugar que antes.

¹⁰<https://emmy-fing.herokuapp.com/publications/61/answers/new>

Ver mi perfil Centro Sesión

Emmy Buscar en todo el sitio... Continuar solución Documentación Intérprete Matlab **mate**

Acceso Estudiantes > Publicaciones > Función tricolor > Responder

Función tricolor

Responder

Nombre y apellido: Institución: Código privado:

A modo de ejemplo prueba este código en el intérprete
`juntar(color(dirc(2),Rojo),juntar(color(dirc(2),Azul),color(dirc(2),Blanco)))`

Estudialo y observa el resultado que produce

Vamos a escribir una nueva función en tu archivo que haga lo siguiente:
"Dados tres círculos concéntricos de radios 3, 2 y 1 queremos poder asignarle los colores que elijamos"

El nombre de la función será "tricolor" o el que tu prefieras.

La "signatura" de la función debe ser:
`tricolor = (Color X Color X Color) -> Fig`

Significa que los argumentos de entrada son tres elementos del conjunto "Color" y la salida es un elemento del conjunto "Fig", es decir, de las Figuras.

La forma de usar la función será, por ejemplo: `tricolor(Verde, Amarillo, Azul)`

Docente: Dirección de pruebas
 Institución: Matefun

Ejercicio introductorio a la escritura de funciones en Matlab. A partir de un ejemplo de código el estudiante debe convertirlo en una función de matlab

Tema principal: Funciones
 Dificultad: **Intermedio**
 Grado: **Intermedio**
 Fecha: mar, 29 Jun, 2021
 Ver más publicaciones del autor

Respuesta:

Archivo Editar Insertar Formato Tabla Ayuda

Escriba su respuesta aquí.

DESARROLLADO POR TITV

CÓDIGO GRÁFICA 1D GRÁFICA 2D

CÓDIGO

Sin Archivos

Incorporamos el código Matlab en el envío de tu respuesta.

ENVIAR RESPUESTA GUARDAR COMO BORRADOR Volver

Instituto de Computación | Facultad de Ingeniería | Universidad de la República - matefun@ing.edu.uy

Inicio Sesión

Figura 9.16: Responder una publicación ¹⁰.

Si la publicación tenía campos interactivos, estos podrán ser completados por el usuario. Los que sean del tipo múltiple opción tendrán un mensaje indicando que deben seleccionar una opción, mientras que los completar dirán que son un campo de texto.

Si la publicación tenía habilitada la opción de enviar texto, tendrán acceso al editor de texto (detallado en la Sección 8.12), mientras que si tenía habilitado el

editor de código o el intérprete tendrán la opción de utilizarlos.

Los estudiantes pueden enviar la respuesta o guardarla como borrador, en cuyo caso obtendrán un código secreto con el cual podrán seguir editándola hasta que la envíen definitiva.

Como los estudiantes no tienen la posibilidad de registrarse en el sitio, se implementó el código de acceso tanto para retomar una solución como para volver a visualizarla. Este código no debería ser compartido entre usuarios y es invariable (una vez generado, se mantiene fijo) (Sección 8.17).

Visualizar respuesta

Cuando se envía la respuesta, se ve la corrección automática, si está habilitada en las opciones de publicación (Figura 9.24). También pueden volver a ella a través del código secreto.

Se muestra a la derecha la tarjeta de respuesta es idéntica a la vista por los docentes e incluye de utilidad para los estudiantes, el código secreto (para que puedan recuperarlo).

Continuar solución

Para continuar una solución (Figura 9.17) los estudiantes pueden ingresar el código secreto en la opción *Continuar solución* de la barra de navegación. Inicialmente esta opción se llamaba *Cargar solución* pero en las pruebas con usuarios manifestaron que se entendía que permite subir una solución, en vez de continuar una existente.

Se agregó en el campo donde se ingresa el código, que no se permitan ingresar espacios, ya que algunos docentes copiaban los espacios antes y después del código antes de insertarlos.

9.2.3. Usuarios

Iniciar sesión

El inicio de sesión (Figura 9.18) se compone de usuario, contraseña (con un botón para mostrarla) y una opción de recordar los datos ingresados. Como el

Figura 9.17: Ingresar código secreto para continuar la solución.

registro de usuarios lo hacen los usuarios administradores, no se muestra esta opción. Esta funcionalidad cuenta además tecnología reCAPTCHA (Sección 8.9.1).

Figura 9.18: Inicio de sesión.

Modificar cuenta

Los usuarios pueden modificar su correo, nombre completo, institución y la contraseña, debiendo ingresar la contraseña actual para confirmar los cambios. El campo de *nombre completo* se llamaba inicialmente *nombre*, pero gracias a la retroalimentación de los usuarios, se lo renombró.

9.2.4. Administradores

El panel de administración se compone de una sección principal, con seis pestañas superiores y un menú a la derecha.

Resumen

El resumen contiene algunas estadísticas de uso de la plataforma (Figura 9.19), se incluye los temas más populares (se limitan a ocho para no saturar la gráfica), en formato de torta. Se visualiza la proporción, cantidad de publicaciones por dificultad, (en formato de columnas mostrando las cantidades) y dos gráficas por fecha, que muestran el uso del sitio. Se listan las publicaciones y respuestas publicadas por fecha de los últimos seis meses

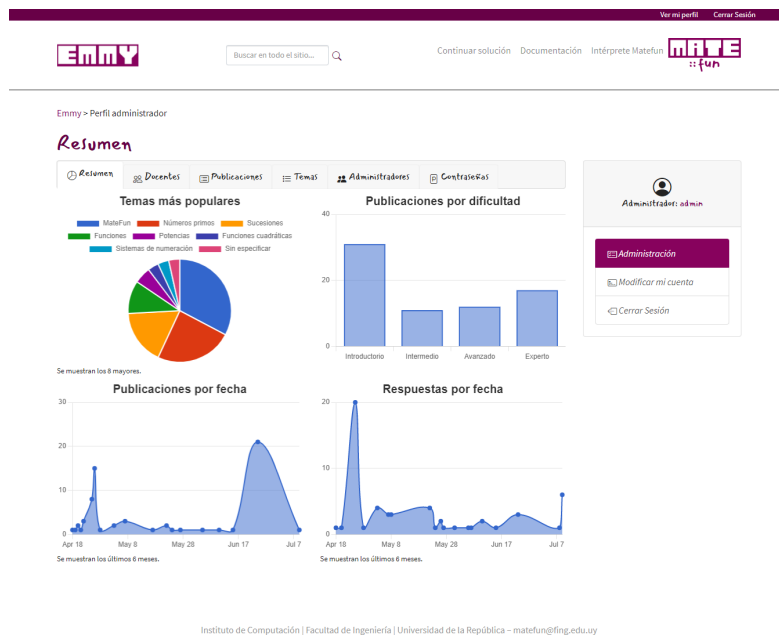


Figura 9.19: Estadísticas de uso para administradores ¹¹.

Administrar docentes y administradores

Se listan las principales propiedades de los usuarios administradores y docentes (cada uno en su pantalla correspondiente). De estos se muestra el nombre (a partir de él se puede acceder al perfil), el correo, el nombre de usuario y la institución.

Se incluye un filtro autocompletable por nombre y filtros por subcadena de correo, usuario e institución. Estos últimos podrían ser útiles por ejemplo para

¹¹<https://emmy-fing.herokuapp.com/>

Emmy > Perfil administrador > Temas

Temas

Retenes Docentes Publicaciones **Temas** Administradores Contraseñas

Buscar por nombre: Filtrar

NOVO TEMA

<input type="checkbox"/>	Nombre ▲	Publicaciones
<input type="checkbox"/>	Álgebra de matrices	2
<input type="checkbox"/>	Amortizaciones	0
<input type="checkbox"/>	Aproximación de números	0
<input type="checkbox"/>	Área y perímetro del círculo	0
<input type="checkbox"/>	Cálculo diferencial	0
<input type="checkbox"/>	Cambio de base	0
<input type="checkbox"/>	Combinatoria	0
<input type="checkbox"/>	Congruencia	0

Administrador: admin

- Administración
- Modificar mi cuenta
- Cerrar Sesión

Instituto de Computación | Facultad de Ingeniería | Universidad de la República - matefun@fing.edu.uy

Figura 9.20: Listado de temas para administradores.

filtrar usuarios con un correo institucional, o para filtrar tipo de instituciones (por ejemplo, con la palabra *liceo* o *colegio*).

Administrar publicaciones

Las publicaciones se listan en formato tabla de tres columnas, incluyen título, descripción corta, el autor y la fecha. Se incluyen filtros por subcadena en título y filtro autocompletable de autor.

Administrar temas

En Figura 9.20 se listan todos los temas en formato tabla, ordenados alfabéticamente. Se incluye la cantidad de publicaciones en estado publicado correspondientes. Los temas se pueden editar seleccionando el nombre (aparece además el campo descripción) y al seleccionar la cantidad de publicaciones se accede al listado de publicaciones por tema.

Emmy > Mi Perfil > Mis publicaciones

Mis Publicaciones CREAR PUBLICACIÓN

Palabra clave: Tema: Grado: Dificultad: BUSCAR Limpiar

Publicación	Fecha	Acciones
La criba de Eratóstenes ✓ Ejercicio sobre la criba de Eratóstenes para hallar números primos	vie. 09 jul. 2021	Ver soluciones Editar Borrar
Determinar cantidad de raíces reales en una función cuadrática ✓ Estudiar las funciones cuadráticas y sus raíces, utilizando el determinante.	jue. 08 jul. 2021	Ver soluciones Editar Borrar
Serie: Introducción a MateFun ✓ Introducción a MateFun a través de una serie de publicaciones	jue. 24 jun. 2021	Editar Borrar
Introducción a las gráficas ✓ Aprende a utilizar el comando de graficar	jue. 24 jun. 2021	Editar Borrar
Buscar elementos en listas ✓ Tres ejercicios relacionados sobre encontrar elementos en listas	lun. 26 abr. 2021	Editar Borrar

1 2 3 4 5 6

Volver

Docente: MateFun
Institución: FING

Mis publicaciones
Modificar mi cuenta
Cerrar Sesión

Instituto de Computación | Facultad de Ingeniería | Universidad de la República - matefun@fing.edu.uy

Figura 9.21: Vista del perfil usuario ¹².

Administrar contraseñas

Como no existe una forma de recuperar la contraseña, se les dio a los administradores la posibilidad de modificarle la contraseña a cualquier usuario (incluido otros administradores).

9.2.5. Docentes

Perfil

El perfil de los docentes consiste de un listado de publicaciones, contiene un título (al seleccionarlo se puede acceder a la publicación), un icono de publicado (que además indica publicado al posicionar el cursor encima) y la descripción corta

¹²<https://emmy-fing.herokuapp.com/users>

en un color gris. Cada una de estas muestran opciones (que se les agregaron iconos) de editar, de borrar y de ver las soluciones enviadas (en caso de existir). También se muestra la fecha en un formato más completo con día de la semana, ya que hubo docentes que manifestaron su utilidad. Se incluyeron además los mismos filtros descritos en la Sección 9.2.1.

El título inicialmente era *Publicaciones* pero generó confusiones con los usuarios de prueba, creyendo que era el listado general de todas las publicaciones de la plataforma, por lo que se renombró a *Mis publicaciones*.

Al botón de crear publicación se le había asignado una posición debajo del menú lateral, pero muchos usuarios (en la primera fase de iteración) tuvieron dificultades para encontrarlo, por lo cual se cambió a la posición que se muestra en la Figura 9.21, ubicación que no causó problemas en la segunda iteración.

Crear publicación

Para crear una publicación (Figura 9.22) se muestra el título, grado, dificultad y tema. Las opciones a estos filtros están fijas (evitando así duplicación de temas con nombres muy similares), aunque para mayor facilidad y para encontrar rápidamente la opción deseada, son autocompletables por subcadena.

Debajo, se muestra el editor de texto, luego le sigue el editor de MateFun, el cual se encuentra por defecto oculto. Al activarlo, se muestra la el editor completo, esto incluye el editor de código, el intérprete y las gráficas. Gracias a las pruebas de usuarios se le agregó una breve descripción al posicionar el cursor sobre el botón de campos interactivos.

Todo el código que se cargue en el editor es precargado en las respuesta de los estudiantes, esto es útil por ejemplo si se desea darles una ayuda; guiándolos con el dominio de una función o dándoles la implementación de una función auxiliar.

Seguido se encuentra la descripción corta, la cual aparece en la tarjeta de publicación y en la tarjeta de publicación extendida. Por último, se muestran las opciones de publicación, por defecto, se encuentran activadas todas.

Si se desactiva el envío de respuestas, automáticamente se desactiva la corrección automática. Esta opción es útil si se desea publicar un ejercicio de práctica y no interesa conocer lo que han hecho los estudiantes. Por otro lado, se pue-

¹³<https://emmy-fing.herokuapp.com/publications/new>

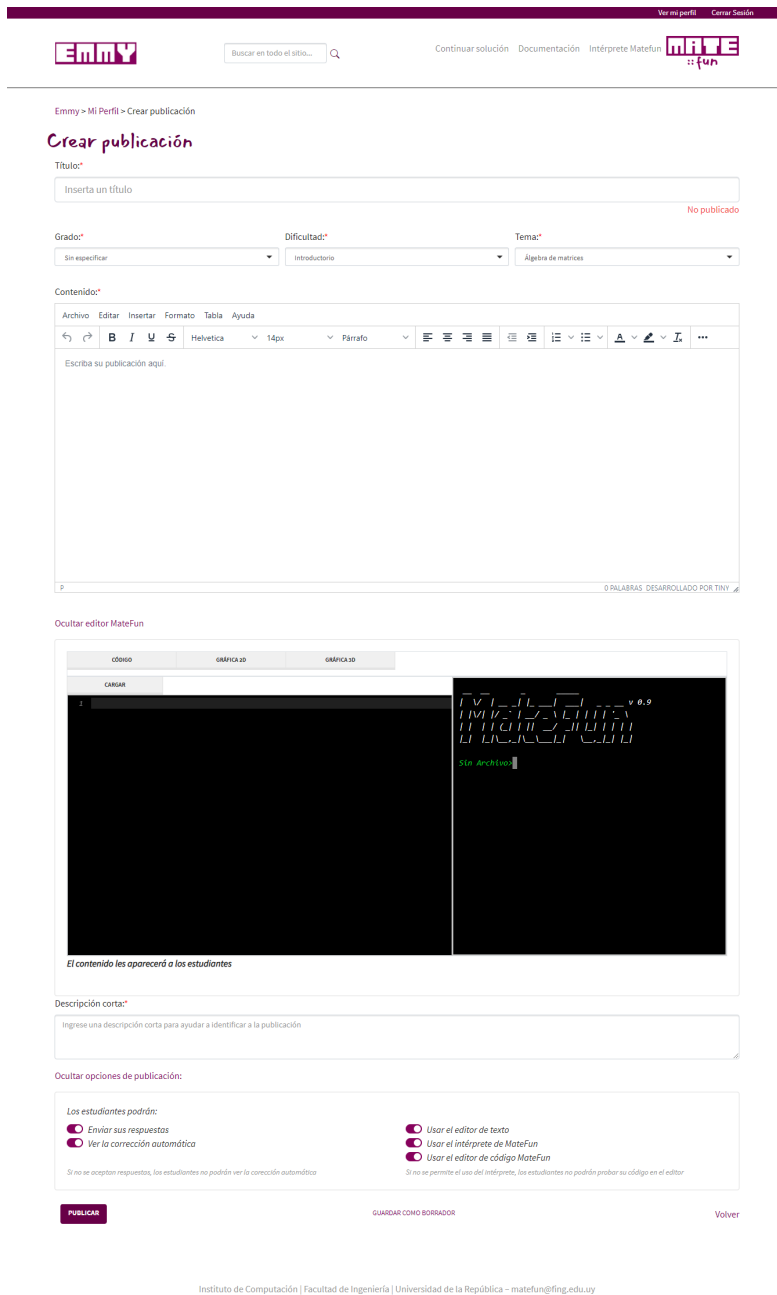


Figura 9.22: Crear publicación, mostrando opciones avanzadas y el entorno MateFun ¹³.

de desactivar solo el editor de texto o de código MateFun, pero si se desactiva el intérprete se deshabilita automáticamente el editor de código (ya que no tiene sentido mostrar un editor de código del cual no se va a poder probar su contenido).

La decisión de ocultar por defecto las opciones de publicación y el editor de MateFun fue simplificar la pantalla de crear publicaciones, tratando de mostrar un editor básico que tuviera todo lo necesario e imprescindible para la publicación de contenido.

Finalmente, la publicación se puede publicar o guardar en borrador (la cual solo podrá ser vista por el autor hasta el momento de publicarla). Una vez publicada, no se puede cambiar su estado.

Listar soluciones

Emmy > Mi Perfil > Mis publicaciones > Respuestas a Determinar cantidad de raíces reales en una función cuadrática

Respuestas a **Determinar cantidad de raíces reales en una función cuadrática**

Buscar por nombre: Buscar por institución: Buscar por código:

<input type="checkbox"/>	Nombre	Institución	Código privado	Enviado
<input type="checkbox"/>	Juan Pérez ✓	Liceo 2021	PrimerosA	jue. 08 Jul. 2021
<input type="checkbox"/>	Juan Pérez ✓	Liceo 2021	PrimerosA	jue. 08 Jul. 2021
<input type="checkbox"/>	Juan Perez ✓			jue. 08 Jul. 2021
<input type="checkbox"/>	María Eugenia Curti ✓	PreU	Lic34-SB	sáb. 05 Jun. 2021
<input type="checkbox"/>	yenny adinoff ✓	IPA	5A	vie. 04 Jun. 2021
<input type="checkbox"/>	Santiago Jaureguiberry ✓	Utu Solymar Norte		mié. 26 May. 2021
<input type="checkbox"/>	Julietta Barreto ✓	Magisterio	liceo 10 primero A	mié. 26 May. 2021
<input type="checkbox"/>	Lota M ✓	Anep	2789	lun. 24 May. 2021
<input type="checkbox"/>	Eliana Oer ✓	liceo	333	sáb. 22 May. 2021
<input type="checkbox"/>	Agustín Breccia ✓	IEP	5to C	vie. 07 May. 2021

- 1 2 -

Volver

Instituto de Computación | Facultad de Ingeniería | Universidad de la República - matelun@fing.edu.uy

Figura 9.23: Listar soluciones a publicación ¹⁴.

El listado de soluciones (Figura 9.23) es en forma de tabla, se incluyen las columnas *nombre*, *institución*, *código privado* y *enviado* y filtros por subcadena

¹⁴<https://emmy-fing.herokuapp.com/users/4/publications/3/answers>

para los tres primeros, ya que con esto se abarca todas las opciones de filtrado posibles. Se puede además ordenar por columna. Las respuestas que ya no están en estado borrador llevan consigo el mismo icono que las publicaciones y al seleccionar su nombre se accede a la misma.

Visualizar respuesta

Al visualizar una publicación para responder (Figura 9.24) se muestra su título centrado y el contenido debajo, junto con la tarjeta de respuesta a la derecha. Esta visualización está disponible para aquellas publicaciones que tengan habilitado el envío de respuestas.

La tarjeta de respuesta es similar a la tarjeta extendida de publicación, la cual incluye el nombre y la institución del estudiante en primer lugar, y luego debajo detalles como su código privado, la fecha y hora (se consideró que la hora de entrega era importante para los docentes) y el código secreto con el enlace de acceso (el cual podía resultar útil en caso de extravío por parte de los estudiantes, algo que los docentes de prueba consideraron útil). Esta tarjeta se mueve hacia arriba de la publicación si la resolución es muy pequeña.

Si la publicación tiene campos interactivos, estos con corregidos automáticamente, algo muy bien valorado por todos los docentes de prueba, esta corrección además no tiene en cuenta mayúsculas ni minúsculas, ni espacios intermedios, algo que los docentes destacaron como muy positivo, y recalcaron que otras plataformas (como CREA) no ofrecen tanta facilidad de uso. Se describen más detalles de sobre la corrección de campos en la Sección 8.13.

La aparición del resto de los componentes (texto ingresado por el usuario, editor de código e intérprete) dependen de que estén habilitados en la publicación.

Si está habilitado el intérprete, el docente puede ejecutar el código del estudiante, así como editar (aunque no se guarda) el código ingresado en el editor de código (si estaba habilitado).

Algunos docentes consultados acerca de la posibilidad de editar las respuestas se manifestaron en contra y preferían que esta sea de solo lectura.

¹⁵<https://emmy-fing.herokuapp.com/users/4/publications/3/answers/43>

Emmy > Mi Perfil > Mis publicaciones > Respuestas a Determinar cantidad de raíces reales en una función cuadrática > Visualizar respuesta

Determinar cantidad de raíces reales en una función cuadrática

Respuesta:

Las funciones de segundo grado, también llamadas cuadráticas, son de la forma:

$$f(x) = ax^2 + bx + c,$$

donde el coeficiente a es distinto de 0.

Cuál es el grado de una función cuadrática? (introduce solo el número) ✓

La fórmula de bhaskara, nos permite hallar raíces de la función, y es de la forma:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

El discriminante, que es de la forma:

$$\Delta = \sqrt{b^2 - 4ac}$$

Nos da información acerca de las raíces de la función:

- Si es >0, entonces las dos raíces son reales
- Si es =0, entonces las dos raíces son iguales (llamadas raíces dobles)
- Si es <0, no tiene raíces reales.

Por ejemplo, la siguiente función, tiene ⚠

$$x^2 + 4x$$

La siguiente función tiene ✓

$$x^2 + 1$$

Por último, esta función tiene ⚠

$$\frac{8}{3}x^2$$

1) Prueba definir las funciones en MateFun y graficarlas con el comando ?grafico para verificar tus resultados

2) Usando el conjunto cantidad, define una función contRaíces que dada una terna de números reales a, b, c, que son coeficientes de una ecuación de segundo grado se supone el primero distinto de 0), contRaíces determina si la ecuación tiene 0, 1 o 2 raíces reales

CÓDIGO	GRÁFICA 1D	GRÁFICA 3D
<pre> 1 f1 :: R -> R 2 f1(x) = ... 3 4 f2 :: R -> R 5 f2(x) = ... 6 7 f3 :: R -> R 8 f3(x) = ... 9 10 cont cantidad = [Error, Unk, Dns] 11 12 contRaíces :: R X R X R -> Cantidad 13 contRaíces(a,b,c) = ... 14 </pre>		

Puedes modificar el código, los cambios no se guardarán.

[Volver al listado](#)

Figura 9.24: Ver solución ¹⁵

Validación

El presente capítulo detalla la etapa de validación de la plataforma, donde se especifican cómo se obtuvieron los usuarios, los detalles de la preparación y diseño de las pruebas y cómo estas se llevaron a cabo, culminando con un análisis de los resultados obtenidos.

10.1. Pruebas con usuarios

Para validar los diseños, las funcionalidades y la implementación se realizaron pruebas con posibles usuarios de la plataforma, estos incluyeron docentes sin conocimiento previo del lenguaje MateFun (en adelante docentes), docentes con conocimiento previo del lenguaje MateFun (en adelante docentes expertos), estudiantes de enseñanza secundaria, estudiantes de nivel terciario y maestras (en adelante familiares).

Para que los usuarios de prueba pudieran experimentar la plataforma por completo, se crearon varias cuentas de usuario con diferentes publicaciones y soluciones, (todas con contenido real), para evitar que no pudieran acceder a ciertas funcionalidades por falta de datos.

Durante el diseño de las pruebas, se confeccionó un manual y se realizaron estimaciones sobre la posible duración de las mismas, esto incluyó reproducir las instrucciones y considerar flujos en el sitio, teniendo en cuenta el tiempo que le podría llevar a un usuario realizar una acción. Estas estimaciones dieron un tiempo de 30 a 45 minutos. Finalmente, las pruebas con docentes y familiares duraron un tiempo aproximado de 40 minutos.

Se definieron tres pruebas de usuarios distintas, dependiendo de los perfiles: docentes, docentes expertos y familiares. Las pruebas realizadas fueron del tipo cualitativas y se dividieron entre moderadas presenciales (familiares), moderadas remotas (docentes) y auto-moderadas (docentes expertos).

Se contactó al Liceo IEP para realizar una invitación formal a los docentes de matemáticas, participando un total de cuatro personas de la institución. Por otro lado, los tutores a través del curso de *Matemática y Programación: potencial didáctico en el aula*¹ realizaron una invitación abierta a sus estudiantes, de

¹Entre los docentes del curso hay profesores de física, ya que se aspira a que la Emmy sea

los cuales se obtuvieron tres participantes más, por otro lado, se invitaron a dos docentes que ya conocían el lenguaje MateFun. Cuatro familiares también participaron de las pruebas, dos docentes de primaria, un estudiante de secundaria y una estudiante de educación terciaria.

En total participaron en las pruebas trece personas, entre docentes y estudiantes.

Previo a las pruebas, se les envió un manual de uso acorde a su perfil (Anexo C).

Ninguno de los participantes conocía el sitio con anterioridad y en el momento de dar la prueba se les dio el enlace y se les presentó la consigna a realizar, que incluía la letra de los ejercicios y las soluciones (Anexo D).

La decisión de dar a conocer el sitio al momento de la prueba se basó en obtener información sobre la primera interacción del usuario con el sitio.

Al finalizar la prueba, se les permitió recorrer el sitio libremente, se les entregó un cuestionario (Anexo D) y se les realizó una entrevista abierta.

Pruebas con docentes y familiares

Con los docentes de pruebas se realizaron pruebas moderadas remotas donde se les pidió compartir la pantalla, mientras que con los familiares se les realizó moderadas presenciales.

Se les pidió realizar una prueba que constaba de tres partes, primero para que se familiarizaran con el sitio se les pedía el flujo de iniciar sesión, crear una publicación (cuyo contenido estaba en el instructivo) para luego visualizarla.

La siguiente parte de la prueba consistió en crear una publicación más enriquecida, la cual requería el uso de contenido multimedia, tablas y campos de completar (provistos en el instructivo), para evaluar la curva de aprendizaje de la plataforma.

La última parte de la prueba consistía en estudiar la implementación del sistema de respuestas. Primero se les pidió listar, filtrar (por institución y código privado) y visualizar una respuesta determinada de un ejercicio, para que pudieran ver una respuesta y su corrección automática.

utilizada por profesores de ciencias naturales.

Luego se les pidió ir a la pantalla principal y se les indicó un ejercicio a responder, dándoles sus datos identificatorios (título, tema, grado, dificultad y autor).

Se observó el flujo realizado, prestándole especial atención a los filtros que decidieron usar y al camino elegido para llegar a responder el ejercicio, el cual consistía en campos de completar y múltiple opción, por lo cual al enviar su respuesta, podían ver la corrección automática, se les pidió que además identificaran su solución con su nombre, el nombre de una institución ficticia, y les dijo que debían usar un código privado determinado.

Docentes expertos

A los docentes expertos se les realizaron pruebas auto-moderadas (se realizaron por correo electrónico), esta se dividió en tres etapas.

La primera etapa consistía en publicar un ejercicio; para hacerlo, se les indicó el título, las características (tema, dificultad y grado), el contenido y la descripción. Este ejercicio contenía imágenes, campos múltiple opción y de completar, para lo cual disponían el manual provisto y el correo electrónico de los desarrolladores.

La segunda etapa consistía en responder un ejercicio, se les indicó las características (título, grado, autor y tema) para que pudieran encontrarlo en el sitio. Para resolverlo requerían del uso del intérprete y las gráficas, este tenía como objetivo probar la implementación realizada. Luego de responderlo tenían acceso a la corrección automática. Se les pidió volver a acceder al ejercicio a través del código secreto y del perfil docente.

La última etapa consistía en pedirles que publicaran en Emmy un ejercicio interactivo propio.

Familiares

Los usuarios estudiantes, realizaron las pruebas de manera presencial. Se les pidió que localizaran un ejercicio brindándoles su tema, grado, dificultad, título y autor, para poder estudiar el flujo que realizaban al buscarlo, y se les pidió responder un ejercicio precargado. Primero se debía enviar la respuesta como borrador, para luego cargarla a partir del código secreto. Una vez enviada de manera definitiva, Emmy les brindaba la corrección automática.

10.1.1. Iteraciones

Inicialmente tres docentes realizaron pruebas mientras se coordinaban las pruebas restantes, se realizaron cambios a la plataforma que reflejaban las recomendaciones que se habían recibido. Los docentes restantes realizaron la prueba con las modificaciones realizadas para evaluar las mismas. En la sección de resultados se detallan pequeños cambios sugeridos que fueron realizados al finalizar la primera iteración. Una vez que se tuvo una nueva versión, se continuó con los docentes restantes, con los familiares y con los expertos.

10.1.2. Formulario de evaluación

Se realizaron formularios de evaluación en *Google Forms*, con el objetivo de recopilar datos cualitativos y cuantitativos a partir de la opinión de los sujetos de prueba. Estos formularios se encuentran disponibles en el Anexo D.

10.1.3. Resultados

Los docentes manifestaron sentirse cómodos durante la prueba, y valoraron positivamente la plataforma junto con todas sus funcionalidades, además de calificarla como fácil de utilizar e intuitiva. También les agradó el diseño, la combinación de colores y la tipografía.

Al momento de realizar una acción, muchos intentaban replicar los pasos que seguían en sus plataformas habituales, como CREA (Contenidos y Recursos para la Educación y el Aprendizaje)² o PAM (Plataforma Adaptativa de Matemática)³, las cuales compararon con Emmy.

Frente a CREA, les gustó como estaba estructurado el sitio y su navegación, la cual permitía acceder fácilmente a cualquier ejercicio a través de un buscador con filtros, y comparando con PAM les gustó a los docentes que tuvieran la posibilidad de subir sus propios enunciados de ejercicios.

En cuanto a las opciones navegación, algunos docentes intentaron iniciar sesión con el botón *Docentes* de la página principal (el cual se había pensado originalmente para esta tarea, pero luego se sustituyó por un enlace al listado de docentes,

²<https://www.ceibal.edu.uy/en/crea>

³<https://www.ceibal.edu.uy/es/articulo/que-es-pam>

porque se pensaba que los usuarios lo utilizarían para eso). En cuanto a los filtros, ningún sujeto utilizó la funcionalidad de filtrar por subcadena hasta que se le notificó que existía esa opción.

Respecto a la corrección automática, la calificaron como esencial en una plataforma educativa de este estilo, así como muy útil. Se vieron sorprendidos de que además no tuviera en cuenta para la corrección las mayúsculas, los espacios ni los tildes, ya que en CREA deben crear varias respuestas posibles contemplando estos casos.

Se valoró también la opción de deshabilitar la corrección automática para estudiantes, configurar las publicaciones para no recibir respuestas, y permitir precargar el código MateFun en las respuestas de los estudiantes.

Previo a la realización de las pruebas, se pensaba que el manejo de códigos privados y códigos secretos daría confusión, pero los usuarios manifestaron entenderlo durante la prueba.

Los docentes manifestaron que a pesar de haber estudiado el manual, este no les aclaraba mucho el funcionamiento de Emmy sin tener acceso al sitio, y decían que lo utilizarían para resolver dudas al momento de utilizar la plataforma.

Conclusiones

Este proyecto abarcó la mayoría de las etapas de la ingeniería de software. Involucró el diseño y la implementación, donde se tomaron decisiones tecnológicas y se adoptaron implementaciones de otros proyectos para conectarse con otros servicios ya implementados, también se hizo un despliegue en un ambiente controlado para las pruebas de usuario.

Inicialmente se pretendía desarrollar un entorno virtual de actividades de aprendizaje de matemática donde se pudiera utilizar el lenguaje MateFun. Esta plataforma debía contar con una interfase, donde los profesores publicarían sus propios enunciados de ejercicios. El sitio debía contar con un enlace externo al intérprete de MateFun.

El objetivo inicial fue que los docentes publicaran ejercicios que los podría resolver cualquier estudiante, en este sentido, se podían resolver de forma independiente los ejercicios, ya que no contemplaba seguimientos de entrega.

La primera dificultad del desarrollo consistió en establecer los requisitos y el alcance de la plataforma, lo cual se hizo intercambiando, validando la especificación con los tutores y a través de comentarios de docentes allegados de los tutores. En cuanto al diseño, se prestó especial atención al posicionamiento de los elementos en pantalla y los posibles flujos, buscando que sea fácil de usar e intuitiva. Estas tareas se apoyaron en un relevamiento de plataformas similares y en base a lo aprendido en asignaturas de la carrera.

En cuanto al desarrollo, las dificultades más grandes fueron integrar la implementación en Angular del frontend de MateFun con Emmy, la cual llevó una parte importante del tiempo de desarrollo y del proyecto. Otra tarea importante fue la conexión con el servicio de intérprete de MateFun.

Las pruebas de usuarios también tuvieron su dificultad, desde la planificación hasta su culminación se dedicaron más de dos meses. Concluimos que es una etapa muy importante y enriquecedora, la cual debe prepararse con tiempo cuidando varios detalles desde la invitación a los usuarios, coordinar los tiempos y generar un clima de confianza durante la prueba en donde puedan expresar sus opiniones demostrando que se evalúa la plataforma y no al usuario.

En el diseño del flujo de las pruebas se tuvo en cuenta posibles dificultades que les podrían aparecer controlando el tiempo que les insumiría. Una tarea que se

creyó “secundaria” y resultado de gran importancia, fue precargar el sitio con otras publicaciones para que no se les presentara un sitio vacío a los usuarios, esto les permitió experimentar un sitio más auténtico.

Partiendo de la idea inicial de desarrollar una plataforma para el aprendizaje de matemáticas y con el relevamiento de plataformas similares junto con la retroalimentación de tutores y usuarios de prueba, se logró obtener una plataforma web de acceso libre que provee un entorno de actividades, pudiendo utilizar para ello el lenguaje MateFun, dibujar gráficas, figuras y animaciones todo dentro del sitio.

Además de matemática o informática, se posibilita el uso de Emmy para otras ciencias al modelar y simular soluciones de ejercicios de sus disciplinas, por ejemplo, con desafíos similares al sitio dedicado a problemas de bioinformática Rosalind.

Emmy da la posibilidad de acceder a los contenidos de manera jerárquica y estructurada, pudiendo acceder a través de los distintos temas, utilizando filtros como dificultad o grado, se puede seleccionar no solo los ejercicios si no también el autor de estos, sin necesidad de estar registrado en el sitio. Se permite al publicar ejercicios embeber imágenes, vídeos se pueden utilizar campos para completar y campos de múltiple opción, con la posibilidad de dar corrección automática, dejar código precargado para escribir programas que se puedan cargar en el intérprete. Los docentes tienen la posibilidad de utilizarla en un ambiente de clase, ya que les permite realizar un seguimiento de los estudiantes y sus tareas, mientras que los administradores tienen control sobre los usuarios, los temas y las publicaciones.

Se lograron llevar a cabo pruebas de usuario en paralelo con otras tareas, que, aunque llevaron más tiempo del previsto (por coordinación de horarios) brindaron información valiosa no solo para validar y corregir la plataforma, si no para futuros desarrollos, así como también sirvieron para sentar las bases de nuevas pruebas de usuarios de futuros proyectos. En cuanto a la plataforma, los usuarios manifestaron que es fácil de usar e intuitiva, donde además fue muy bien vista y considerada de utilidad por los expertos en MateFun.

Faltó la integración con el frontend de MateFun en Angular, específicamente, el intérprete, editor de código y las gráficas, que por desconocimiento de la tecnología y por no haber podido integrarlo con el framework de Ruby On Rails se terminó traduciendo todo el código a JavaScript, lo cual permitió tener la mayoría de las funcionalidades disponibles en la plataforma. En cuanto al intérprete, hubiera sido ideal almacenarlo en el propio servidor de la plataforma, en vez de consumirlo como

servicio, pero su alcance hubiera consumido todo el tiempo de implementación destinado a la plataforma.

Esta plataforma establece una base para un futuro desarrollo, el lenguaje base es muy amigable con los usuarios y su curva de aprendizaje es muy buena, además, el framework elegido sigue convenciones de desarrollo que facilitan la implementación, los cambios y el prototipado. El proyecto cuenta con una completa cobertura de pruebas en backend y sigue las convenciones de sintaxis de código, lo que facilita la comprensión.

En cuanto a las pruebas de usuario, este proyecto aporta una base para futuras pruebas, así como ideas para nuevos desarrollos.

Las pruebas con usuarios también brindaron grandes ideas para el futuro de la plataforma, como más opciones de corrección, más tipos de publicaciones, manejo de usuarios y más opciones de calificación y seguimiento de entregas. Aunque nunca estuvo en la especificación inicial, y se decidió desde un principio dejar el registro de estudiantes fuera del sitio, si se quiere que Emmy se use en el aula (como a muchos docentes de pruebas les gustaría) se debería incluir el registro de estudiantes.

Si bien algunos profesores de Física usan MateFun, las pruebas de la plataforma se llevaron a cabo con profesores de matemática. Queda planteada la posibilidad de hacer pruebas con profesores de ciencias naturales e incluir sus comentarios y sugerencias en futuros desarrollos del sitio.

Futuras líneas de desarrollo

Emmy representa un MVP que surgió del estudio y priorización de requerimientos con los tutores, en este sentido, presenta una completa funcionalidad de publicaciones y respuestas, una implementación del intérprete y editor de código MateFun, con funcionalidad básica de gráficas 2D y 3D. En cuanto al sistema de usuarios, solo los docentes y administradores pueden registrarse.

Se propone en esta sección futuros cambios para Emmy, que por motivos de alcance y restricciones de tiempo no se pudieron hacer, donde algunos requerirían de un proyecto de grado entero para su realización.

Por otro lado, muchas de estas nuevas funcionalidades y cambios surgieron de las entrevistas con los usuarios de prueba, algunos de estos cambios, por su extensión y por la etapa del proyecto en la que surgieron, fueron imposibles de implementar, mientras que los más importantes e imprescindibles (sobre todo de usabilidad) se implementaron durante las iteraciones de las pruebas de usuarios y al terminar las mismas.

12.1. Nuevas funcionalidades

12.1.1. Disciplinas

El desarrollo del MVP del sitio se basó en una plataforma general para matemática, incluir a otras ciencias, implicaría crear disciplinas y asociarlas a los temas.

12.1.2. Internacionalización

Rails soporta internacionalización con su gema *I18n* [14], y su uso es bastante sencillo. Alcanza con definir un archivo YML y traducir las etiquetas correspondientes, las cuales luego se utilizan en las vistas HTML. Se han traducido los modelos y algunos de sus atributos ¹.

Para terminar de implementar el manejo de idiomas, habría que reemplazar

¹Por convención los nombres de los modelos y atributos van en inglés.

todos los textos de la plataforma por etiquetas, y luego definir el texto correspondiente.

12.1.3. Botón de ayuda

Un docente comentó que PAM tenía la posibilidad de reportar ejercicios que tuvieran erratas. Se propone agregar un botón de ayuda al sitio para reportar errores en ejercicios así como proponer nuevas funcionalidades.

12.1.4. Agregar filtros por fecha

Actualmente no se pueden filtrar por fecha las publicaciones ni las respuestas, así como tampoco pueden los administradores establecer un rango a las estadísticas del sitio. Esto se puede implementar fácilmente con formularios y un elemento *datepicker*

12.1.5. Filtros y ordenación

Emmy no permite filtrar y ordenar por columna al mismo tiempo. Cambiarlo, supone concatenar la petición desde el Frontend, y sustituir el *helper* de ordenación por funcionalidades en la acción del controlador.

Para la paginación se utiliza la gema *Will paginate*, aunque existen versiones más modernas y de mejor rendimiento, como *Kaminari*.

12.1.6. Subir imágenes

El almacenamiento de imágenes se hace convirtiéndolas en base64 y se almacenan en base de datos junto con el contenido de la publicación. Una mejor implementación sería utilizar *Active Storage*² y configurar un servicio de nube para el almacenamiento de imágenes [1].

12.1.7. Enviar respuesta otro docente

Pensando en un sitio colaborativo surge a partir de las encuestas de usuario la idea de que un estudiante pueda resolver un ejercicio que será evaluado por

²https://edgeguides.rubyonrails.org/active_storage_overview.html

un docente distinto al autor de la publicación. Actualmente el estudiante puede resolver un ejercicio de cualquier docente, y enviar el código secreto al docente que desee. Se presenta el inconveniente de que el autor de la publicación elimine la misma o la solución enviada.

12.2. Sistema de usuarios

Como ya mencionado en el Anexo A, se dejó afuera el registro de estudiantes por no considerarse parte del MVP, y porque Emmy surgió como un espacio abierto para poder ser utilizado por cualquier estudiante (al estilo del sitio relevado NRICH, en Sección 5.1.1) y no como una plataforma cerrada de uso exclusivo de aula.

A partir de la idea inicial, se fueron agregando funcionalidades para el manejo de clase y seguimiento de estudiantes (con la visualización de tareas en progreso y recuperación de entregas), por lo que futuras mejoras al sitio deberían mantener esta dualidad entre plataforma abierta y plataforma de uso cerrado en el aula, agregando la opción de registrar estudiantes, aunque sin perder la capacidad de responder publicaciones sin estar registrado. A esto debería agregarse la opción de que los docentes puedan elegir si sus publicaciones pueden ser respondidas solamente por usuarios registrados.

Un docente planteó que los estudiantes tuvieran sesión en el sitio, para asegurarse que cada estudiante realice una única entrega y que resolvió la propuesta en un único intento. Para esto, los estudiantes deberían registrarse en el sitio con su número de cédula de identidad.

También surgió de las pruebas de docentes el requerimiento de poder crear grupos de clase, donde los estudiantes puedan tener un docente asignado, a su vez, estos docentes tendrían que poder ver cuáles estudiantes han realizado determinadas tareas.

12.2.1. Calificaciones

En cuanto a la calificación surgieron varias propuestas, una escala sobre 12, una escala sobre 100 y una escala semáforo (útil para evaluaciones diagnósticas a principio de año).

Según comentarios de un docente, utilizar una escala de semáforo es útil para realizar pruebas diagnósticas, estas pruebas suelen realizarse al comenzar el año lectivo ya que según el resultado es como planifican el dictado del curso y pueden evaluar en que temas se necesita realizar un repaso previo. El docente utilizaría el sitio para cargar pruebas diagnósticas que se corrigen solas y que suelen ser similares todos los años. Para este caso el resultado de las pruebas diagnósticas, debe ser visible por grupo, ya que al docente le interesa conocer el resultado global, de todo su grupo y no por estudiante.

Se pidió también que Emmy calcule automáticamente una nota global para la respuesta, con la posibilidad de sobrescribirla.

Como nuevo requerimiento surgió que cada docente tenga un resumen de cada ejercicio publicado, permitiendo así al docente conocer cómo les fue a todos los estudiantes. También se pidió tener un resumen global de notas, tanto por ejercicio como por área. Figura 12.1 muestra un listado de calificaciones que un docente de pruebas nos permitió obtener de sus estudiantes en PAM.

Objetivo de aprendizaje	Capítulo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
Propiedad distributiva y geometría	-	68				76										32	0	85		56			
Ecuaciones 18/8	-	90	75			100		75	100							55			85	70			
Ecuaciones 1	-	62		100		75			100										75	100	96	63	
Ecuaciones 2	-	86	100	78				78	58							50			97	75	83	47	
Ecuaciones 3	-	74	92					74	0										94	74			
operaciones con enteros	-	89	88	88		95	88		97	100						71	92	6	89	73		100	
Triángulos: Relación entre ángulos y lados	-	85					88	82								44	59		85	50	53	62	24
Operaciones con Fracciones	-	65					98	87								48			85	70		64	
Triángulos	-	87					73	77								33	23				57	73	53
Operaciones combinadas enteros	-	74	87		88	76	93	82	90							78	94		85	76	94	85	
Operaciones combinadas con fracciones	-	65					89	74								74	27			71		40	
Ecuaciones prueba y error	-	74			88			79											88	90		100	
Ecuaciones método de la balanza	-	87	95				95	76											67	95		68	
Armando expresiones y ecuaciones.	-	75						72	72							38	52		95	50			
Armando expresiones algebraicas	-	78						83											70	87		67	
Operaciones con monomios	-	87				91		85								72		57	100	70			
Producto de Monomios	-	91				91	100	84								59	53	78	100		84		
Ecuaciones Presencialidad	-	74				74	100									58	74		100		68		
Ecuaciones Presencialidad 1	-	78					89												87	80	76	41	
Operaciones con monomios 1	-	83	81			81	86		95										50	86	90	95	
Propiedad distributiva	-	68					80		80	30									80	68			

Figura 12.1: Calificación por área de los estudiantes de la plataforma PAM.

12.3. Publicaciones

Se podría agregar estadísticas similares a las que ven los usuarios administradores a cada perfil docente, donde pudieran ver la cantidad de respuestas a sus publicaciones, el tiempo que les lleva completarlas y la tasa de éxito.

12.3.1. Tipos de publicaciones

Se propuso tener la opción de agregar un cronometro que el docente puede habilitar en un ejercicio para utilizar la plataforma en el salón de clases para así conocer cuánto tiempo le llevo a cada estudiante realizar el mismo. La consigna de la publicación no se revelaría al estudiante hasta el momento que le docente la habilite, ya sea estableciendo una fecha y hora o de manera manual.

Una propuesta similar surge agregando un temporizador para evaluar una prueba en un salón de clases, donde el tiempo está determinado previamente.

Se sugirió también agregar otra clase de publicaciones, que consista en una colección de otras publicaciones.

12.3.2. Editor de texto

Se consideró la elección del editor de texto TinyMCE (Sección 8.12) como una buena opción, ya que tiene gran capacidad para personalizarlo, continuo desarrollo por una empresa y buena documentación con ejemplos.

Este editor sin embargo, carece de la posibilidad de utilizar código \LaTeX para escribir ecuaciones (algo de baja prioridad según la validación de los tutores). Se estudió que existen dos extensiones que podrían utilizarse para cumplir con esta funcionalidad ³ ⁴.

También se propone estudiar un modo nocturno, ya que uno de los docentes de prueba tenía activado el modo en su computadora.

³<https://github.com/moonwave99/TinyMCELaTeXPlugin>

⁴<https://github.com/imspent/tinymce-latex>

12.3.3. Campos interactivos

Se pudo implementar campos de completar y campos de múltiple opción, lo que supone una excelente base para futuros cambios (ya que no se pudo encontrar ejemplos similares antes de implementarlo). A estos campos se les podría agregar otros, como poder seleccionar más de una opción o campos verdadero-falso, la documentación de Moodle contiene más ejemplos de campos que son utilizados en un ambiente académico [13].

Por otro lado, a pesar de que la interfaz para ingresar las opciones es bastante simple, en las evaluaciones algunos docentes manifestaron ser un poco confusas (sobre todo hasta que pudieron utilizarlas en la práctica), se ingresa una opción por renglón y en el caso de las preguntas múltiple opción se le debe agregar un carácter de @. Se propone una interfaz como la mostrada en Figura 12.2

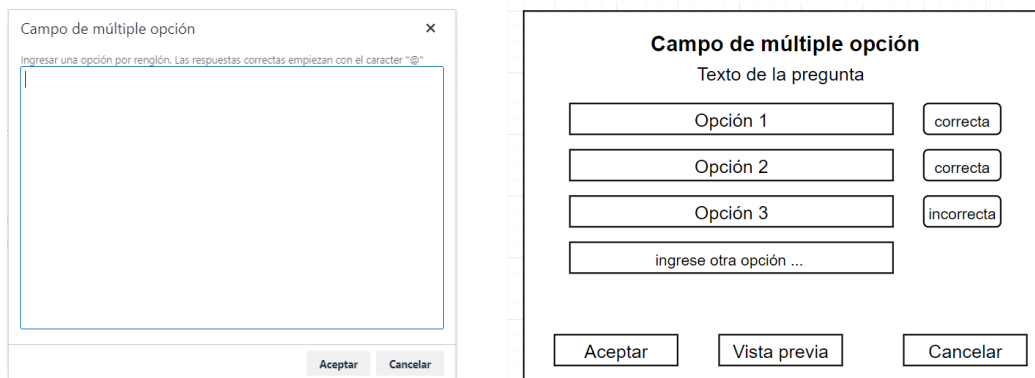


Figura 12.2: Implementación actual (izquierda) y propuesta (derecha).

Algunas plataformas, como Moodle, permiten además establecer respuestas no solo como texto plano (donde se evalúa la igualdad), si no también establecer formulas, por ejemplo, decir que las respuestas posibles son $10 \pm 0,5$.

12.3.4. Respuestas

En el listado de respuestas se podría agregar la posibilidad de marcar respuestas como ya visitadas.

Corrección automática

La corrección automática de campos implementada no tiene en cuenta las mayúsculas, minúsculas, espacios ni tildes, algo que se valoró muy positivamente por los docentes y que además no implementaban sus plataformas (como CREA).

Esta corrección se realiza del lado del servidor (para no compartir las respuestas con el frontend) cada vez que se visualiza una respuesta, una mejor solución sería almacenar la corrección de cada campo interactivo en la base de datos, simulando una especie de *cache*, y cada vez que se visualice la misma, detectar si han cambiado las preguntas o sus respuestas, volviendo a computar la corrección solo en ese caso. De esta forma, se evitaría consumir recursos en corregir las mismas preguntas.

Por otro lado, la corrección automática (Figura 12.3) no muestra las opciones que estaban disponibles al estudiantes, ni cual era la respuesta correcta. Se propone habilitar la opción de mostrarle al estudiante cual era la respuesta esperada.

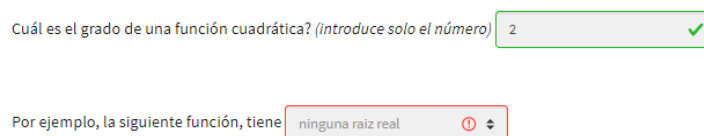


Figura 12.3: Corrección automática.

El sitio no permite sobrescribir la corrección para una respuesta particular (para hacerlo, hay que editar la publicación y modificar las respuestas correctas). En las pruebas de usuarios se planteó la posibilidad de sobrescribir la corrección, tomando como válida la respuesta del estudiante incluso cuando esta no estaba dentro de las posibles (ya que a veces lograban llegar a resultados equivalentes), se pidió también considerar como válidas cualquier tipo de permutación de la respuesta (sin tener que ingresarlas todas), ya sea teniendo un botón que desordene y se considere como válida alguna de ellas o que se resuelva de forma interna. Emmy tampoco permite definir respuestas como conjuntos (todas las respuestas se toman literal manteniendo el orden de los elementos).

También se planteó modificar la nota cuando el estudiante realiza una nueva entrega, pero no modificar la respuesta de la entrega anterior del estudiante, se pretende tener un historial. Para mantener este historial se debe permitir realizar comentarios en cada una de las entregas, pero los docentes comentaron que no de-

berían modificar la respuesta original del estudiante, quisieran que quede separado la respuesta del estudiante con los cambios del docente.

Esto se basó en que los docentes quisieran ver las entregas anteriores para saber en qué se equivocaron previamente. Actualmente en el sitio el docente tiene acceso a la dirección donde puede visualizar como está el estudiante resolviendo el ejercicio (aun cuando la solución del estudiante se encuentra como borrador y no ha sido enviada) puede ver todas las entregas intermedias, pero estará visualizando siempre la última versión.

Otro requisito es dar una devolución a la entrega junto con una nota junto con comentarios, donde en caso de ser necesario informarle en que tema deberá profundizar. Se pidió además permitir ponderar cada parte del ejercicio. En las pruebas de usuario los docentes plantearon que para un estudiante es importante ver una calificación ya que los motiva a realizar más ejercicios.

Manifestaron que les gustaría agregar una nota a la respuesta del campo, no pudiendo el docente modificar la respuesta del estudiante, pero si permitir agregar una aclaración al lado (consideraron importante que el docente no pueda cambiar la respuesta del estudiante).

Campos de intérprete

Un tipo de campo que no se llegó a implementar, es el campo de intérprete, donde los estudiantes puedan introducir y ejecutar código MateFun y donde el docente pueda escribir las respuestas posibles.

12.4. Editor de código MateFun e intérprete

Aunque la implementación del intérprete y editor de código contiene todas las funcionalidades, hay detalles de diseño que no se llegaron a implementar.

La Figura 12.4 muestra las sugerencias de código, iconos de advertencia y error en el lateral izquierdo así como un indicador de número de columna y línea en la parte inferior. De estos detalles visuales se pudo implementar el más notorio, los colores de sintaxis del lenguaje.

Cómo se mencionó en la Sección 8.10, esta implementación se realizó convirtiendo el proyecto de Angular en JavaScript, por limitaciones tecnológicas, restric-

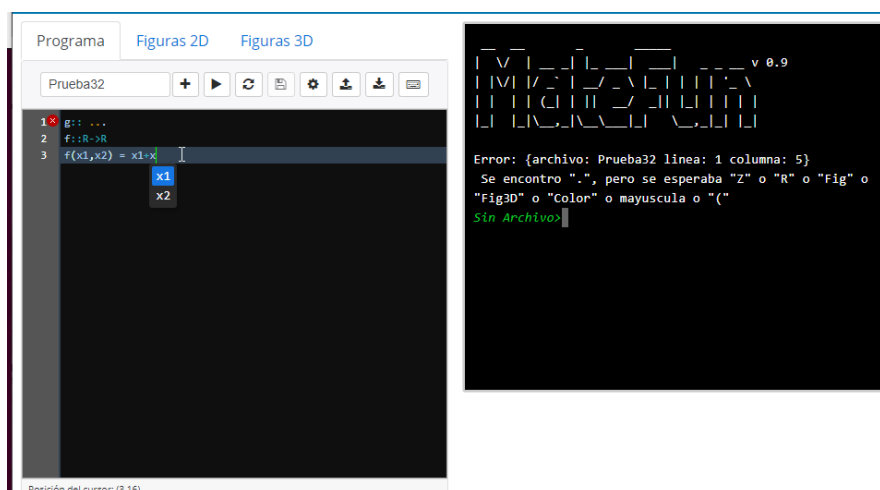


Figura 12.4: Intérprete del sitio MateFun.

ciones de tiempo y desconocimiento del lenguaje. Un proyecto de grado, donde el equipo tenga conocimiento de Angular podría traer toda la interfaz del entorno MateFun a la plataforma.

Por último, se considera como la mejor opción usar el intérprete de MateFun de manera local, para no depender de la conexión a otros sitios. Esto implicaría implementar comunicación asíncrona entre la consola de Frontend hacia el intérprete, lo que requiere también la administración de hilos para atender a los usuarios.

Historial de respuestas

Para que los docentes tengan un historial de lo que ha realizado el estudiante, podría almacenarse en el sitio todos los envíos intermedios que ha realizado. Esto se podría extender al código fuente de MateFun, donde se almacenarían también los códigos que ha cargado.

12.5. Entorno de gráficas

La implementación de gráficas 2D solo permite graficar una función a la vez, y solo se logró poder graficar en el dominio de los Reales. Se podría estudiar la implementación para lograr esta funcionalidad, pero se considera mejor una implementación que incluya todo el entorno en Angular a la plataforma.

12.6. Extensión con MateFun Infantil

Se podría extender el entorno de MateFun actual, al implementado en el proyecto de grado *MateFun Infantil* [23]. De esta forma, la plataforma podría tener disponibles los dos entornos, el entorno de texto de MateFun actual y el entorno de bloques del infantil. Cuando el docente cree la publicación, este podría elegir en qué entorno se resolverá.

Glosario

API Interfaz de programación de aplicaciones. 28

Backend Parte del software con la cual procesa la entrada desde el frontend y no interactúa con los usuarios. 28, 29, 38

Breadcrumb Técnica de navegación utilizada en la interfaz de usuario para indicar su ruta de navegación. 42

CDN Red de distribución de contenidos. 39

CSS Hoja de estilos en cascada. 38

código privado Identificador acordado entre el docente y sus estudiantes, su utilidad es facilitar la identificación de soluciones. Un estudiante al enviar su solución puede ingresar opcionalmente un código, el docente al acceder al listado de soluciones puede filtrarlas, obteniendo así las soluciones de un grupo en particular.. 21

código secreto Identificador único en la plataforma, es proporcionado por esta al estudiante cuando responde una publicación. Con este código el estudiante puede volver a cargar su borrador para continuar el trabajo o visualizar la respuesta una vez enviada.. 21, 42, 56, 68

DRY Don't repeat yourself (No repitas). 30

Emmy Ejercicios de Matemáticas en MateFun ya!. 20

Fat model, Skinny Controller Patrón estructural, consiste en mover la lógica de negocios y de base de datos fuera de los controladores, dejando esta última en los modelos. 32

Framework Entorno de trabajo que agrega al lenguaje base funcionalidades con el fin de reutilizar código, para simplificar y acelerar el desarrollo. 29, 30

Frontend Parte del software con la cual interactúan los usuarios. 28, 32, 38, 39, 75, 82

gema Programas y librerías del lenguaje Ruby. 31, 32, 34, 36, 38, 39, 41, 42, 74

HTML Lenguaje de marcado de hipertexto. 32, 36–38, 40, 74

IDE Entorno de desarrollo integrado. 8

Lexema Unidad de significado léxico que se relaciona con un conjunto de palabras a través de la flexión lingüística. En bases de datos son palabras que han sido normalizadas para unificar las variantes de una misma palabra. 34

Merge-request Peticiones para unir una línea de código a otra. 32

modelo Un modelo es una clase de Ruby que representa una tabla en la base de datos. 31, 37

MVC Modelo, vista, controlador. 30, 32

MVP Mínimo producto viable. 19, 32, 74

NIST Instituto Nacional de Estándares y Tecnología. 38

refactorización Reestructurar código existente sin alterar su funcionalidad, para reducir su extensión, complejidad, aumentar tu mantenimiento y su legibilidad. 30

sanitización Limpieza y filtrado de elementos para evitar un uso indeseado o malicioso. 38

STI Herencia en única tabla. 31

ts vector En PostgreSQL, es una lista ordenada de lexemas. . 34

visitantes Personas sin sesión iniciada. 31, 41

vista Representación final de una petición. 30, 32, 37

YML Formato de serialización de datos. 74

Bibliografía

- [1] Active storage overview. https://edgeguides.rubyonrails.org/active_storage_overview.html. Accedido: 05-2021.
- [2] Activesupport::timezone. <https://api.rubyonrails.org/classes/ActiveSupport/TimeZone.html>. Accedido: 05-2021.
- [3] Breadcrumb navigation increasingly useful. <https://www.nngroup.com/articles/breadcrumb-navigation-useful/>. Accedido: 05-2021.
- [4] Breadcrumbs: 11 design guidelines for desktop and mobile. <https://www.nngroup.com/articles/breadcrumbs/>. Accedido: 05-2021.
- [5] Coderunner. <https://moodle.org/plugins/qtype/coderunner/>. Accedido: 05-2021.
- [6] Desktop screen resolution stats uruguay. <https://gs.statcounter.com/screen-resolution-stats/desktop/uruguay>. Accedido: 05-2021.
- [7] Digital identity guidelines. <https://pages.nist.gov/800-63-3/sp800-63b.html>. Accedido: 05-2021.
- [8] Grid system. <https://getbootstrap.com/docs/4.0/layout/grid/>. Accedido: 05-2021.
- [9] Hardware sirio 2021. <https://www.ceibal.edu.uy/es/articulo/hardware-sirio-2021>. Accedido: 07-2021.
- [10] Matefun - participantes. <http://d11.com.uy/web/matefun/dev03/equipo-responsable/>. Accedido: 06-2021.
- [11] Miga de pan (informática). [https://es.wikipedia.org/wiki/Miga_de_pan_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Miga_de_pan_(inform%C3%A1tica)). Accedido: 05-2021.
- [12] Mobile screen resolution stats uruguay. <https://gs.statcounter.com/screen-resolution-stats/mobile/uruguay>. Accedido: 05-2021.
- [13] Question types. https://docs.moodle.org/311/en/Question_types. Accedido: 06-2021.

- [14] Rails internationalization (i18n) api. <https://guides.rubyonrails.org/i18n.html>. Accedido: 05-2021.
- [15] Short-term memory: Up to 7 items, but highly volatile. https://thebrain.mcgill.ca/flash/capsules/experience_jaune03.html#:~:text=Short%2DTerm%20Memory%3A%20Up%20to%207%20Items%2C%20But%20Highly%20Volatile&text=Many%20psychology%20experiments%20have%20shown,%2C%20depending%20on%20the%20individual. Accedido: 05-2021.
- [16] Vlp features. <https://vpl.dis.ulpgc.es/index.php/about/features>. Accedido: 05-2021.
- [17] Vlp support. <https://vpl.dis.ulpgc.es/index.php/support>. Accedido: 05-2021.
- [18] What is recaptcha. <https://www.google.com/recaptcha/about/>. Accedido: 05-2021.
- [19] will paginate documentation. https://github.com/mislav/will_paginate/wiki. Accedido: 05-2021.
- [20] Gonzalo Cameto et al. Proyecto de grado. page 81.
- [21] Sylvia Da Rosa et al. Mathematics and matefun, a natural way to introduce programming into school. pages 1–10.
- [22] Martín Méndez Gonzalo Cameto. Matefun ide web. 2017.
- [23] Felipe Parodi Lucía Labat. Matefun infantil, versión web. 2019.
- [24] Maximiliano Poses Nicolás Chiaramello, Lucia Etchart. Matefun infantil, versión andriod. 2019.
- [25] Niels Provos and David Mazier. A futureadaptable password schem.
- [26] Ian Sommerville. *Software Engineering, Tenth Edition*. Pearson, 2015.
- [27] Marcos Viera Sylvia Da Rosa. Matefun - manual. pages 1–6.

Anexos

Documento de requisitos

La especificación de requerimientos para la primer versión se realizó buscando determinar el mínimo producto viable. Se dejan afuera de este *MVP* las funciones de registro de estudiantes y todas las funcionalidades de este tipo de usuarios así como el sistema de calificación de publicaciones. Logrando así una versión horizontal y simplificada para luego iterar sobre las funcionalidades.

La especificación está descrita utilizando historias de usuario.

Nota: Para simplificar el documento y facilitar su comprensión, algunas de las historias aquí descritas son del tipo épicas, que se separaron en historias de usuario más simples para el desarrollo.

A.1. Generales

Listar temas *Como usuario de la plataforma quiero poder ver una lista de temas.*

Criterios de aceptación:

Se deberán listar todos los temas que tengan al menos una publicación en estado publicado

Elegir un tema lista las publicaciones clasificados bajo ese tema.

Listar docentes *Como usuario de la plataforma quiero poder ver una lista de docentes.*

Criterios de aceptación:

Elegir un docente muestra sus publicaciones en estado publicado.

No deberán mostrarse en el listado docentes que no han publicado ningún publicación.

Listar grados *Como usuario de la plataforma quiero ver una lista de grados que contengan publicaciones.*

Criterios de aceptación:

Elegir un grado lista las publicaciones clasificados bajo ese grado.

No deberán mostrarse grados que no contengan publicaciones.

Listar publicaciones *Como usuario de la plataforma quiero poder ver una lista de publicaciones.*

Criterios de aceptación:

Se debe poder filtrar por palabra clave, grado, nombre de profesor, nivel de dificultad y tema .

Los listados deben ordenarse por fecha descendiente.

Solo se deben listar aquellas publicaciones en estado publicado. Si se accede al listado a través de un tema, grado o docente, no debe mostrarse el filtro correspondiente Desde un publicación se puede acceder a otros publicaciones que contengan mismo: grado, tema, dificultad o docente.

A.2. Usuarios

Iniciar Sesión *Como usuario de la aplicación quiero poder iniciar sesión para hacer uso de ella*

Criterios de aceptación:

El usuario debe haber quedado identificado en el sistema a partir del nombre de usuario.

El sistema debe direccionar al usuario a su perfil.

Si el usuario intenta ingresar con credenciales inválidas se debe mostrar un mensaje de error.

Se debe poder activar y desactivar para visualizar o no la contraseña.

Se debe completar un *reCAPTCHA* para validar que el usuario es humano

Cerrar Sesión *Como usuario quiero poder cerrar la sesión de la aplicación en cualquier momento para salir de la misma.*

Criterios de aceptación:

El sistema lleva al usuario a la pantalla principal.

Modificar perfil *Como usuario quiero modificar mis datos*

Criterios de aceptación:

Se debe permitir modificar: el correo, nombre completo, institución (excepto administradores) y contraseña.

Se debe confirmar los cambios indicando la contraseña actual.

A.2.1. Docentes

Realizar publicación *Como docente quiero poder crear una publicación.*

Criterios de aceptación:

El editor de texto debe contar con opciones de formato, estilo, diseño.

Cada publicación debe tener asignado: título, un grado, dificultad, y tema.

Las publicaciones deben tener una breve descripción.

Se permite embeber imágenes, vídeos, espacios para completar.

Se debe permitir establecer opciones de publicación: determinar si al resolver un publicación se puede enviar respuestas, ver la corrección automática, habilitar el editor de texto, habilitar el intérprete de MateFun y el editor de código.

La opción: corrección automática debe estar disponible solo si el publicación admite el envío de respuestas.

La opción: editor de código MateFun, debe estar disponible solo si el publicación permite utilizar el intérprete.

Se debe permitir al docente ingresar código MateFun que será precargado en el editor de código de los estudiantes.

Se debe poder guardar como borrador si aún no ha sido publicada.

Las publicaciones sin publicar solo deben mostrarse en el perfil docente, y no pueden ser accedidas sin estar autenticado como el autor.

Modificar publicación *Como docente quiero poder modificar una publicación.*

Criterios de aceptación:

Se permiten los mismos criterios de la historia: *Realizar publicación.*

Borrar publicación *Como docente quiero poder borrar un publicación*

Criterios de aceptación:

Se realiza un borrado lógico.

Se borran todas las respuestas relacionadas.

Listar mis publicaciones *Como usuarios quiero ver todas mis publicaciones.*

Criterios de aceptación:

Se listan todas las publicaciones del docente.

En cada publicación debe indicarse si la misma fue publicada o no.

Se debe permitir filtrar por: contenido (título o cuerpo), tema, grado y dificultad.

El listado debe estar ordenado cronológicamente de forma descendente por fecha de última modificación.

Listar soluciones *Como docente, quiero ver un listado de soluciones correspondientes a una publicación.*

Criterios de aceptación:

Se debe permitir filtrar por nombre, institución y código privado.

Se debe permitir ordenar por nombre, institución, código privado y fecha de la última modificación.

Se debe filtrar por subcadena.

Se listan todas las soluciones: publicadas y no publicadas.

Para cada solución del listado debe indicarse si la misma ha sido entregada o está en proceso (borrador).

Ver una solución *Como docente, quiero ver la solución de una publicación.*

Criterios de aceptación:

Se debe visualizar el nombre, institución, código privado ingresado por el estudiante y fecha/hora de entrega.

Se debe mostrar el código secreto.

Se debe realizar la corrección automática, de todos los campos interactivos que tengan una respuesta definida.

Se debe permitir al docente ejecutar código MateFun del estudiante.

Se debe poder modificar el código MateFun, pero no deben guardarse las modificaciones para no alterar sus respuestas.

Borrar soluciones por lote *Como docente quiero eliminar varias soluciones.*

Criterios de aceptación:

Se debe poder seleccionar varias soluciones para eliminar.

Se realiza un borrado físico.

Borrar una publicación *Como docente quiero eliminar una publicación.*

Criterios de aceptación:

Se realiza un borrado físico.

A.2.2. Administradores

Ver resumen *Como usuario administrador quiero ver un resumen*

Criterios de aceptación:

Se debe mostrar un resumen de temas, publicaciones y respuestas.

Mostrar los temas con mayor cantidad de publicaciones.

Se debe mostrar la cantidad de publicaciones por dificultad y por fecha.

Se debe mostrar la cantidad de respuestas por fecha.

Registrar usuario *Como administrador quiero registrar un usuario.*

Criterios de aceptación:

Los campos obligatorios son: nombre completo, correo y contraseña.

Si se está creando un usuario docente se debe poder ingresar la institución.

La contraseña debe tener largo entre 8-64, mayúsculas, minúsculas y símbolos.

Debe elegirse si el usuario será un administrador o un docente.

Eliminar usuario *Como usuario administrador quiero poder borrar usuarios de la plataforma.*

Criterios de aceptación:

Se realiza un borrado lógico.

Si el usuario a eliminar es docente, se borran todas sus publicaciones.

Crear un tema *Como usuario administrador quiero poder crear una tema de publicación.*

Criterios de aceptación:

Se crea un nuevo tema.

Se debe permitir agregar una descripción al tema (opcional).

Borrar tema *Como usuario administrador quiero poder borrar una tema de publicación.*

Criterios de aceptación:

Se realiza un borrado lógico.

Se le quitará este tema a las publicaciones que lo tuvieran.

Las publicaciones pasarán a tener como tema: "Sin especificar".

Crear nueva contraseña para un usuario. *Como usuario administrador quiero crear una nueva contraseña para un usuario.*

Criterios de aceptación:

Se debe especificar el usuario.

La contraseña debe tener largo entre 8-64, mayúsculas, minúsculas y símbolos.

Listar administradores. *Como usuario administrador quiero listar todos los administradores.*

Criterios de aceptación:

Listar todos los administradores.

Se debe poder filtrar por nombre, nombre de usuario o correo.

Se debe poder filtrar por subcadena.

Se debe poder ordenar por nombre, nombre de usuario o correo.

Filtrar publicaciones en perfil administrador. *Como usuario administrador quiero filtrar las publicaciones.*

Criterios de aceptación:

Se deben poder filtrar publicaciones en el perfil, por título y nombre de autor.

Listar temas en perfil administrador. *Como usuario administrador quiero buscar temas según el nombre.*

Criterios de aceptación:

Se debe poder filtrar por nombre de tema.

Se debe poder ordenar por nombre o cantidad de publicaciones.

Listar docentes en perfil administrador. *Como usuario administrador listar docentes en panel de administración.*

Criterios de aceptación:

Se deben poder filtrar a todos los docentes por nombre, correo, nombre de usuario e institución.

Se deben poder filtrar todos los campos por subcadena.

Se debe poder ordenar por: nombre, correo, nombre de usuario e institución.

A.3. Estudiantes

Resolver publicación *Como estudiante quiero responder un publicación.*

Criterios de aceptación:

Se debe poder acceder a la letra de la tarea.

Debe ingresarse nombre y apellido. Opcionalmente se puede ingresar la institución y código privado que permite identificar la entrega.

En caso de estar habilitado se puede resolver una tarea utilizando el entorno de MateFun y editor de texto.

En caso de estar habilitado se debe poder compilar código MateFun.

Se debe poder guardar la respuesta como borrador para continuarla luego.

Solo se podrá enviar o guardar la respuesta si la publicación admite respuestas.

Al enviar o guardar la respuesta, se debe obtener el código secreto de recuperación.

Continuar publicación a partir de un código *Como estudiante quiero continuar con la resolución de un publicación.*

Criterios de aceptación:

Se debe poder cargar la solución de un publicación a partir del código secreto para continuar con su modificación.

Cuando se realicen modificaciones el código deberá mantenerse.

Ver solución enviada *Como estudiante quiero visualizar una tarea entregada anteriormente.*

Criterios de aceptación:

Utilizando el código secreto el estudiante puede acceder a entregas realizadas anteriormente.

En caso de ser un publicación con corrección automática, el estudiante deberá ver la corrección de todos los campos de completar con respuestas definidas.

No se debe poder modificar ningún campo.

Bocetos del sitio

login



MateFun

Image cap

Navegar por tema

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Image cap

Navegar por grado

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Image cap

Navegar por profesor

This is a wider card with supporting text below as a natural lead-in to additional content.

Image cap

Navegar por...

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.



MateFun

Navegar por tema

[Index](#) > [Navegar por tema](#)

Image cap

Números primos

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Image cap

Funciones lineales

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Image cap

Sistemas de ecuaciones

This is a wider card with supporting text below as a natural lead-in to additional content.

Image cap

....

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.



Lista de ejercicios de números primos

- molestiae quas vel sint commodi repudiandae consequuntur voluptatum laborum numquam
- blanditiis harum quisquam
- eius sed odit fugiat iusto fuga praesentium
- optio, eaque rerum! Provident similique accusantium nemo autem. Veritatis
- obcaecati tenetur iure eius
- earum ut molestias architecto
- voluptate aliquam nihil, eveniet aliquid culpa
- officia aut! Impedit sit sunt quaerat, odit,
- tenetur error, harum nesciunt ipsum debitis
- quas aliquid. Reprehenderit, quia. Quo neque error repudiandae fuga?
- Ipsa laudantium molestias eos sapiente
- officii modi at sunt excepturi expedita sint?
- Sed quibusdam recusandae alias error
- harum maxime adipisci
- amet laborum
-



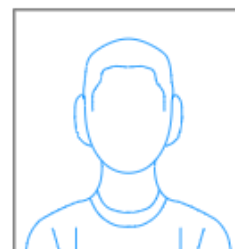
Juan
Pablo
martinez



MateFun

Ejercicios publicados

List group item heading	3 days ago
Donec id elit non mi porta gravida at eget metus. Maecenas sed diam eget risus varius blandit. Donec id elit mi porta.	
Ver solución	editar
List group item heading	3 days ago
Donec id elit non mi porta gravida at eget metus. Maecenas sed diam eget risus varius blandit. Donec id elit mi porta.	
Ver solución	editar
List group item heading	3 days ago
Donec id elit non mi porta gravida at eget metus. Maecenas sed diam eget risus varius blandit. Donec id elit mi porta.	
Ver solución	editar



Cambiar imagen

Cambiar datos

Cambiar contraseña

Cambiar ...

Salir

1

2

3

Crear Ejercicio



Juan Pablo Martinez



MateFun

Crear Ejercicio

Introducción a Matefun

B Bold / **I** Italic / **U** Underline / **abc** Striken
Italic Text CTRL + I

Subir video

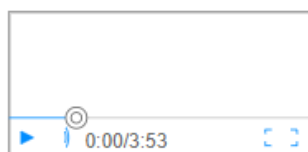
Adjuntar

Generar cuadro autocompletable

Matefun es un *lenguaje de programación* para aprender matemáticas.



Mira este video introductorio



Que dominios conoces, completa en el cuadro

escribe aqui: $\{= R, Z, N, Q\}$

Ahora prueba definirlos en matefun!

escribe aqui: $= \{R, Z, N, Q\}$

correr

Genial, Ahora verifica la hipótesis de Riemann:

Consola de matefun

Publicar

Guardar Borrador

Volver

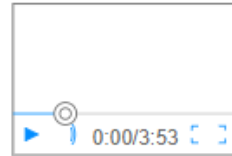


MateFun

Titulo del ejercicio

Temas que incluye:
Números primos, división, multiplicación.
Pensado para: 5to año

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus elementum mauris mi, ac fringilla est varius sit amet. Maeceenas faucibus, nisi eu pellentesque lacinia, orci quam condimentum ipsum, vitae vulputate velit lectus at erat. Quisque sapien ex, suscipit vel iaculis non, iaculis sed est. Ut iaculis ultrices nulla, at hendrerit lectus accumsan sed. Quisque eget urna in est cursus facilisis interdum ut augue. Nunc lobortis, purus in mollis faucibus, orci metus dapibus sem, eu egestas lorem ex quis dui. Interdum et malesuada fam



Nombre profesor
institución
educativa

Compilador matefun

Escribir texto

Enviar

Volver



Juan Pablo Martinez



MateFun

Soluciones al ejercicio: numeros primos con matefun

Soluciones de estudiantes

1	2	3	4	@	Input Group Example
Martin Rodriguez Liceo 43					
Xiomara Barrientos Liceo 43					
Pedro Hernandez Liceo 13					
Martin Rodriguez Instituto de la costa					
Daniel Salsedo Liceo 98					
Daniel Encasa Liceo 98					
Luis Gilberto Hernandez Liceo 43					
Juan Pablo Rodriguez Liceo 13					
Luis Pedro suarez colegio y liceo san pablo					
Soledad Cardozo Liceo 98					
Damian Gómez Liceo 43					
Soledad Gimenez Liceo 6					
Ana Lopez Liceo 54					
Lucia Dominguez Liceo 6					
Miguel Ramos Liceo 6					

Volver

Heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Codigo fuente:

Ejecutar

Consola de Matefn

Manual de usuario

EMMY

Manual de usuario



Objetivo

Presentar un manual de uso de la plataforma

Verónica Bentancor, Gastón Barreto

matefun@fing.edu.uy

CONTENIDO

Página Principal:.....	2
Inicio de sesión:.....	2
Estudiantes:.....	3
Navegar por tema:.....	3
Navegar por grado:.....	3
Navegar por docente:.....	3
Ver publicación:.....	4
Resolver publicación:	4
Continuar solución	6
Docentes	7
Iniciar sesión.....	7
Perfil docente	7
Crear publicación.....	8
Editar Publicación	11
Listar Soluciones.....	11
Ver solución.....	11
Administradores.....	12
Administrar Usuarios.....	12
Listar.....	12
Crear.....	12
Borrar	12
Administrar Temas	13
Listar.....	13
Crear y modificar.....	13
Administrar Publicaciones.....	13
Cambiar contraseña	13

EMMY:

Es un entorno virtual integrado para la publicación y resolución de desafíos matemáticos y de ciencias pudiendo utilizar el lenguaje MateFun. El sitio tiene tres tipos de usuarios, visitantes (que pueden ser estudiantes), los cuales resuelven publicaciones (pudiendo completar sus datos personales para identificar su solución), los docentes, los realizan publicaciones (pudiendo definir códigos privados con sus estudiantes), y deben estar previamente registrados por un administrador, quien se encarga de mantener el sitio.

Este documento presenta un manual de uso accesible para los tres tipos de usuarios.

Se definen dos conceptos que serán utilizados a lo largo del documento.

Código privado: es un identificador acordado entre el docente y los estudiantes, su utilidad es facilitar la identificación de soluciones por grupo. Un estudiante al enviar su solución ingresa un código, el docente al acceder al listado de soluciones puede buscar soluciones por código privado, obteniendo así las soluciones de un grupo en particular.

Código secreto: este identificador es proporcionado por la plataforma al estudiante cuando responde una publicación. Con este código el estudiante puede volver a cargar su borrador para continuar el trabajo o visualizar la respuesta una vez enviada.

PÁGINA PRINCIPAL:

La pantalla de inicio se compone de un buscador de publicaciones y dos botones, uno para acceso de estudiantes y otro para docentes

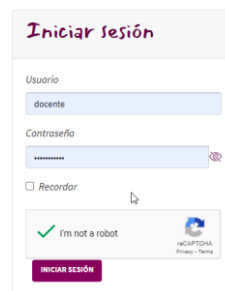
Desde la página principal se tiene acceso al buscador, a la documentación y a iniciar sesión.

INICIO DE SESIÓN:

Los usuarios administradores y docentes deben iniciar sesión para realizar sus tareas, lo cual pueden hacer en cualquier momento desde la barra de navegación superior.

El sitio no permite el registro de usuarios, por lo que los administradores deberán previamente crear las cuentas de los docentes.

Para iniciar sesión, se debe marcar la casilla de verificación.



The screenshot shows a login form with the following elements:

- Title: **Iniciar Sesión**
- Label: **Usuario**
- Input field: **docente**
- Label: **Contraseña**
- Input field: ********* (password masked)
- Checkbox: **Recordar**
- Verification: **I'm not a robot** (with a green checkmark)
- Buttons: **INICIAR SESIÓN** (purple) and **INICIAR SESIÓN** (blue)
- Footer: **EDUCACIONAL** (with a logo) and **Inicio - Ayuda**

ESTUDIANTES:

Todas las operaciones a realizar por parte del estudiante serán sin estar registrado.

El botón *Acceso Estudiantes* permite encontrar publicaciones, al ingresar por el mismo se presentan tres opciones de navegación: por tema, grado y docente. Las publicaciones se clasifican en cuatro dificultades (introductorio, intermedio, avanzado y experto) y se señalan con los colores celeste, verde, amarillo y rojo respectivamente. Algunos de estos pueden tener una corrección automática o ser interactivos.

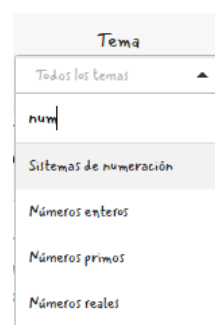
Se comienza a responder una publicación una vez seleccionada.

NAVEGAR POR TEMA:

Se listan todos los temas de los cuales existen publicaciones en el sitio.

Al seleccionar un tema, se presentan sus publicaciones indicando el nombre del docente que lo ha publicado, su fecha y descripción.

Se permite realizar un filtro en el listado, pudiendo filtrar por palabra clave, grado, dificultad y autor. Se puede introducir palabras en los filtros para acortar la búsqueda. Presionando el botón buscar se listan las publicaciones con el filtro seleccionado. El filtro se puede deshacer desde *Limpiar*.



NAVEGAR POR GRADO:

Se presentan los grados: de primero a sexto año escolar, (al elegir escolar se involucra a los anteriores), de primero a tercero de ciclo básico (ciclo básico involucra a estos tres), de cuarto a sexto de bachillerato (donde bachillerato involucra los tres anteriores), universitario o sin especificar.

Con grado sin especificar se listan publicaciones que no han sido catalogados por los docentes.

Según el grado seleccionado se presenta las publicaciones relacionadas, nuevamente se dispone de los filtros buscar por palabra clave, por tema, por dificultad y por autor. Se puede deshacer el filtro en *Limpiar*.

NAVEGAR POR DOCENTE:

Se listan todas las publicaciones de los docentes. Se muestra el nombre y apellido de aquellos que han publicado publicación, indicando la institución educativa a la que pertenecen.

Se dispone de los filtros por palabra clave, por tema, grado y por dificultad. Se puede deshacer el filtro en *Limpiar*.

VER PUBLICACIÓN:

Esta pantalla se compone de dos paneles, un panel donde se muestran datos del docente que publicó la publicación (nombre, apellido e institución), seguido del tema principal de la publicación, dificultad y grado del mismo y si tiene corrección automática o no. Desde este panel también se puede acceder a otras publicaciones del mismo autor (tocando “*Ver más publicaciones del autor*”) y tocando el grado o tema se puede acceder a listados con ese filtro.

El otro panel consiste en el de publicación, lo primero que se ve es el título de la publicación, seguido de la letra del mismo.

Al final con el botón *resolver* el estudiante puede generar su solución, y en *Volver a publicaciones*, se regresa a las mismas.

Docente: Docente de pruebas
Institución: Matefun

Tema principal: Números primos

Dificultad: **Experto**

Grado: **Universitario**

Fecha: jue. 27 may. 2021

Con corrección automática

[Ver más publicaciones del autor](#)

RESOLVER PUBLICACIÓN:

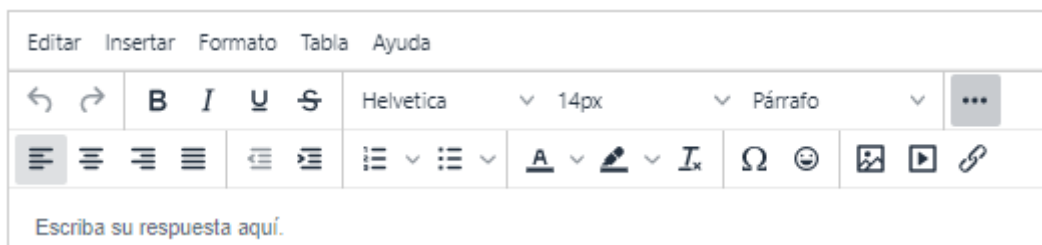
No todas las publicaciones admiten respuestas. En aquellas que si lo permiten, hay que identificarse como estudiante, para esto se requiere ingresar *nombre* y *apellido*, opcionalmente se puede ingresar, institución a la cual pertenece, y un código privado (debe coordinarse con el docente)

Debajo del panel identificatorio, se muestra también la letra completa de la publicación (con su respectivo panel lateral).

Si la publicación tiene datos interactivos, estos serán de completar y de múltiple opción.

Las publicaciones pueden tener diferentes componentes habilitados; estos pueden ser el cuadro de texto, el editor de código MateFun y el intérprete:

Cuadro de texto:



Al redactar la solución se presenta un cuadro con opciones en menú y otros iconos como atajo.

Desde la esquina inferior derecha se puede ajustar el tamaño del cuadro de texto.



Desde el menú las opciones son:

- Formato, se puede elegir el tipo de letra, tamaño, color de letra, color de fondo, estilo; tachado, negrita, subrayado, cursiva, alineación y también elegir títulos, subtítulos tipo de letra y limpiar formato.
- Editar, se entran opciones de copiar, pegar y cortar, además se puede seleccionar todo y buscar y reemplazar.
- Insertar, se pueden insertar caracteres especiales, matemáticos, emoticones. Se pueden agregar imágenes y documentos desde una URL.

- Tabla, se puede insertar una tabla o agregar columnas y filas a una existente. Luego de tener una tabla en propiedades de celda o ajustando la tabla se puede modificar el tamaño.

En fila y columna se pueden eliminar, insertar, copiar, pegar filas y columnas respectivamente. También se puede editar las propiedades de la tabla, color, fondo, y eliminar la misma.

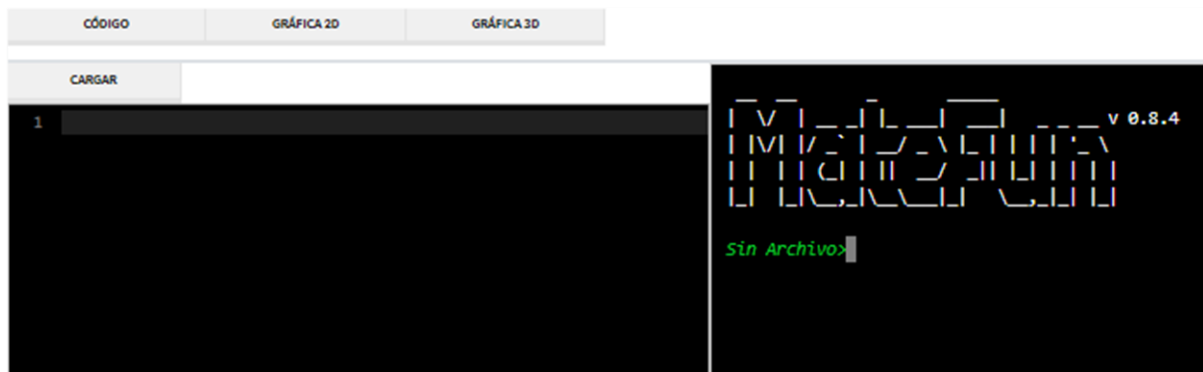
- En Ayuda se encuentran atajos de teclado.

Campos interactivos:

En caso de que la publicación tenga cuadros múltiple opción, se debe seleccionar de la lista la opción correcta y se puede elegir la solución en blanco en caso de no querer responderla.

En los cuadros a completar se debe ingresar la respuesta. No hay que preocuparse por las mayúsculas, los espacios ni los tildes, ya que la plataforma no los tiene en cuenta para la corrección automática.

Editor de código e intérprete de MateFun:



En caso de que el autor de la publicación haya habilitado el intérprete y el editor de código, se presentarán dos ventanas, la de la izquierda donde debe ingresarse el código fuente y en la ventana de la derecha donde se ejecutan los comandos.

Cada vez que se modifique el código fuente en la ventana de la izquierda debe presionarse el botón *Cargar*, mientras que en la ventana derecha se pueden ejecutar los comandos e invocar a las funciones definidas en el código fuente.

Para visualizar las figuras y gráficas 3D deberá ir a la solapa: Gráfica 2D y Gráfica 3D respectivamente o ingresar un comando que genere las figuras en el panel derecho.

En gráficas 2D se permite descargar las figuras, ajustar los ejes y la velocidad de animación.

En las gráficas 3D se puede indicar la velocidad de animación así como mover sobre los ejes.

En el caso de estar deshabilitado el editor de código, se mostrará solamente la consola con las opciones de gráficas.

Enviar solución:

En caso de que la publicación tenga habilitada el envío de respuestas, se muestra la opción de enviar la respuestas y de guardarla como borrador para continuarla más tarde, donde se proporcionará un código secreto y un enlace con el cual podrá acceder para continuar con la solución.

Cuando se desee realizar la entrega, se presionará *Enviar*, se abrirá un cuadro de confirmación, al presionar aceptar la solución queda enviada sin poder volver a ser modificada. Se puede seguir utilizando el código anterior o su enlace para revisar la entrega.

Si el publicación tenía habilitada corrección automática se indicará en verde si las opciones ingresadas son correctas y en rojo en caso de ser incorrectas.

CONTINUAR SOLUCIÓN

Para continuar una solución, se necesita contar con el código privado o en el enlace. En caso de tener el código, se puede introducir yendo a *Continuar solución* en el panel superior del sitio.

Si no se tiene el código ni el enlace, puede consultar con su docente para conseguirlo.



The screenshot shows a web browser window with a dark blue header. On the left is the 'EMMY' logo. On the right, there are navigation links: 'Continuar solución', 'Documentación', and 'Intérprete Matefun'. A 'Iniciar sesión' link is in the top right corner. A logo for 'matefun' is also visible. A white dialog box titled 'Continuar solución' is centered on the screen. It contains a text input field with 'ABC123' entered, and the label 'Código de respuesta:'. Below the input field is the text: 'Introduce tu código secreto para acceder a la solución. Si no la recuerdas, comunícate con tu docente'. At the bottom of the dialog are two buttons: 'IR A LA SOLUCIÓN' and 'Cerrar'.

DOCENTES

Los docentes crean publicaciones en la plataforma, los cuales tienen un tema, un grado y una dificultad asignada. Pueden ser interactivos, con corrección automática visible para los estudiantes y hacer uso del intérprete.

INICIAR SESIÓN

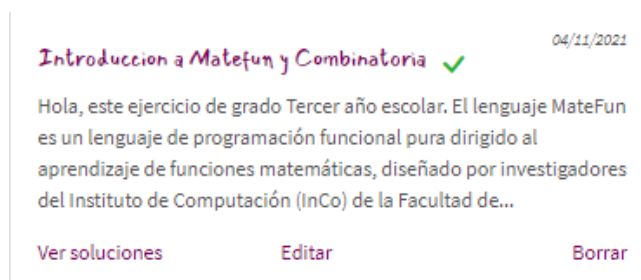
Para iniciar sesión se utiliza el menú superior. Se debe especificar usuario y contraseña. En caso de olvido de contraseña, debe contactarse con un administrador. Al iniciar, se visualiza el perfil del docente.

PERFIL DOCENTE

Esta pantalla se compone de dos paneles, un panel donde se muestran datos del docente y un menú para acceder a las publicaciones, modificar el perfil y cerrar sesión. El otro panel varía el menú en que se encuentra:

Mis publicaciones

Se listan las publicaciones en estado publicado y borrador, junto con su descripción corta, ordenadas cronológicamente. Aquellas que estén publicadas tendrán un *tick* verde ✓ en su título (al acercarse al curso se indica *publicado*). Cada una de ellas tendrán un enlace para editar, borrar y en caso de tenerlas, ver las respuestas que ha recibido.



Se permite realizar un filtro en el listado, pudiendo filtrar por palabra clave y seleccionar tema, grado y dificultad. Presionando el botón buscar se listan las publicaciones con el filtro seleccionado. El filtro se puede deshacer desde *Limpiar*.

Se pueden borrar publicaciones seleccionándolas de manera individual, esta acción es irreversible y borra también todas las respuestas que ha recibido.

Modificar mi cuenta

Se permite modificar el correo, el nombre, la institución y la contraseña (La contraseña debe tener largo entre 8-64, mayúsculas, minúsculas y símbolos)

Cerrar sesión


Cierra la sesión y regresa a la pantalla principal.

CREAR PUBLICACIÓN

Al crear una publicación se debe especificar el título, el grado, la dificultad y el tema.

Cuadro de texto:

Al redactar la publicación se presenta un cuadro con opciones en menú y otros iconos como atajo.

Desde la esquina inferior derecha se puede ajustar el tamaño del cuadro de texto. 

Desde el menú las opciones son:

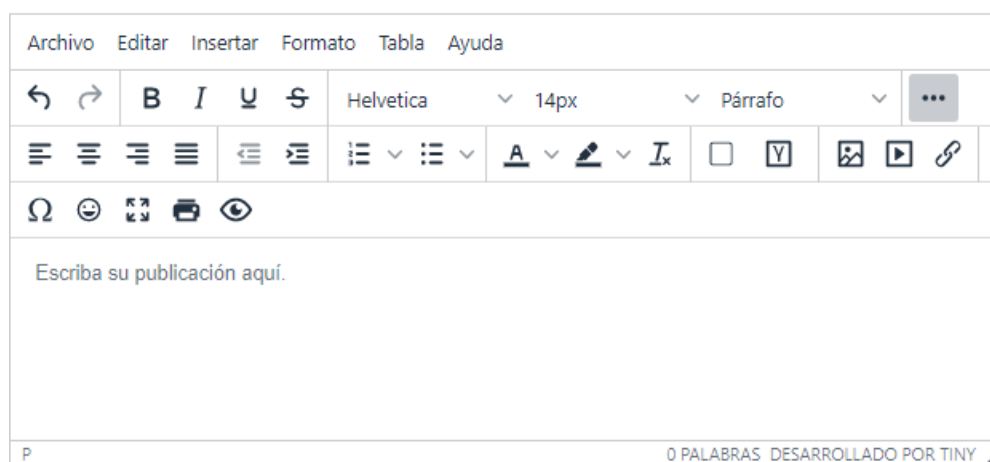
- Formato, se puede elegir el tipo de letra, tamaño, color de letra, color de fondo, estilo; tachado, negrita, subrayado, cursiva, alineación y también elegir títulos, subtítulos tipo de letra y limpiar formato.
- Editar, se entran opciones de copiar, pegar y cortar, además se puede seleccionar todo y buscar y reemplazar.
- Insertar, se pueden insertar caracteres especiales, matemáticos, emoticones, campos de completar y campos de múltiple opción. Se pueden agregar imágenes y documentos desde una URL.

- Tabla, se puede insertar una tabla o agregar columnas y filas a una existente. Luego de tener una tabla en propiedades de celda o ajustando la tabla se puede modificar el tamaño.

En fila y columna se pueden eliminar, insertar, copiar, pegar filas y columnas respectivamente. También se puede editar las propiedades de la tabla, color, fondo, y eliminar la misma.

- En Ayuda se encuentran atajos de teclado.

Al posicionarse en un nuevo renglón de la publicación, se puede cargar una imagen almacenada en el equipo o crear una tabla.




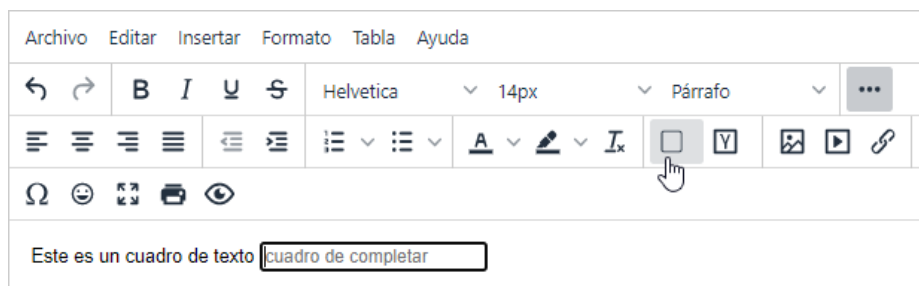
Campos interactivos:

Se tienen dos cuadros interactivos: de completar y de múltiple opción, los cuales son corregidos automáticamente por la plataforma al ver la solución de los estudiantes. El corrector automático tomará respuestas equivalentes sin importar las mayúsculas, las minúsculas, los espacios en blanco ni los tildes.


Es posible transformar un cuadro de completar en uno de múltiple opción (o viceversa) simplemente dejando el cursor sobre el cuadro generado y seleccionando la opción de múltiple opción (o de completar)

Campo de completar

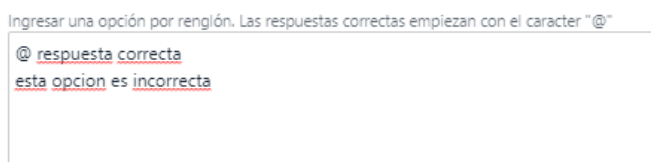
Al seleccionar el menú de cuadro de completar  se abre un cuadro emergente. Aquí se deben listar todas las opciones correctas, una por renglón. Una vez insertado, se pueden consultar y modificar los valores, posicionando el cursor el cuadro resultante y volviendo a presionar el mismo botón.



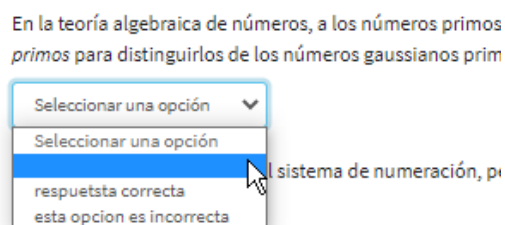
Cuadro múltiple opción

Al seleccionar el menú de campo múltiple opción , se abre un cuadro emergente. Aquí se deben listar todas las opciones posibles, tanto correctas (que deben empezar con un carácter @) como incorrectas, una por renglón, que serán luego mostradas al estudiante. Una vez insertado, se pueden consultar y modificar los valores, dejando el cursor sobre el cuadro resultante y volviendo a presionar el mismo botón.

Campo de múltiple opción



Los campos múltiple opción serán vistos por el estudiante de esta forma:



Intérprete MateFun



Se puede hacer uso del intérprete mientras se realiza la publicación, el cual consiste dos ventanas, la de la izquierda, donde debe ingresarse el código fuente y la ventana de la derecha donde se ejecutan los comandos.

Cada vez que se modifique el código fuente en la ventana de la izquierda debe presionarse el botón Cargar, mientras que en la ventana derecha se pueden ejecutar los comandos e invocar a las funciones definidas en el código fuente.

Para visualizar las figuras y gráficas 3D deberá ir a la solapa: Gráfica 2D y Gráfica 3D respectivamente o ingresar un comando que genere las figuras en el panel derecho.

En gráficas 2D se permite descargar las figuras, ajustar los ejes y la velocidad de animación. Mientras que en las gráficas 3D se puede indicar la velocidad de animación así como mover sobre los ejes.

El código que este escrito en el editor le aparecerá al estudiante al momento de responder en caso de haber habilitado el intérprete para los mismos.

Descripción corta

La descripción corta es un breve resumen de la publicación, la cual ayuda a identificarlo y es visible al navegar entre el listado de publicaciones, debe contar con un largo mínimo de 20 caracteres.

Opciones de publicación

Luego de la descripción corta, se pueden habilitar una serie de opciones para la publicación:

Opciones de publicación:

<i>Los estudiantes podrán:</i>	
<input checked="" type="checkbox"/> Enviar sus respuestas	<input checked="" type="checkbox"/> Usar el editor de texto
<input checked="" type="checkbox"/> Ver la corrección automática	<input checked="" type="checkbox"/> Usar el intérprete de MateFun
	<input checked="" type="checkbox"/> Usar el editor de código MateFun
<i>Si no se aceptan respuestas, los estudiantes no podrán ver la corrección automática</i>	<i>Si no se permite el uso del intérprete, los estudiantes no podrán probar su código en el editor</i>

Finalmente, la publicación puede ser guardada como borrador o publicada. En caso de guardarla como borrador esta no podrá ser visualizada por los estudiantes. Una vez publicada, no se podrá guardar como borrador y quedará visible permanentemente.

EDITAR PUBLICACIÓN

Se puede ir a editar la publicación a través del panel lateral del perfil docente o a través del botón *editar* al visualizar la publicación. Al editarla se pueden modificar todas las opciones y los campos de completar y múltiple opción (posicionando el cursor sobre éste y luego presionando el botón de insertar campo correspondiente).

Las publicaciones indican abajo del título si han sido publicadas o no.

LISTAR SOLUCIONES

Para listar las soluciones de un publicación se debe acceder a estas a través del listado de publicaciones del perfil de usuario. Aquellas publicaciones que hayan recibido soluciones tendrán habilitada la opción de verlas.

Una vez accedido, se listan en orden cronológico las respuestas que ha recibido la publicación. Por cada una de ellas se muestra el nombre, el estado de la solución (las soluciones borrador carecen de tick verde ✓), y podrá aparecer el nombre de la institución y un código privado.

Estas soluciones se pueden filtrar por nombre, institución y código privado o se pueden ordenar por cualquier columna al tocar su nombre. También se podrán borrar de a una o en conjunto.

Finalmente, se puede acceder a ellas tocando cualquier renglón de la tabla.

VER SOLUCIÓN

Para visualizar la solución se muestra primero la letra de la publicación. En caso de que el publicación tuviera cuadros interactivos, Emmy realizará la corrección automática de los campos, tomando respuestas equivalentes sin importar las mayúsculas, las minúsculas, los tildes o los espacios en blanco.

A la derecha de la publicación se muestra información acerca de la solución, esto incluye el nombre, la institución, el código privado, la fecha de última modificación, el estado (borrador o definitivo), y el código secreto.

Se muestra también el código privado, el cual puede ser compartido entre varios estudiantes y se sugiere coordinarlo con estos, ya tiene la utilidad de poder identificar soluciones por grupos.

El código secreto junto con el enlace es útil para que el estudiante pueda volver a cargar su borrador o visualizar su entrega y puede entregárselos en caso de olvido.

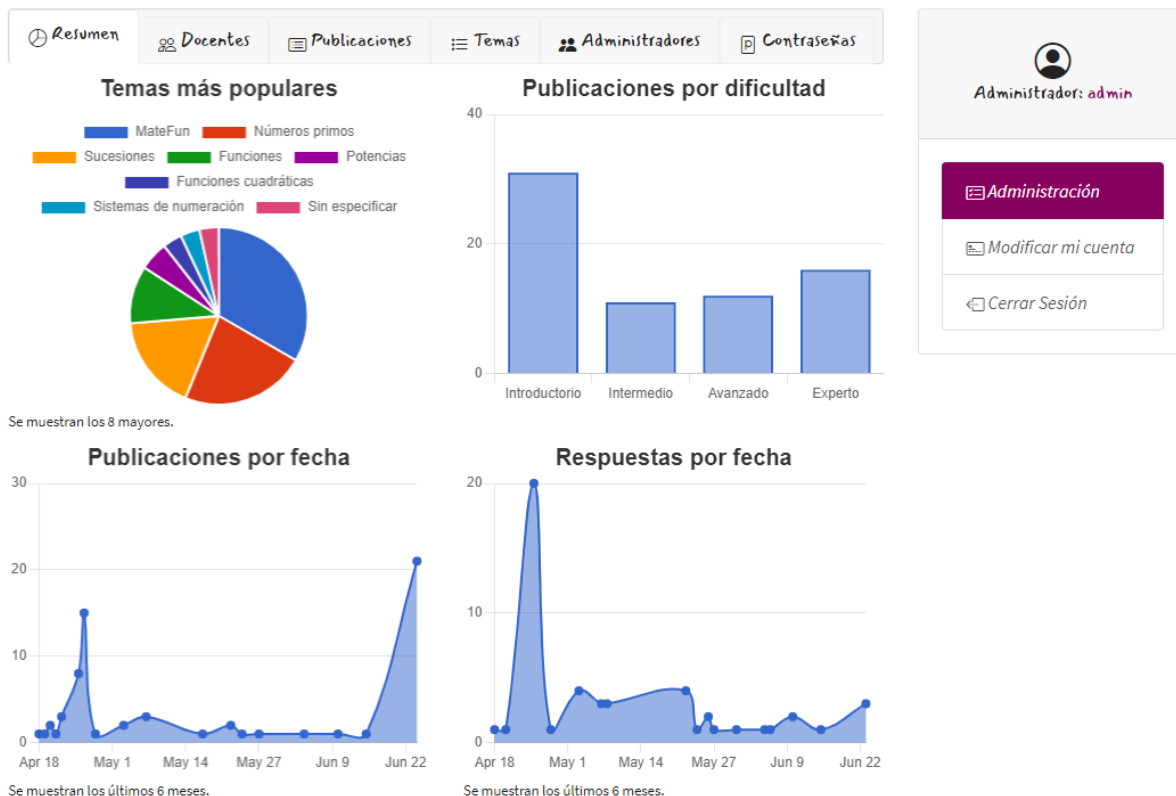
Dependiendo de las opciones de la publicación, la respuesta puede incluir el texto del estudiante, el intérprete y el código MateFun.

<i>Estudiante:</i> Margarita Natalia
<i>Institución:</i> Liceo N°1 Prof. Juan Luis Perrou
<i>Código privado:</i> Liceo-PrimeroA-2
<i>Fecha de entrega:</i> mié. 26 may. 2021, (7:13 PM)
<i>Código secreto</i> claskb
<i>Enlace de acceso</i> http://localhost:3000/claskb
<i>Estado:</i> Borrador

ADMINISTRADORES

El panel de administración incluye un listado con los temas más populares (aquellos con mayor cantidad de publicaciones), la cantidad de publicaciones por dificultad, y dos gráficas que muestran en el tiempo, la cantidad de publicaciones y la cantidad de respuestas.

Resumen



ADMINISTRAR USUARIOS

LISTAR

Los usuarios docente y administrador se pueden controlar desde el panel de administración.

En la pestaña *Docentes* se lista este tipo de usuario, y se puede filtrar u ordenar por nombre, correo, usuario e institución (no ambos a la vez).

CREAR

Se pueden crear usuarios administradores (activando la opción) y usuarios docentes.

Se debe especificar el correo, el nombre de usuario (debe empezar con una letra, tener un largo entre 2-40, todo en minúsculas), el nombre completo, la institución (solo para docentes) y la contraseña (debe tener un largo entre 8-64, mayúsculas, minúsculas y símbolos)

BORRAR

Los usuarios se pueden borrar seleccionándolos en el listado correspondiente

ADMINISTRAR TEMAS

LISTAR

En la pestaña *temas* se listan los temas del sitio. Se pueden filtrar por nombre u ordenar por nombre o cantidad de publicaciones.

CREAR Y MODIFICAR

Se puede crear un tema tocando el botón correspondiente, o editar uno existente tocando su nombre en el listado.

El nombre es obligatorio, mientras que la descripción es opcional.

ADMINISTRAR PUBLICACIONES

En la pestaña *publicaciones* se listan las publicaciones, el autor y la fecha de última modificación. Se puede filtrar por título o autor, u ordenar por título, autor o fecha.

Para borrar una publicación se debe seleccionar en el listado. Al hacerlo, se borran también todas sus respuestas.

CAMBIAR CONTRASEÑA

En la pestaña *contraseñas* se puede cambiar la contraseña de cualquier usuario, para eso se selecciona el nombre de usuario en el listado y se actualiza la contraseña, la debe tener un largo entre 8-64, mayúsculas, minúsculas y símbolos.

Documento de pruebas de usuario

EMMY

Instructivo de pruebas



Objetivo

Validar el flujo de uso de la plataforma Emmy

Verónica Bentancor, Gastón Barreto

CONTENIDO

Objetivo.....	2
Sujetos de prueba.....	2
Metodología.....	2
Diagrama de Pruebas con docentes.....	3
Diagrama de Pruebas con expertos	4
Flujos	5
Desarrollo	6
Diseño de pruebas	7
como Docente	7
prueba de docente #1	7
prueba de docente #2	8
Prueba de docente #3.....	10
prueba de docente #4.....	10
prueba de docente #5.....	12
como Estudiante.....	13
PRueba de estudiante #1	13
PRueba de estudiante #2	14
PRueba de estudiante #3	14
Formulario evaluación	17
Evaluación docente y estudiante	17

EMMY:

Es un entorno virtual integrado para la publicación y resolución de desafíos matemáticos, con la posibilidad de utilizar el lenguaje MateFun para resolverlos. El sitio tiene tres tipos de usuarios, visitantes (que pueden ser alumnos o no), los cuales resuelven publicaciones (pudiendo indicar un código privado y completar sus datos personales para identificar su solución), los docentes, los cuales publican ejercicios (pudiendo definir códigos privados con sus estudiantes), y deben estar previamente registrados por un administrador, quien se encarga de mantener el sitio.

Se definen dos conceptos que serán utilizados a lo largo del documento.

Código privado: es un identificador acordado entre el docente y los estudiantes, su utilidad es facilitar la identificación de soluciones por grupo. Un estudiante al enviar su solución ingresa un código, el docente al acceder al listado de soluciones puede buscar soluciones por código privado, obteniendo así las soluciones de un grupo en particular.

Código secreto: este identificador es proporcionado por la plataforma al estudiante cuando responde una publicación. Con este código el estudiante puede volver a cargar su borrador para continuar el trabajo o visualizar la respuesta una vez enviada.

OBJETIVO

El objetivo de este documento es presentar casos de uso para de usuarios, para validar las decisiones de diseño de la plataforma.

SUJETOS DE PRUEBA

Se contó con un grupo heterogéneo para la realización de la prueba. Para explotar las características únicas de cada uno, se dividieron los sujetos de pruebas en tres categorías, docentes, estudiantes y expertos en el lenguaje MateFun.

El peso de las pruebas con los docentes y estudiantes permitió evaluar la usabilidad y funcionalidades de Emmy. Las pruebas enfocadas a los expertos en el lenguaje se centraron en las características y las implementaciones propias del intérprete, el editor de código y las gráficas.

METODOLOGÍA

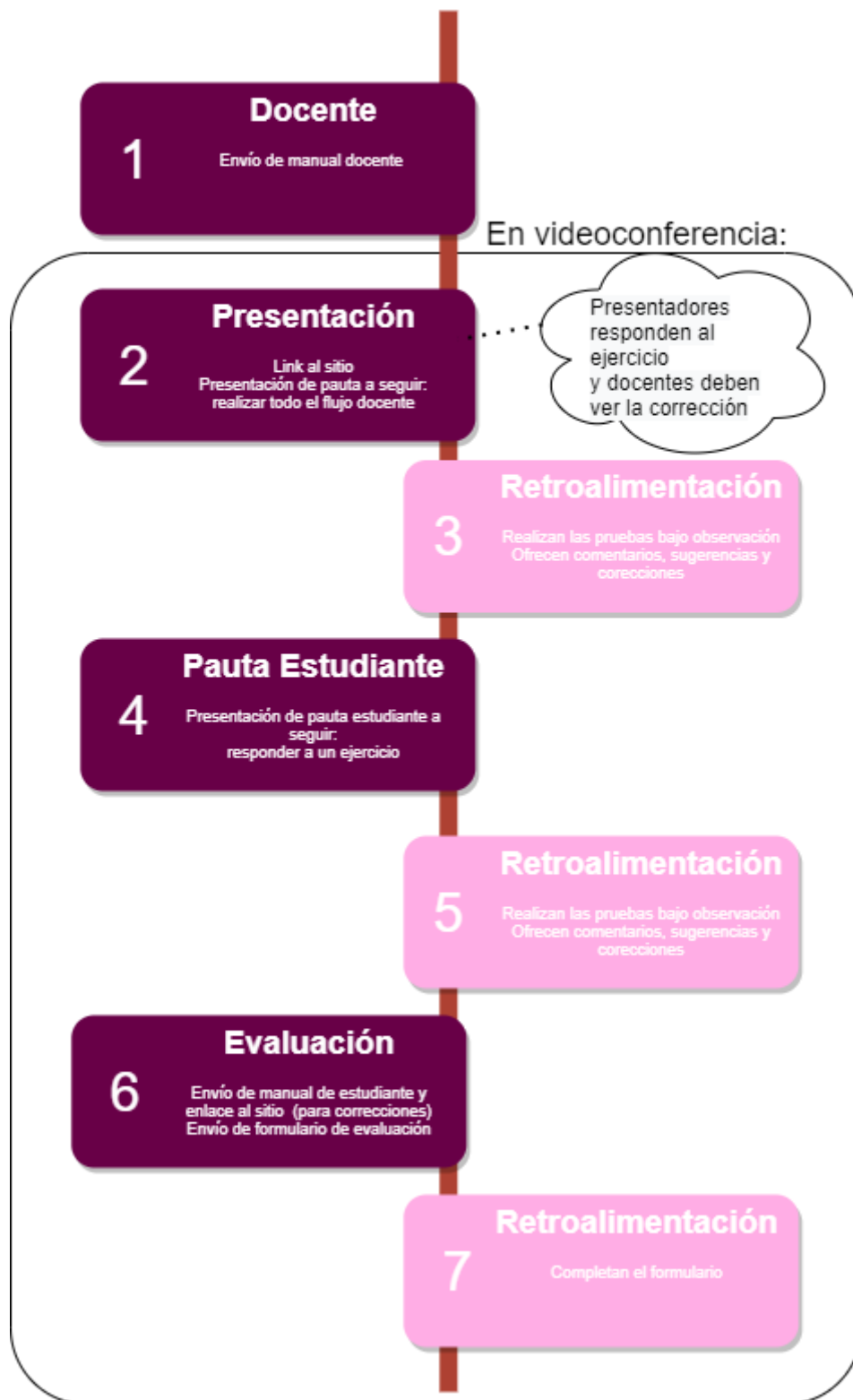
Para poder realizar las pruebas primero se cargaron en el sitio varios usuarios, con publicaciones y respuestas reales, para que los usuarios pudieran ver el sitio con todas sus características.

Previo a las pruebas se les enviará a los usuarios un manual de uso acorde a su perfil y un instructivo con las pruebas a realizar.

Para los docentes, se realizaron pruebas en tiempo real (vía video conferencia), donde se observó (sin intervención) la operativa del usuario al replicar las instrucciones recibidas, mientras que las pruebas a usuarios expertos fueron llevadas en solitario.

Al finalizar las pruebas se les entregó un formulario de evaluación escrito y se les realizó una entrevista abierta (en video conferencia para docentes, y por mail a los expertos).

Pruebas con docentes



Pruebas con usuarios expertos



Se definieron posibles flujos para los dos tipos de usuarios de la plataforma

Como docente

- 1) Comenzar en la página principal
- 2) Iniciar sesión
- 3) Publicar un ejercicio
 - a) Ingresar una letra del ejercicio
 - b) Introducir campos de completar y múltiple opción
 - c) Probar un código en el Interpretador
 - Probar graficar una función
 - d) Dejar un fragmento de código en el intérprete para el usuario
- 4) Editar un ejercicio
 - a) Cambiar su tema, grado y dificultad
 - b) Modificar su contenido
 - c) Cambiar las opciones de autocorrección.
- 5) Buscar un ejercicio en su perfil
 - a) Listar soluciones
 - Ver una solución
 - Borrar una solución particular

Como estudiante

- 6) Comenzar en la página inicial
- 7) Buscar una publicación
 - a) A partir de una palabra clave
 - b) A partir del menú de navegación
- 8) Resolver la publicación
 - a) Identificarse
 - b) Responder los campos interactivos
 - c) Escribir un código y ejecutarlo en el intérprete
 - d) Graficar
 - e) Enviar la solución como borrador
- 9) Salir del sitio
- 10) Volver a entrar a la solución el código privado de respuesta
- 11) Enviar la respuesta

DESARROLLO

A los docentes se les pidió realizar las tres pruebas en orden, primero, una sencilla para que se familiarizaran con el sitio, luego, pasaron a realizar la siguiente, que ya requería agregar elementos multimedia, tablas y campos de completar.

Una vez que habían visto cómo publicar ejercicios, se les pidió que realizaran la prueba de estudiantes, para que vieran desde otro punto de vista la plataforma e hicieran uso de los códigos privados. Por último, tuvieron que realizar la tercer prueba docente, que consistía en filtrar y visualizar respuestas de un grupo de estudiantes en particular, para que pudieran ver el flujo completo de los códigos privados.

Para los usuarios expertos, se les pidió publicar ejercicios, y luego responder una publicación más completa, que contenía la definición de funciones y gráficas. Además, se les pidió aportar un ejercicio nuevo a la plataforma y contestar alguna publicación de su elección.

A los usuarios estudiantes, de manera presencial, se les pidió que ingresaran al sitio e intentaran resolver una publicación.

Los docentes y los expertos recibieron previamente los manuales correspondientes, mientras que los estudiantes se les pidió completar la consigna sin haber visto el sitio ni tener conocimiento de los manuales.

PRUEBA DE DOCENTE #1

Esta prueba tiene como objetivo realizar una publicación sencilla en el sitio

PASOS:

Inicia sesión:

Utiliza el usuario y contraseña provisto.

Crear la siguiente publicación

La publicación se debe llamar: Introducción a las gráficas, se trata de un ejercicio *introdutorio*, de la temática de *Funciones*, que está pensado para *bachillerato*.

En el cuerpo de la publicación, copia este texto

Graficar las siguientes funciones usando el comando ?grafica.

Por ejemplo ?grafica recta1

recta1 :: R -> R

recta1(x) = 2 * x + 1

recta2 :: R -> R

recta2 (x) = x - 3

recta3 :: R -> R

recta3 (x) = recta2(recta1 (x))

recta4 :: R -> R

recta4 (x) = -3/5*x -3

recta5 :: R -> R

recta5 (x) = 3/2*x

parabola :: R -> R

parabola (x) = 2*x*x - 3*x +1

Escribe como descripción:


Introducción al comando ?grafica de MateFun con ejemplos de funciones.

Envía la publicación.

Resultado esperado:

MateFun > Acceso Estudiantes > Publicaciones > Ejemplos de graficar funciones en MateFun

Ejemplos de graficar funciones en MateFun



Docente: *MateFun*

Institución: *FING*

Tema principal:

Funciones

Dificultad:

Introductorio

Grado:

Bachillerato

Fecha:

25 abril 2021

[Ver más publicaciones...](#)

Graficar las siguientes funciones usando el comando `?grafica`. Por ejemplo `?grafica recta1`

recta1 :: $\mathbb{R} \rightarrow \mathbb{R}$
recta1(x) = $2 * x + 1$

recta2 :: $\mathbb{R} \rightarrow \mathbb{R}$
recta2 (x) = $x - 3$

recta3 :: $\mathbb{R} \rightarrow \mathbb{R}$
recta3 (x) = recta2(recta1 (x))

recta4 :: $\mathbb{R} \rightarrow \mathbb{R}$
recta4 (x) = $-3/5 * x - 3$

recta5 :: $\mathbb{R} \rightarrow \mathbb{R}$
recta5 (x) = $3/2 * x$

parabola :: $\mathbb{R} \rightarrow \mathbb{R}$
parabola (x) = $2 * x * x - 3 * x + 1$

PRUEBA DE DOCENTE #2

Esta prueba tiene como objetivo evaluar algunas funcionalidades avanzadas de publicación.

PASOS:

Inicia sesión:

Utiliza el usuario y contraseña provisto.

Crea la publicación

La publicación se debe llamar: *Criba de Eratóstenes*, es una publicación sobre números primos, pensada para universitarios expertos.

En el cuerpo de la publicación, copia este texto:

Eratóstenes fue un matemático, astrónomo y geógrafo griego, que desarrolló la Criba de Eratóstenes.

La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

Algoritmo:

Primero se forma una tabla con todos los números naturales comprendidos entre 2 y n y se van tachando los números que no son primos de la siguiente manera:

Comenzando por el 2, se tachan todos sus múltiplos;

comenzando de nuevo, cuando se encuentra un número que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente.

El proceso termina cuando el cuadrado del siguiente número confirmado como primo es mayor que n .

Se pide implementar la criba de Eratóstenes para un N arbitrario y completar el siguiente cuadro (separados con coma, sin espacios entre medio):

Inserta la imagen de este link, y alinéala a la derecha:

<https://upload.wikimedia.org/wikipedia/commons/b/b3/Eratosthene.01.png>

El título 'Algoritmo:' debe ir como encabezado 2.

Luego dibuja esta tabla:

N=10	2,3,5,7
N=20	
N=30	

Inserta dos cuadros de completar en el renglón de N=20 cuya respuesta correcta es :

2,3,5,7,11,13,17,19, para N=30, la respuesta correcta es: 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47.


En descripción corta agrega lo siguiente:

La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

Publica el ejercicio

Resultado esperado:

Criba de Eratóstenes


Docente: *Verónica B.*
Institución: *FIING*

Tema principal:
Números primos

Dificultad:
Experto

Grado:
Universitario

Fecha:
25 abril 2021
Con corrección automática
[Ver más publicaciones...](#)

Eratóstenes fue un matemático, astrónomo y geógrafo griego, que desarrolló la Criba de Eratóstenes, la cual es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

Algoritmo:

Primero se forma una tabla con todos los números naturales comprendidos entre 2 y n y se van tachando los números que no son primos de la siguiente manera:

- Comenzando por el 2, se tachan todos sus múltiplos;
- comenzando de nuevo, cuando se encuentra un número que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente.
- El proceso termina cuando el cuadrado del siguiente número confirmado como primo es mayor que n .

Se pide implementar la criba de Eratóstenes para un N arbitrario y completar el siguiente cuadro (separados con coma, sin espacios entre medio):

N=10	2,3,5,7
N=20	cuadro de texto
N=30	cuadro de texto



PRUEBA DE DOCENTE #3

En esta prueba de uso se busca validar la operativa de visualizar las soluciones de un ejercicio

PASOS

Busca la publicación sobre la Criba de Eratóstenes, solucionario

Busca la solución, puedes utilizar filtros...

Te pusiste de acuerdo con tus estudiantes del Liceo 1, de primero A, en usar el código privado *Primero-A*. Visualiza cualquier solución de estos estudiantes

Borra las soluciones

Vuelve al listado anterior y borra todas las soluciones que no tengan código privado.

PRUEBA DE DOCENTE #4

Esta prueba tiene como objetivo evaluar algunas opciones avanzadas de publicación.

PASOS:

Inicia sesión:

Utiliza el usuario y contraseña provisto.

Crea la publicación

La publicación se debe llamar: *Criba de Eratóstenes*, es una publicación sobre números primos, pensada para universitarios expertos.

En el cuerpo de la publicación, copia este texto:

Eratóstenes fue un matemático, astrónomo y geógrafo griego, que desarrolló la Criba de Eratóstenes.

La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

Algoritmo:

Primero se forma una tabla con todos los números naturales comprendidos entre 2 y n y se van tachando los números que no son primos de la siguiente manera:

Comenzando por el 2, se tachan todos sus múltiplos;

comenzando de nuevo, cuando se encuentra un número que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente.

El proceso termina cuando el cuadrado del siguiente número confirmado como primo es mayor que n .

Se pide implementar la criba de Eratóstenes para un N arbitrario y completar el siguiente cuadro (separados con coma, sin espacios entre medio):

Inserta la imagen de este link, y alinéala a la derecha:

<https://upload.wikimedia.org/wikipedia/commons/b/b3/Eratosthene.01.png>

El título 'Algoritmo:' debe ir como encabezado 2.

Agrega este vídeo

<https://www.youtube.com/watch?v=GST7EhThqpQ>

Luego dibuja esta tabla:

N=10	2,3,5,7
N=20	
N=30	

Inserta dos cuadros de completar en el renglón de $N=20$ cuya respuesta correcta es:

2,3,5,7,11,13,17,19, para $N=30$, la respuesta correcta es: 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47.

En descripción corta

La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

Opciones de publicación

Haz que los estudiantes puedan enviarte sus respuestas, pero no ver la corrección automática, también haz que no puedan enviar texto (solo completar los cuadros interactivos)

Publica el ejercicio

PRUEBA DE DOCENTE #5

Esta prueba tiene como objetivo realizar una publicación sencilla en el sitio con algunas opciones de publicación

PASOS

Inicia sesión:

Utiliza el usuario y contraseña provisto.

Crear la siguiente publicación

La publicación se debe llamar: Introducción a las gráficas, se trata de un ejercicio *introdutorio*, de la temática de *Funciones*, que está pensado para *bachillerato*.

En el cuerpo de la publicación, copia este texto

Graficar las siguientes funciones usando el comando ?grafica.

Por ejemplo ?grafica recta1

recta1 :: R -> R

recta1(x) = 2 * x + 1

recta2 :: R -> R

recta2 (x) = x - 3

recta3 :: R -> R

recta3 (x) = recta2(recta1 (x))

recta4 :: R -> R

recta4 (x) = -3/5*x -3

recta5 :: R -> R

recta5 (x) = 3/2*x

parabola :: R -> R

parabola (x) = 2*x*x - 3*x +1

Para la descripción

Escribe:

Introducción al comando ?grafica de MateFun con ejemplos de funciones.

Opciones de publicación

Este es un ejercicio para practicar de forma autodidacta... haz que no puedan enviar sus respuestas, ni utilizar el editor de código.

Envía la publicación.

PRUEBA DE ESTUDIANTE #1

Con esta prueba se pretende validar el proceso de responder a un ejercicio

PASOS

Ve al sitio

Busca la publicación

Es un ejercicio avanzado de bachillerato, se titula *Determinar cantidad de raíces reales en una función*, su autor es el docente *MateFun*.

Determinar cantidad de raíces reales en una función cuadrática

Responder

Nombre y apellido:*	Institución:	Código privado:
<input type="text"/>	<input type="text"/>	<input type="text"/>

Las funciones de segundo grado, también llamadas cuadráticas, son de la forma:

$$f(x) = ax^2 + bx + c,$$

donde el coeficiente a es distinto de 0.

Cuál es el grado de una función cuadrática? (introduce solo el número)

La fórmula de bhaskara, nos permite hallar raíces de la función, y es de la forma:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

El discriminante, que es de la forma:

$$\Delta = \sqrt{b^2 - 4ac}$$

Nos da información acerca de las raíces de la función:


- Si es >0 , entonces las dos raíces son reales
- Si es $=0$, entonces las dos raíces son iguales (llamadas raíces dobles)
- Si es <0 , no tiene raíces reales.

Por ejemplo, la siguiente función, tiene

$$x^2 + 4x$$

La siguiente función tiene

$$x^2 + 1$$



Docente: **MateFun**

Institución: **FING**

Estudiar las funciones cuadráticas y sus raíces, utilizando el determinante.

Tema principal: Funciones cuadráticas

Dificultad: Avanzado

Grado: Bachillerato

Fecha: vie. 04 jun. 2021

Con corrección automática

[Ver más publicaciones del autor](#)

Responde la publicación

Introduce tu nombre real y una institución. Previamente has coordinado con el docente, y te da pedido introducir, y el código privado *Liceo34-5B*

Completa los campos

Responde todos los campos de completar

Envía la respuesta y ve la corrección

PRUEBA DE ESTUDIANTE #2

Con esta prueba se pretende validar el proceso de responder a un ejercicio

PASOS

Esta prueba comienza igual que la prueba de estudiante #1:

Responde la publicación...

Guarda la publicación como borrador en vez de enviarla

Anótate el código privado

Ve al sitio principal

Vuelva a la publicación

Intenta ingresar a la respuesta utilizando el código privado

Realiza cambios en tu respuesta

Envíala definitivo.

PRUEBA DE ESTUDIANTE #3

Con esta prueba se pretende validar la implementación del intérprete, de las gráficas y el código privado.

PASOS

Ve al sitio

Busca la publicación

Es un ejercicio avanzado de bachillerato, se titula *Determinar cantidad de raíces reales en una función*, su autor es el docente *MateFun*.

Determinar cantidad de raíces reales en una función cuadrática

Responder

Nombre y apellido:*

Institución:

Código privado:

Las funciones de segundo grado, también llamadas cuadráticas, son de la forma:

$$f(x) = ax^2 + bx + c,$$

donde el coeficiente a es distinto de 0.

Cuál es el grado de una función cuadrática? (introduce solo el número)

La fórmula de bhaskara, nos permite hallar raíces de la función, y es de la forma:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

El discriminante, que es de la forma:

$$\Delta = b^2 - 4ac$$

Nos da información acerca de las raíces de la función:

- Si es >0 , entonces las dos raíces son reales
- Si es $=0$, entonces las dos raíces son iguales (llamadas raíces dobles)
- Si es <0 , no tiene raíces reales.

Por ejemplo, la siguiente función, tiene

$$x^2 + 4x$$

La siguiente función tiene

$$x^2 + 1$$

Por último, esta función tiene

$$\frac{8}{3}x^2$$

1) Prueba definir las funciones en MateFun y graficarlas con el comando ?grafica para verificar tus resultados

2) Usando el conjunto *cantidad*, define una función *cantRaices* que dada una terna de números reales a, b, c que son coeficientes de una ecuación de segundo grado (se supone el primero distinto de 0), *cantRaices* determina si la ecuación tiene 0, 1 o 2 raíces reales



Docente: MateFun

Institución: FING

Estudiar las funciones cuadráticas y sus raíces, utilizando el determinante.

Tema principal: Funciones cuadráticas

Dificultad: Avanzado

Grado: Bachillerato

Fecha: vie. 04 jun. 2021

Con corrección automática

[Ver más publicaciones del autor](#)

CÓDIGO	GRÁFICA 2D	GRÁFICA 3D
CARGAR		
1	F1 :: R -> R	
2	f1(x) = ...	
3		
4	F2 :: R -> R	
5	f2(x) = ...	
6		
7	F3 :: R -> R	
8	f3(x) = ...	
9		
10	conj Cantidad = {Cero, Una, Dos}	
11		
12	cantRaices :: R X R X R -> Cantidad	
13	cantRaices(x,y,z) = ...	

Incluiremos el código MateFun en el envío de tu respuesta.

ENVIAR RESPUESTA

GUARDAR COMO BORRADOR

Volver

Responde la publicación

Introduce tu nombre real y una institución. Previamente has coordinado con el docente, y te da pedido introducir, y el código privado *Lic34-5B*

Completa los campos

Responde los campos y resuelve los puntos 1 y 2

Envía la respuesta como borrador

Guarda el código privado

Ve a la página principal

Intenta ingresar por código privado

Envía la respuesta y ve la corrección

FORMULARIO EVALUACIÓN

A todos los tipos de usuario se les envió un formulario de evaluación de uso, una vez finalizada las pruebas.

Los formularios se dividieron en dos, el de docentes y estudiantes se enfocaba en el uso de la plataforma en general, y las opciones que estas proveía, mientras que a los usuarios expertos se les envió además un segundo formulario que buscaba evaluar la implementación del intérprete y las gráficas, ya que por su experiencia pudieron hacer eso extensivo de ellas en las pruebas.

EVALUACIÓN DOCENTE Y ESTUDIANTE

Luego de la prueba de uso se les realizó una entrevista abierta donde se les pidió que evalúen el sitio en general, donde debían expresar si tuvieron dificultades al realizar a alguna tarea, y debían comparar el sitio con las plataformas a las que estaba habituado.

Se les envió también un formulario hecho en *Google Forms*¹, que buscaba medir de manera cualitativa y cuantitativa su experiencia con la plataforma, a través de preguntas de escala (Siendo 10 lo mejor), preguntas de múltiple selección, y preguntas abiertas.

El mismo contaba con las siguientes preguntas:

¿Qué tan fácil es usar la plataforma en general? [Escala 1-10]

¿Qué tan intuitivo es el sitio? [Escala 1-10]

Se busca evaluar la facilidad y qué tan intuitivo es el sitio. Estas preguntas se complementan con las observaciones realizadas durante la prueba, donde se pudo observar en cuáles acciones los sujetos dudaban o se trancaban en su operativa.

¿Qué le cambiarías o que le agregarías al sitio? [Pregunta abierta]

Se busca que expresen su opinión acerca del sitio y sugieran cambios, que se esperan estén basados en su comparación con los sitios que utilizan y las carencias que detectaron en la plataforma.

¿Te encontraste con alguna dificultad para utilizar algún menú o al buscar alguna opción? [Pregunta abierta]

Se evalúa la disposición de títulos, menús y opciones en la plataforma.

¿Qué partes necesitan mejorar? [Pregunta múltiple opción]

1. Editar un ejercicio, me fue incómodo utilizarlo
2. Me costó encontrar lo que quería hacer
3. El sitio en general, es distinto a mis plataformas habituales (por ejemplo CREA)
4. Responder un ejercicio, me costó encontrarlo y/o resolverlo
5. Los códigos privados de ejercicio, no entendía para qué servían o como usarlos
6. Los campos de múltiple opción y de autocompletar, no se entienden bien
7. Las publicaciones borrador, no me quedaron claras

¹ Formulario para docentes y estudiantes: <https://forms.gle/vTxGHJtpqp68yQj8>;
Formulario para expertos <https://forms.gle/iaCYeiwJGjDrarFx5>

Para ayudar al sujeto a expresar su opinión del sitio y darle ideas sobre posibles cambios, se le ofrece una lista de posibilidades, con opción de agregar más. Se menciona como ejemplo CREA ya que es la plataforma más conocida por docentes y estudiantes

¿Qué tan útil fue el manual provisto? [Escala 1-10]

¿Qué le cambiarías al manual provisto? [Pregunta abierta]

Estas últimas dos preguntas sirven para evaluar el manual, el cual les fue enviado previo a la reunión.

EVALUACIÓN DE EXPERTOS

Se les realizará además de la encuesta docente, una encuesta de expertos que evalúa las implementaciones específicas de MateFun.

Gráficas 2D y 3D

¿Qué opinas de la implementación de gráficas 2D y 3D. *

¿Agregarías alguna opción? Quitarías opciones? ¿Crees que algo es confuso?

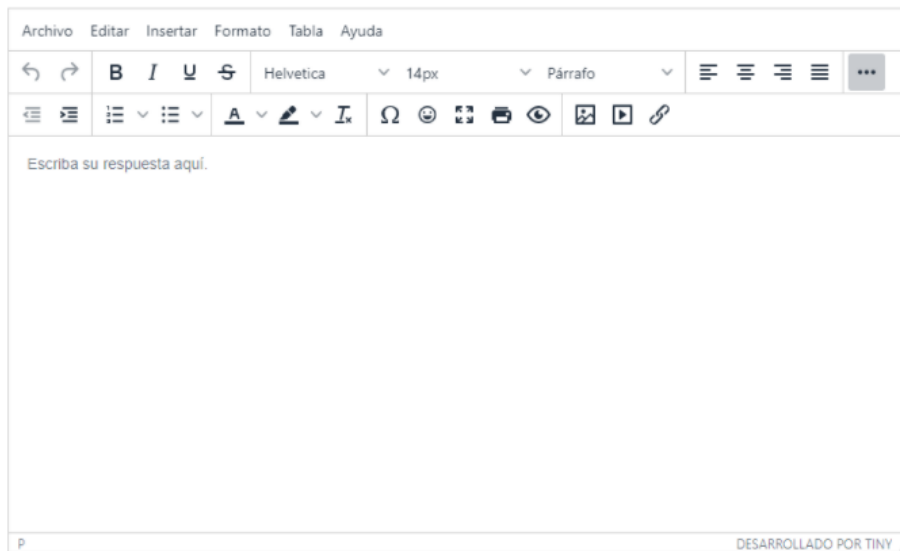


Se busca evaluar la implementación de las gráficas, ver si hacen falta nuevas funcionalidades y si la disposición de los elementos resultaba cómoda y e intuitiva

Editor de texto

¿Qué opinas del editor de texto de [MateFun?](#) *

¿Cumple su funcionalidad? ¿Le faltó algo?



Se busca evaluar el editor de texto adoptado y ver si tenía todas las opciones que creían necesarias y útiles para la plataforma educativa Emmy. Si bien esta pregunta no estaba específica en el formulario para los docentes, se les preguntó en la entrevista.

Intérprete

¿Qué opinas de la implementación del intérprete de [MateFun?](#) *

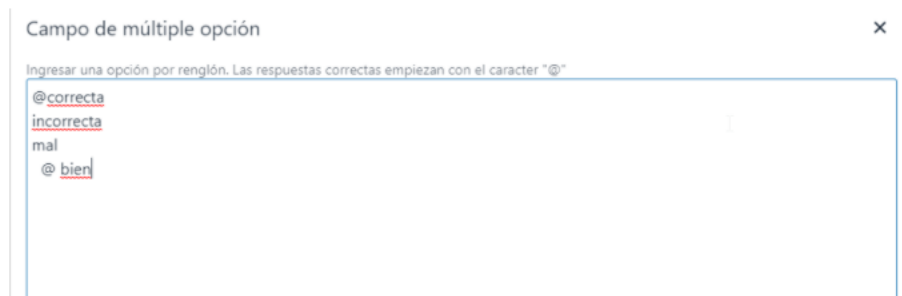
```
|||||C|||||_||| |||||
||| ||\||\||\||\ ||\|||

Error: {archivo: Matefun_hi3il27blghocoj36alu34vh8g linea: 2 columna: 24}
Se encontro ".", pero se esperaba expresion
Cargando Matefun_hi3il27blghocoj36alu34vh8g
Matefun_...>sumaAutoPotencia(2)
2
Matefun_...>?ayuda
Comandos del Interprete
!salir          salir
!cargar <programa>  cargar un programa
!recargar       volver a cargar el programa actual
?funcs          lista las funciones
?vars           lista las variables y sus valores
?conj          lista los conjuntos
?fun <funcion>    despliega el dominio y codominio de una funcion
?var <variable>  despliega el valor de una variable
?conj <conjunto> despliega la definicion del conjunto
?grafica <funcion> grafica una funcion de R -> R
?ayuda         despliega este mensaje de ayuda
Matefun_...>
```

Se evaluará la implementación del intérprete en la plataforma.

Campos múltiple opción y de completar: *

¿Qué opinas de los campos de múltiple opción y de completar? ¿Son intuitivos y relativamente cómodos?



Campo de múltiple opción x

Ingresar una opción por renglón. Las respuestas correctas empiezan con el caracter "@"

@correcta
incorrecta
mal
@ bien

Una característica importante y nueva eran los campos de completar y múltiple opción. Se les pregunta acerca de la utilidad y la facilidad de uso.

Del lado del estudiante; ¿crees que falta alguna funcionalidad? ¿Hay algo que modificarías?

Busca evaluar si faltó incluir alguna funcionalidad que creen que usarían sus estudiantes, y si se les ocurrió alguna idea nueva útil

¿Usarías la plataforma para enseñar? Explícanos por qué o por qué no la usarías

Evaluar la plataforma en forma general, donde interesa conocer los puntos más fuertes a favor o en contra.

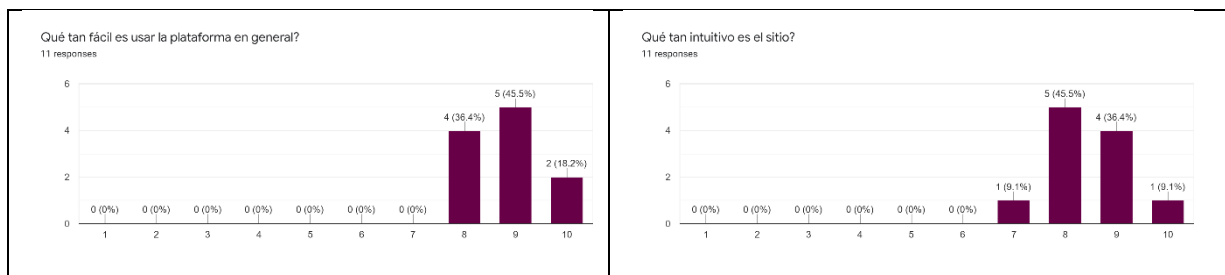
RESULTADOS

Se presentan los resultados de las encuestas

DOCENTES Y ESTUDIANTES

Se tuvieron 11 respuestas a la encuesta por parte de estudiantes y docentes.

Facilidad de uso



Todos los usuarios encontraron la plataforma fácil de usar e intuitiva.

¿Qué le cambiarías o qué le agregarías al sitio?

Respuestas:

- *Más información al pasar el ratón sobre algunas herramientas.*
- *Agregaría una devolución de nota y de un comentario al estudiante.*
- *Botón Login más visible.*
- *La presentación de los ejercicios en formato "tarjetas" se complica cuando son muchos. No me di cuenta de cómo ordenarlos o listarlos de otra forma*
- *Un botón de vuelve al inicio*
- *Agregar comentarios en las tareas*
- *Un cambio favorable a resolver sería que cuando envió una respuesta de un ejercicio, sin activar el reCAPTCHA previo, las respuestas son borradas.*
- *Por el momento no le cambiaría ni le agregaría nada. De hecho, la plataforma... en comparación a la que debo utilizar usualmente (Crea) tiene una estructura más organizada.*
- *Mejoraría el buscador de publicaciones.*
- *Quizás que pudiese organizarse las publicaciones y actividades por grupos o clases. También al docente tener una visión general por grupo o clase. Eso ayudaría a los estudiantes a no tener que buscar!!!!*

Las recomendaciones más importantes se relacionaron con nuevas funcionalidades para las respuestas de estudiante.

Te encontraste con alguna dificultad para utilizar algún menú o buscar alguna opción?

- *Con alguna herramienta que no indicaba para qué era.*
- *Tuvimos dificultades con el buscador de publicaciones.*
- *No conseguí visualizar la solución de "estudiantes" usando el código Primero-A*
- *no, en general resulta bastante fácil*
- *No estaba el tema que buscaba para publicar el ejercicio*

- *Me resulta bastante fácil buscar los ejercicios a través del buscador y en la sección de temas.*
- *Al contrario; logré encontrar lo que buscaba de manera eficaz y rápida. Si continuo con la comparación que comencé anteriormente, puedo agregar que (Emmy) tiene un menú de búsqueda más sencillo, (e igualmente eficiente), que la plataforma Crea.*
- *La única dificultad fue que no funciono el código cuando quise continuar un ejercicio.*
- *No encontré rápidamente el botón Login. El buscador de publicaciones no funcionaba. No vi un sistema de búsqueda por filtros o palabras clave que podría ser útil.*

Por lo general, las dificultades estuvieron en encontrar una funcionalidad o botón

Qué le cambiarías al manual provisto?

- *Al leer el manual antes de usar la plataforma, no entendí mucho qué estaba explicando. En general uso los manuales para resolver problemas, no para familiarizarme con la plataforma.*
- *Pude interpretar el manual sin dificultad.*
- *es bastante claro el manual, pero aclaré más personalmente, como por ejemplo darle el código al estudiante si lo perdía*
- *Una imagen explicando las distintas funciones de la plataforma*
- *Le agregaría más imágenes que acompañen la descripción de cada pantalla para ir familiarizándonos con el sitio*
- *Sin ver el sitio, el manual no me aportó mucho. En general uso el manual para resolver algún problema puntual.*

Los usuarios manifestaron que el manual lo usarían para resolver dudas, y que les gustaría algunas imágenes más.

EXPERTOS

Hubo dos usuarios expertos, (con conocimiento en MateFun) que realizaron las pruebas de forma independiente sin embargo, se mantuvo comunicación abierta con ambos durante la realización de la prueba.

Los expertos calificaron la implementación como completa y útil

¿Qué opinas de la implementación de gráficas 2D y 3D.

- *Están muy bien. No encuentro diferencia con el intérprete normal. Es fácil de ver y usar.*

¿Qué opinas del editor de texto de Emmy?

- *Completo. Suficiente. Cumple su funcionalidad*

¿Qué opinas de la implementación del intérprete de MateFun?

- *Sería bueno que de alguna advertencia cuando está deshabilitado o caído*

Campos múltiple opción y de completar:

- *Me parece que están bien.*

Del lado del estudiante; ¿crees que falta alguna funcionalidad? Hay algo que modificarías?

- *Poniendo consignas claras creo que no es necesario añadir nada más para el estudiante. Tal vez estudiaría la posibilidad de registrar un historial de actividad del estudiante con algunas estadísticas. Gamificar un poco, etc. Estaría bueno que cuando el estudiante resuelve un ejercicio salga de la vista de ejercicios por hacer y pase a su lista de ejercicios hechos.*

¿Usarías la plataforma para enseñar? Explícanos por qué o por qué no la usarías

- *Sí, la usaría. Me parece genial poder lanzar problemas por este medio. No solo para enseñar MateFun sino para enseñar cualquier cosa que se pueda enseñar con MateFun*

Documento de implementación

UNIVERSIDAD DE LA REPÚBLICA

FACULTAD DE INGENIERÍA

Documento de implementación

Autores:

Verónica BENTANCOR

Gastón BARRETO

Supervisores:

Sylvia DA ROSA

Marcos VIERA



**FACULTAD DE
INGENIERÍA**

Universidad de la República

12 de agosto de 2021

Índice general

1. Introducción	2
2. Implementación	3
2.1. Equipo de desarrollo	3
2.2. Ruby on Rails	5
2.3. Base de datos	6
2.4. Reutilización de código y refactorización	8
2.5. Arquitectura MVC	8
2.6. Pruebas de código	26
2.7. Búsqueda de publicaciones	31
2.8. Integridad de datos, validación y seguridad	32
2.9. Endpoints	36
2.10. Frontend	40
2.11. Editor de código e interprete	44
2.12. Gráficas	47
2.13. Editor de texto enriquecido	48
2.14. Ejercicios interactivos	50
2.15. Almacenamiento de imágenes	60
2.16. Fechas y zonas horarias	61
2.17. Paginación de resultados	63
2.18. Código Secreto	63
2.19. Internacionalización	64
2.20. Breadcrumbs	65
2.21. Listas desplegadas autocompletables	66

Introducción

En este documento se presenta la documentación de implementación de la plataforma educativa Emmy. Contiene información más detallada acerca de su funcionamiento interno, decisiones de diseño y las tecnologías utilizadas.

Implementación

Se realizó el desarrollo del sitio utilizando Git, en el servidor de Gitlab de la Facultad de Ingeniería ¹. Como ambiente local se utilizó una instalación de Ubuntu server 20.04 LTS y como ambiente de producción se utilizó la nube de Heroku, este fue el lugar además donde se presentaron las versiones a los tutores y se llevaron a cabo las pruebas de usuario.

2.1. Equipo de desarrollo

Los conocimientos del equipo de desarrollo determinaron la elección de tecnologías. El equipo tenía conocimientos básicos de Ruby on Rails adquiridos en Proyecto de Ingeniería de Software, donde se implementó un Backend, específicamente, una API para una aplicación móvil. No se tenía conocimientos sobre programación Frontend aunque se conocía los fundamentos de funcionamiento de JavaScript. En Programación 4 se había trabajado levemente Bootstrap y Java (Web services), en Fundamentos de Base de Datos se llegó a utilizar PostgreSQL, además uno de los miembros había cursado Interacción Persona Computadora, lo que facilitó elecciones para el diseño. No se tenía experiencia laboral en desarrollo.²

La metodología de desarrollo utilizada fue basada en prácticas Ágiles, haciendo el desarrollo de manera incremental e iterativo. Se partió de historias de usuarios en lenguaje natural, las cuales pasaron a ser *incidencias* de GitLab (Figura 2.1).

En cuanto al flujo del trabajo se utilizó GitFlow [5], subiendo Merge-request a la plataforma (estos tenían en general, la *incidencia* asociada). Los *commits* se trataron de mantener concisos e informativos, como muestra la Figura 2.2.

¹<http://gitlab.fing.edu.uy/>

²Hasta haber terminado prácticamente la etapa de implementación










Filtrar soluciones por código de estudiante #44 · opened 3 months ago by Veronica Dahiana Bentancor Cedrez Docente Historia	CLOSED  0 updated just now
Cambiar contraseña de otro usuario siendo administrador #106 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 1 week ago
Listar administradores #105 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar publicaciones en perfil administrador #104 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar docentes en panel de administracion #103 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Ordenar docentes por substring en consola de administracion #81 · opened 2 months ago by Gaston Daniel Barreto Sugliani Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar temas en perfil administrador #102 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Administrador Historia	CLOSED  0 updated 2 weeks ago
Filtrar respuestas de publicación #101 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Docente Historia	CLOSED  0 updated 2 weeks ago
Quitar filtro autor en perfil docente #100 · opened 2 weeks ago by Veronica Dahiana Bentancor Cedrez Corección	CLOSED  0 updated 2 weeks ago

Figura 2.1: Historias de usuario escritas como *incidencias* de GitLab

Las historias de usuario definieron el desarrollo de Emmy de manera incremental, definiendo *sprints* de duración variable (entre dos y tres semanas), intercalados con reuniones con los tutores para mostrar y validar el trabajo realizado, así como definir y priorizar las siguientes historias de usuario a implementar.

Para lograr un historial de desarrollo más limpio, se deshabilitaron los Merge-commit³ (Figura 2.2), se realizaron *squash* a los commits de un mismo Merge-request⁴.

Se definieron tres ambientes: producción (utilizado por Heroku), desarrollo (en el cual se realizaban las implementaciones) y testing (donde se corrían las pruebas) cada uno con sus configuraciones adecuadas a la tarea, los cuales ya venían configurados por el Framework.

Se mantuvieron dos ramas principales, *master* se actualizaba con las nuevas versiones mientras que *develop* recibía constantemente las nuevas funcionalidades.

Se configuró Heroku para utilizar el ambiente de producción, con las versiones de Emmy en la rama *master*, se le configuró además la base de datos postgresql y

³Se dejaron solo para la liberación de versiones.

⁴Sobre la segunda mitad de la etapa de desarrollo, al crecer el tamaño del proyecto. Para los cambios más pequeños, se permitió el uso de *commits* directos a la rama *develop*.






















24 Apr, 2021 5 commits		
	buscador de ejercicios por codigo Veronica Dahiana Bentancor Cedrez authored 3 weeks ago	876c1554  
	cambios en version de captcha dependiendo de la accion Veronica Dahiana Bentancor Cedrez authored 3 weeks ago	75091759  
	Editor matefun por defecto escondido para crear publicacion Veronica Dahiana Bentancor Cedrez authored 3 weeks ago	47179ad8  
	opciones de publicación Veronica Dahiana Bentancor Cedrez authored 3 weeks ago	f1cb2065  
	coreccion al graficar en 2d Veronica Dahiana Bentancor Cedrez authored 3 weeks ago	3bbd5477  
23 Apr, 2021 1 commit		
	Correccion de las rutas, de los filtros de busqueda y los breadcrumbs Gaston Daniel Barreto Sugliani authored 3 weeks ago	e1469486  
21 Apr, 2021 4 commits		
	Corregido ejercicios borrador que no se veían Gaston Daniel Barreto Sugliani authored 3 weeks ago	5fa1cac4  

Figura 2.2: Historial de *commits* de Git.

las variables de entorno para poder trabajar con reCAPTCHA.

Se realizaron pruebas de integración desde el Frontend manuales durante todo el desarrollo del proyecto y se definieron pruebas unitarias y de controlador automatizadas sobre el final del desarrollo.

2.2. Ruby on Rails

Se utilizó el Framework de código abierto de Ruby on Rails, el cual utiliza el lenguaje Ruby, un lenguaje enfocado al desarrollador de sintaxis clara y simple. Se hizo esta elección como Backend de Emmy por haber trabajado con esta tecnología en Proyecto de Ingeniería de Software, además de ser un framework web.

Ruby on Rails es un Framework maduro, que utiliza el patrón de diseño MVC de modelo-vista-controlador, cuya primer versión surgió en el año 2005 [11], y cuenta con una gran cantidad de librerías, llamadas gemas, de código abierto. Algunos sitios famosos que utilizan esta tecnología son Github, Slideshare, Twitch, Soundcloud, Twitter⁵ y Airbnb. Ruby on Rails trae por defecto una serie de con-

⁵En el pasado.

figuraciones ya realizadas, (su filosofía es de convención sobre configuración y de no repetir código), lo que reduce tiempo de desarrollo y trae muchas de las funcionalidades más comunes ya implementadas, además de simplificar el proceso de desarrollo, permite llegar más rápido a un mínimo producto viable MVP, siendo una elección muy popular entre *Startups* y que se adapta perfectamente al proceso de desarrollo ágil.

A pesar de traer consigo muchas configuraciones hechas por defecto, cambiarlas es muy simple. Para el caso de las bases de datos, por ejemplo, Ruby on Rails implementa el patrón *Active Record* con lo cual logra independizarse de la elección de base de datos, y cambiarla implica simplemente cambiar de librería, de esta forma, logra ser agnóstico.

Este Framework permite de manera muy simple, no solo atender consultas desde el Frontend web, sino también cumplir el rol de Backend de una aplicación móvil, sin necesidad de configuraciones extras ni rediseño de su lógica. Para el proyecto esto supone una gran ventaja a futuro, ya que permitiría la implementación de aplicaciones móviles, sin necesidad de desarrollar un nuevo Backend.

2.3. Base de datos

Se utilizó PostgreSQL como motor de base de datos para el ambiente de desarrollo y de producción (el cual se utilizó en la asignatura Base de Datos), mientras que para tener un ambiente más ligero, se utilizó SQLite en el ambiente de pruebas.

La decisión del uso de PostgreSQL se basa en que se utilizó la búsqueda completa de texto para el buscador de ejercicios, que no solo tiene un buen rendimiento [22], sino que también, utilizando ts vector, se implementó la normalización de palabras del lenguaje español. A diferencia de búsquedas del tipo *LIKE/ILIKE*, no solo el rendimiento es mayor, sino que además se puede indexar las consultas[15], por lo que se agregaron columnas específicas a nivel de base de datos para usarlas como memoria cache.

Migraciones

Ruby on Rails maneja el uso de migraciones para la base de datos. Cada vez que se quiere realizar un cambio a esta, se genera un archivo indicando el cam-

bio a realizar, identificado por una fecha y hora (Figura 2.3), siendo este archivo registrado por Git y parte del control de versiones.

Cada vez que un desarrollador quiere empezar a trabajar, si Ruby on Rails detecta migraciones nuevas, avisa automáticamente para que pueda realizarlas.

Teniendo esta forma de trabajo no solo ayuda en el proceso de desarrollo incremental, sino que el hacer y deshacer cambios se hace de manera sencilla y aislada. Poder identificar los cambios que ha sufrido la base de datos se reduce a analizar los distintos archivos de migración, además, facilita el desarrollo haciendo que todos los desarrolladores tengan la misma versión de la base de datos de manera sencilla y automatizada.

Sin embargo, debido al tamaño del equipo de desarrollo y al no contar con versiones en producción, en las primeras etapas de desarrollo se reutilizaron las migraciones de una misma tabla para evitar que estos archivos se volvieran muy numerosos, debido a que se desarrollaba de manera incremental y los requerimientos eran variantes gracias a la constante retroalimentación e introducción de nuevas funcionalidades, además de que el tamaño del equipo de desarrollo hacía fácil recrear manualmente la base de datos y no hay pérdida de información ya que no hay datos reales de producción. Una vez que se obtuvo una versión en producción con datos reales, se pasó a trabajar con migraciones independientes.

Name	Last commit	Last update
..		
20201227234813_devise_create_users.rb	Test restantes	2 weeks ago
20201230112206_create_grades.rb	Cambios en listados vacios, contadores de publicaciones ...	1 month ago
20201230112213_create_topics.rb	Borrado por lote de docentes, publicaciones y temas	1 month ago
20201230112218_create_difficulties.rb	Cambios en listados vacios, contadores de publicaciones ...	1 month ago
20201230112220_create_publications.rb	Precargar codigo del profesor	1 month ago
20201230112255_create_answers.rb	Guardar y recuperar respuesta borrador	1 month ago
20210128175114_add_searchable_to_publi...	Test de controlador y modelo	3 weeks ago
20210128175303_add_index_to_searchable...	Test de controlador y modelo	3 weeks ago
20210424121621_add_options_to_publicati...	opciones de publicación	1 month ago

Figura 2.3: Cada archivo de migración se relaciona con cambios específicos de una sola tabla

2.4. Reutilización de código y refactorización

Se siguió la filosofía de desarrollo de Ruby on Rails de convención sobre configuración, por lo cual se optó por seguir todas las convenciones de desarrollo del Framework, simplifica su configuración, mantenimiento y reduce los puntos de falla posibles, además permite a cualquier desarrollador experimentado leer, entender y continuar con el desarrollo de la plataforma de manera relativamente fácil.

La otra filosofía respetada fue DRY (Don't repeat yourself), implicó reutilizar la mayor cantidad de código posible, implementando funciones paramétricas que se adapten a varios requerimientos, (en vez de replicar código muy similar), esto ayuda a reducir errores, mantener el código limpio y mantenible, así como reducir el tamaño de métodos, clases y sus tests correspondientes.

Esta filosofía también contribuye a las decisiones de diseño del código, ya que implica por ejemplo seguir el patrón de *Fat model*, *Skinny Controller*, por lo cual toda la lógica de negocios se implementó en los modelos, mientras que los controladores se dejaron como nexo entre estos y las vistas.

Al finalizar cada funcionalidad, se le hizo una refactorización al código.

2.5. Arquitectura MVC

Modelos

Siguiendo la arquitectura MVC, los modelos contienen la lógica de negocio.

Figura 2.4 presenta el diagrama basado en el diseño de la base de datos, por simplicidad carece de índices y optimizaciones de PostgreSQL como el provisto por los ts vector.

Todos los modelos, a excepción del Respuestas, tienen implementado el borrado lógico, mediante la gema *paranoid* (ya que se pretende poder reutilizar los códigos secretos generados). Aunque no está implementada la recuperación de elementos borrados, estos se pueden acceder por base de datos y se podría implementar la funcionalidad agregándola a los controladores.

Difficulty: Modela la dificultad de los ejercicios. El orden por defecto es por nivel de dificultad, por lo cual se ordena por valor ascendente.

- Name

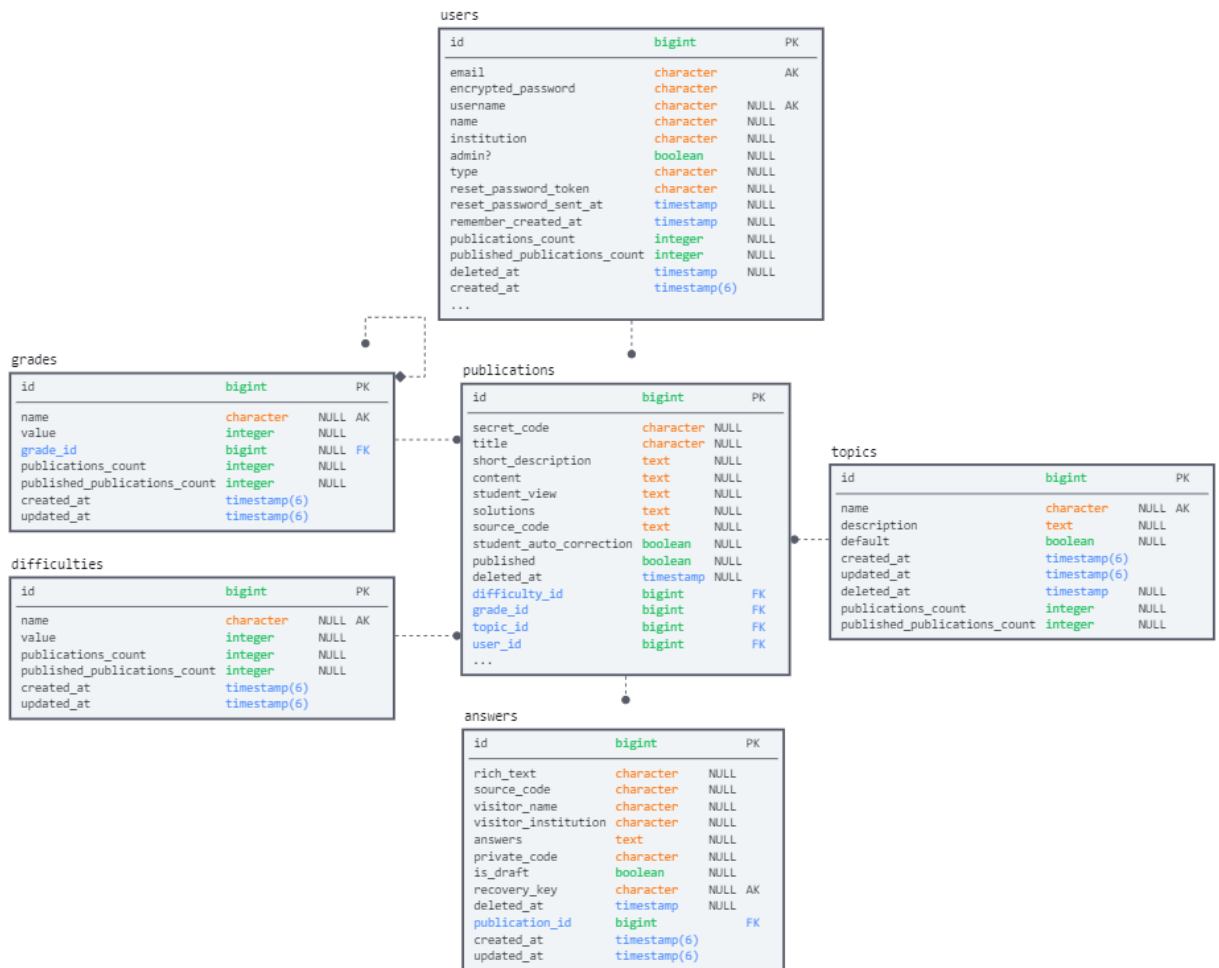


Figura 2.4: Diagrama basado en la base de datos

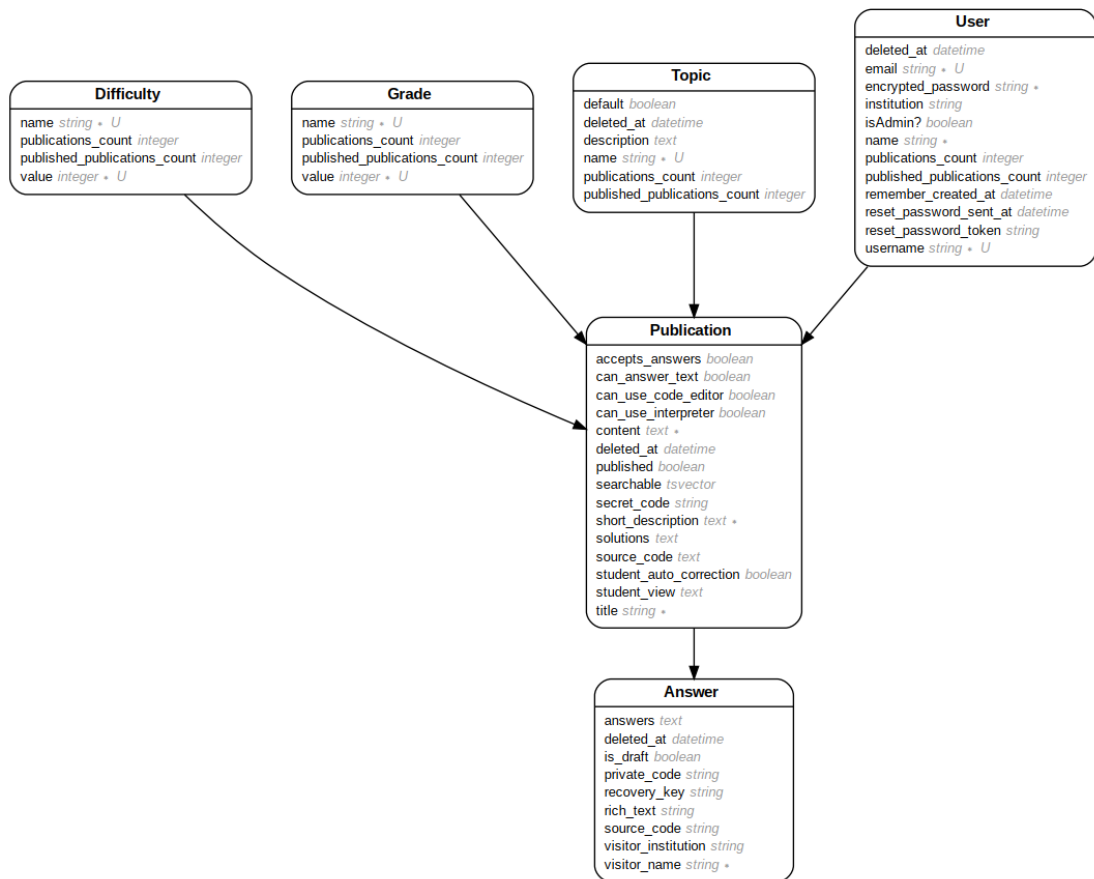


Figura 2.5: Definición de modelos

- Debe ser único
- No puede ser vacío.
- Largo mínimo de dos letras y máximo de cuarenta.
- Publications_count
 - Cantidad de publicaciones asociadas
- Published_publications_count
 - Cantidad de publicaciones publicadas asociadas
- Valor
 - Entero para ordenar la dificultad, de más fácil a más difícil.
 - Debe ser único
 - Es obligatorio

Grade: Modela los grados para los cuales son pensados las publicaciones. El orden por defecto es por año académico, por lo cual se ordena por valor ascendente.

- Name
 - Debe ser único
 - No puede ser vacío.
 - Largo mínimo de dos letras y máximo de cincuenta.
- Publications_count
 - Cantidad de publicaciones asociadas
- Published_publications_count
 - Cantidad de publicaciones publicadas asociadas
- Valor
 - Entero para ordenar el grado, de menor (escolar) a mayor (universitario)

- Se controla unicidad tanto a nivel de base de datos como a nivel de modelo
- Es obligatorio

Las dificultades y los grados se encuentran definidos por código y solo los desarrolladores pueden modificarlos. Para hacerlo, existe un archivo de *semillas* fácilmente editable.

Topic: Modela los temas de las publicaciones, cada una puede tener asociada solo un tema. Se encuentran precargados más de cien temas, que fueron extraídos de los temarios de enseñanza primaria, secundaria y terciaria del Uruguay. Los administradores pueden agregar, quitar y modificar temas desde el menú de administración, y los desarrolladores pueden modificar el archivo de semillas para modificar los temas precargados.

Cuando se borra un tema, se le asigna automáticamente, mediante un evento de disparo *before_destroy* a todas sus publicaciones relacionadas el tema por defecto. El tema por defecto no puede ser modificado por los administradores, y está implementado a nivel de código también en el archivo de semillas.

Por defecto, se listan de a 12 elementos.

- Default:
 - Indica si el tema es por defecto o no.
 - Solo puede existir un tema por defecto. Se controla a nivel de base de datos
 - Por defecto es falso
- Description
 - Se permite dejarla vacía o con un largo mínimo de tres letras y máximo de cien.
- Name
 - Debe ser único
 - No puede ser vacío.
 - Largo mínimo de dos letras y máximo de cuarenta.

- Publications_count
 - Cantidad de publicaciones asociadas
- Published_publications_count
 - Cantidad de publicaciones publicadas asociadas

Usuario: Implementado con la gema Devise. Los usuarios pueden ser docentes o administradores, esta herencia se implementó mediante STI de Rails, por lo cual todos los tipos de usuarios comparten la misma tabla de la base de datos y se diferencian por un atributo identificador.

Esto se debe a que la única diferencia a nivel de atributos de base de datos entre docentes y administradores es que los primeros pueden tener asociada una institución. Los usuarios pueden tener muchas publicaciones asociadas y por defecto se listan de a doce. Al borrar un usuario se borran de manera automática todas sus publicaciones.

- Email
 - Correo del usuario.
 - No puede ser vacío
 - Formato validado por Devise
 - Se controla unicidad
- Encrypted_password
 - Las contraseñas se guardan encriptadas en la base de datos
 - Para encriptar se utiliza OpenBSD bcrypt [23], al ser la implementación por defecto en Devise.
 - Debe tener un largo entre ocho y sesenta y cuatro caracteres, mayúsculas, minúsculas y al menos un símbolo.
- Institution
 - Institución de los docentes.

- Para evitar tener que administrar instituciones, cada docente puede escribir la suya en formato texto plano.
- Campo opcional, de largo máximo ochenta caracteres
- isAdmin?
 - Determina si es un administrador
- Name
 - Nombre completo de la persona
 - Se controla unicidad
 - Largo mínimo dos, largo máximo de ochenta caracteres
- Publications_count
 - Cantidad de publicaciones asociadas
- Published_publications_count
 - Cantidad de publicaciones publicadas asociadas
- Username
 - Nombre de usuario
 - Se controla unicidad
 - Largo mínimo dos, máximo cuarenta

La gema Devise agrega los campos:

- Remember_created_at
- Reset_password_at
- Reset_password_token

Publicación: Las publicaciones tienen una dificultad, un tema, un grado y un autor, así como muchas respuestas. Borrar una publicación borra automáticamente todas sus respuestas.

Cuenta con opciones booleanas que modifican las posibilidades de respuesta, por defecto todas valen verdadero.

- `Accepts_answers`
 - Se acepta el envío de respuestas
- `Can_answer_text`
 - Se acepta el envío de texto en las respuestas
- `Can_use_code_editor`
 - Se permite el uso del editor de código MateFun
- `Can_use_interpreter`
 - Se permite el uso del intérprete.
- `Student_auto_correction`
 - Se muestra la corrección automática a los estudiantes una vez que han enviado su respuesta como final.

Aquellas publicaciones que no acepten respuestas, no podrán tener habilitada la corrección automática para estudiantes, mientras que si no aceptan el uso del intérprete, no podrán habilitar el editor de código MateFun. Estas validaciones se realizan tanto a nivel de vista como de modelo.

Si una publicación permite el envío de respuestas y luego lo deshabilita, todas las respuestas ya enviadas podrán seguir siendo accedidas tanto por los estudiantes como por los docentes. Por otro lado, si se deshabilita por ejemplo el uso del intérprete o el ingreso de texto en las respuestas, este ya no será visible, aunque su contenido se sigue almacenando en la base de datos y estará disponible la próxima vez que se habilite la opción.

Las publicaciones además contienen:

- `Content`
 - Contenido de la publicación tal como lo ve el docente en el editor de texto. Contiene los campos de completar en formato input HTML y las respuestas de cada uno de ellos.
- `Published`

- Si está publicada o no una publicación.
- Solo puede ser cambiada a publicada y luego no puede ser modificada.
- Searchable
 - Utilizado por PostgreSQL para ponderar los atributos título, descripción corta y contenido, utilizando el lenguaje español como base.
- Short_description
 - Descripción corta, se muestra en las tarjetas de publicaciones
 - Largo mínimo de 20, máximo de 450
- Solutions
 - Cadena de texto en formato JSON que contiene los identificadores y las soluciones cada campo de completar
 - Es utilizado por el Answers para corregir las respuestas de los estudiantes.
- Source_code
 - Código fuente que se concatena por defecto al código fuente del estudiante cuando crea una respuesta por primer vez.
- Student_view
 - Es la vista HTML que se presenta a los estudiantes.
 - A diferencia del contenido, no contiene las soluciones a los campos de completar.
- Title
 - Largo mínimo de tres caracteres, máximo cien

El student_view se genera automáticamente en el lado del cliente cada vez que crea o actualiza la publicación. Resulta de quitarle las respuesta a los campos de completar, y de transformar los campos de múltiple opción de un input HTML

(que es como lo ve el docente en el editor TinyMCE) a elementos *select* de HTML (que es como se ve en la publicación).

Tanto los campos de contenido como la vista docentes son filtrados para evitar posibles ataques maliciosos, como la inyección de código de SQL u otro código maligno, utilizando la gema *Sanitize*.

Los contadores de publicaciones y publicaciones publicadas se implementaron utilizando los *counter cache* de *ActiveRecord*, de esta manera, las cantidades se encuentran almacenadas en la base de datos y la penalización de rendimiento al solicitar estos contadores (por ejemplo al listar temas, que se listan por defecto por cantidad de publicaciones publicadas) son insignificantes. Las publicaciones al ser creadas o actualizadas, actualizan los contadores de publicaciones de dificultad, temas, grados y usuarios, ya que la cantidad de veces (y el costo) que esto sucede es mucho menor a la cantidad de veces que se consultan los contadores.

Respuestas: Las respuestas son hechas por visitantes y pertenecen a una publicación. Aunque no se asocia la respuesta a usuarios registrados, se dejó la asociación hecha para poder implementar las funcionalidades, sin tener que modificar la arquitectura ni la base de datos, alcanza con modificar el controlador de respuestas. Se listan por defecto de a diez.

- Answers
 - Texto que contiene los identificadores y las respuestas a los campos de completar de las publicaciones
- Is_draft
 - Indica si es borrador o no.
- Private_code
 - código privado
 - Texto plano acordado entre docentes y estudiantes
 - Opcional
- Recovery_key

- código secreto.
- Se controla la unicidad a nivel de base de datos y a nivel de
- Rich_text
 - Contenido de la respuesta
- Source_code
 - Código fuente de MateFun.
- Visitor_institution
 - Al igual que a los docentes, es texto plano que contiene el nombre de la institución ingresada por el estudiante.
- Visitor_name
 - Nombre del estudiante
 - Obligatorio
 - Largo mínimo dos, máximo ochenta

Tanto el campo de texto enriquecido como el código fuente están presentes al usuario si la publicación tiene habilitada la opción al momento de crear o ver la respuesta. El campo de respuesta del estudiante es filtrado para evitar posibles ataques maliciosos, como la inyección de código de SQL u otro código maligno, utilizando la gema *Sanitize*.

Optimizaciones a nivel de base de datos

Además de las optimizaciones de búsqueda descritas en la Sección 2.7, se crearon índices (adicionales al índice por *id*) para acelerar las consultas a la base de datos, además, como se implementa el borrado lógico, la gema *paranoid* introdujo un atributo *deleted_at* del tipo índice a cada (ya que al listar, por defecto se excluyen los elementos borrados).

Los índices extras creados, incluyen:

Usuarios Se crearon índices de *email* y *username* ya que son los atributos por los cuales se inicia sesión. El resto de las consultas se hacen referenciando por *ID*.

Grados, Temas y Dificultades Se crearon índices por *nombre* (que además es único) y por cantidad de publicaciones publicadas y totales, ya que se existen filtrados por estos parámetros.

Publicaciones Se crearon índices por el atributo booleano *publicado*, ya que solo las publicaciones en ese estado son mostradas a los visitantes.

Respuestas Se crearon índices por código privado, *nombre* e *institución*, ya que se espera sean utilizados por los docentes para filtrar respuestas, y también por código secreto (que además es único), ya que los estudiantes van a recuperar soluciones a partir de este atributo, y al momento de generar un código, el modelo consulta la base de datos para validar la unicidad.

Datos precargados

Se definieron archivos de semillas, que son archivos diseñados para cargar datos por defecto en la base de datos, usando la sintaxis propia de Ruby on Rails.

Para el ambiente de producción se definieron en estos archivos los temas, los grados, las dificultades y el usuario administrador, mientras que para el ambiente desarrollo se agregan usuarios, publicaciones y respuestas de prueba.

Al tener configurados datos por defecto en un ambiente de desarrollo, se facilitan las pruebas al sitio, ya que este comienza con un juego de datos completos. Se crearon publicaciones y respuestas de largo variable, así como distintas combinaciones respuestas a campos interactivos.

En total se crean tres usuarios docentes, los cuales se distribuyen mil publicaciones de manera aleatoria, distribuidas en doce temas, donde para las quinientas primeras publicaciones se crean entre uno y quince respuestas de manera aleatoria.

Para cargar estos datos en un ambiente de desarrollo, alcanza con correr los comandos correspondientes, sin necesidad de interactuar con la base de datos, de la misma forma, al estar los datos de pruebas escritos en código, modificarlos no trae mayores complicaciones.

Controladores

La arquitectura de la aplicación sigue la el patrón de diseño MVC. Por lo tanto, las funciones de los controladores están ligadas a los modelos, y se mantuvo

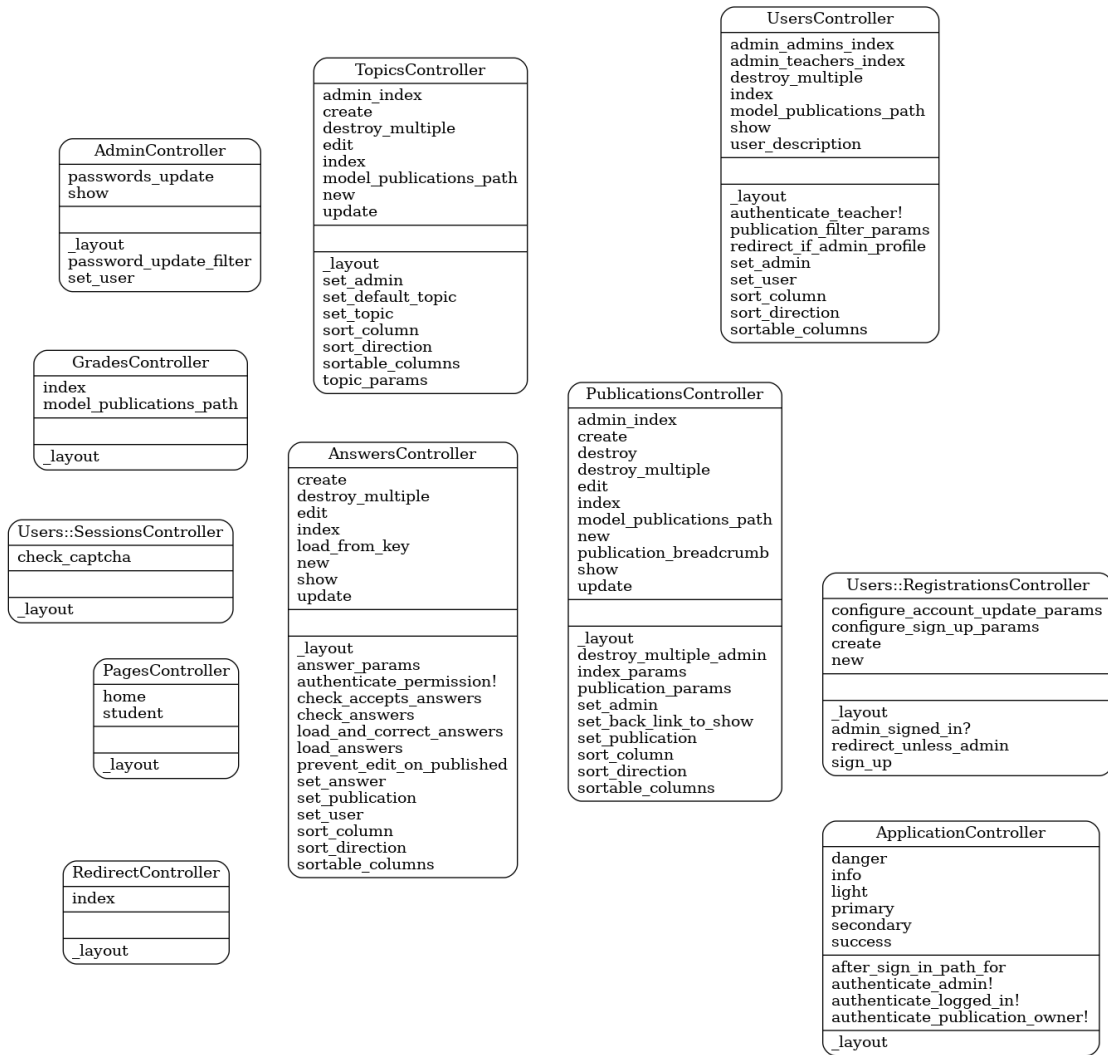


Figura 2.6: Definición de Controladores

la filosofía de Rails de Fat model, Skinny Controller.

Toda la lógica de negocio relacionadas a las clases se encuentra implementada en los modelos, y no los en los controladores, los cuales son meramente un nexo entre las vistas y los modelos, interactuando con los últimos para proveer de información a los primeros.

Contienen también toda la lógica esperable de un controlador, como la redirección de peticiones, filtro y manejo de parámetros y peticiones http, seguridad de acceso y validación. Los controladores también proveen *helpers*, que contienen lógica auxiliar para las vistas.

Como medida de seguridad, todos los controladores, según la acción a realizar, controlan la sesión del usuario y además filtran los parámetros recibidos, además de proteger de manera nativa contra todo tipos de ataques malintencionados, como por ejemplo los de inyección de SQL.

PagesController

Controlador sin modelo asociado. Muestra la pantalla de bienvenida y la de estudiantes.

RedirectController

Encargado del direccionamiento, realiza la misma cuando se ingresa solamente el código secreto de ejercicio en la URL.

ApplicationController

Provisto por defecto por Rails. Todos los demás controladores heredan de él. Define autenticaciones de alto nivel para administradores y docentes.

AdminController

Encargado de la interacción con las funciones específicas del modelo de administradores. Interactúan con las vistas para mostrar el panel de administración, las estadísticas de uso y la pantalla de cambio de contraseña.

- Mostrar perfil de administrador

- Consiste del menú de administración y las gráficas estadísticas
- Actualizar contraseña
 - Permite actualizar la contraseña de cualquier usuario.

GradesController

Encargado de interacción con las funciones específicas del modelo de grados, como de estos solo se permite la indexación, la interacción que tienen con las vista es solamente el Listado.

TopicsController

Encargado de la interacción con el modelo de temas, estas incluyen al CRUD de acciones (excepto el destruir, que se sustituyó por un borrado múltiple), y además se implementó una indexación especial para administradores (la cual acceden desde el panel de administración).

Se incluyen funcionalidades de administradores, las cuales validan al sesión, estas incluyen:

- Índice de administrador
 - Indexa los temas de forma tabular para los administradores.
 - Se incluyen además funciones para el ordenamiento y filtrado
- Crear
- Actualizar
- Borrado múltiple
 - Se valida que no se borre el tema por defecto

Para las acciones de crear, actualizar y borrado múltiple se controlan los parámetros. Las funcionalidades para visitantes incluyen:

- Indexar

- Indexa todos los temas para los cuales hay al menos una publicación en estado publicado, ordenando de manera descendente por cantidad de publicaciones publicadas

AnswersController

Encargado de la interacción con el modelo de respuestas. Incluye las funcionalidades del CRUD (excepto el destruir, que se sustituyó por un borrado múltiple).

Para los docentes autores de la publicación, incluye las funcionalidades:

- Indexar
 - Indexa las respuestas de forma tabular.
 - Provee funciones para el ordenamiento y filtrado, ordenando por fecha de modificación
- Mostrar
 - Permite ver cualquier respuesta asociada, sea un borrador o no.
 - Carga las respuesta a los campos de interacción
 - Corrige las respuestas de los campos de interacción, a través de su invocación en el modelo respuestas, para mostrar siempre la corrección automática
- Borrado múltiple

Mientras que para el resto de los usuarios incluye:

- Crear
- Editar
 - Se controla que no se permita la modificación de respuestas ya que no están en estado borrador
 - Siempre que se incluya el código secreto.
- Mostrar

- Siempre que se incluya el código secreto.
- Carga las respuestas de los campos de interacción
- Corrige las respuestas de los campos de interacción, a través de su invocación en el modelo respuestas, siempre que esté habilitada la opción de corrección automática
- Redireccionar a partir de código secreto
 - Como el código secreto es única, redirecciona a la misma en caso de existir.

Para las funcionalidades de crear y editar, se controla también invocando al Concern de *reCAPTCHA* la validez de la respuesta.

UsersController

Como se usa la gema de autenticación y manejo de usuarios Devise, se tienen controladores auxiliares, los cuales se encuentran separados de los implementados para el sitio. Sin embargo, todos estos controladores pueden ser reescritos y se puede reutilizar total o parcialmente su funcionalidad. Para este proyecto se optó por modificar lo menos posible su lógica, y, utilizando herencia, se sobrescribieron algunos métodos, que luego ejecutan el método padre, para aprovechar lo máximo posible la reutilización de código.

Las funcionalidades para administradores incluyen:

- indexar administradores
- indexar profesores
- Borrado múltiple

Para los indexar se incluyen ordenamientos por filas y filtros Para los docentes incluyen:

- Mostrar
 - Muestra el perfil del usuario, junto con su lista de publicaciones

- Provee funciones para el ordenamiento y filtrado, ordenando por fecha de modificación
- Permite acceder a las publicaciones, y las respuestas de estas

Sessionscontroller

Hereda del SessionsController de la gema Devise. Se encarga del manejo de la sesión, se le agregó un llamado a la verificación del reCAPTCHA (invocando el Concern correspondiente).

Implementado en la gema, incluye editar los campos de usuario. Se agregó la funcionalidad de seguridad, de filtrar parámetros prohibidos, como el campo que determina si es un usuario administrador.

RegistrationsController

Hereda del RegistrationsController de la gema Devise. Se encarga del registro de usuarios. Se modificó el filtro de parámetros para permitir aquellos que se fueron incluyendo en el modelo de usuarios, y se agregaron controles de permisos para las funcionalidades de usuarios registrados y administradores. Por último, se modificó para permitir que administradores pudieran (estando con su sesión iniciada) registrar a otros usuarios, y se deshabilitó el registro de usuarios por parte de los visitantes.

Implementado en la gema, incluye el registro de usuarios.

PublicationsController

Encargado de la interacción con el modelo de respuestas. Incluye las funcionalidades del CRUD. Se agregó también un destruir múltiple, un indexado para administradores y funciones auxiliares para las vistas.

Para administradores incluye:

- Indexa
 - Indexa todas las publicaciones, publicadas o no, de la plataforma, ordenadas por fecha de modificación

- Provee funciones para el ordenamiento y filtrado
- Borrado múltiple
 - borra todas las respuestas en cascada

Para los docentes incluye:

- Crear
 - Se puede crear como publicado o como borrador.
- Actualizar
 - Controla los parámetros recibidos, verificando que no se puedan despublicar publicaciones.
- Borrar
 - Borra todas las respuestas en cascada

Para los visitantes incluye:

- Indexar
 - Indexa todas las publicaciones publicadas, ordenadas por fecha de modificación
 - Incluye funciones para el filtrado

2.6. Pruebas de código

Durante todo el desarrollo del proyecto, desde sus primeras concepciones, se realizaron test de integración manuales desde el Frontend. Sobre la última etapa de desarrollo, luego de haber tenido un MVP y antes de presentar una versión a producción, se implementaron las pruebas unitarias y de controlador, esto se dejó para la última etapa ya que se buscaba llegar rápido al MVP para su validación.

Se realizaron test unitarios y de controlador en Minitest, para todas los modelos y los controladores desarrollados, exceptuando los provenientes de gemas,

como Devise o reCAPTCHA, o los *helpers* (utilizados desde el Frontend) y los *breadcrumbs*. Se utilizó además la gema *shoulda* para la escritura de un código más legible y mantenible por parte de los desarrolladores.

Los test de integración y las pruebas de Frontend (que incluyen HTML y JavaScript) se realizaron manualmente para las funcionalidades que se agregaron en cada Merge-request, con revisiones a fondo de todas las funcionalidades con cada liberación.

Se midió la cobertura de las pruebas utilizando la gema *simplecov* logrando una cobertura del 100 %⁶.

En Figura 2.7 se muestra la cobertura de código archivo por archivo, donde se tiene un promedio mayor a tres ejecuciones por línea.

Siguiendo con la convención de *Rails*, se escribió un archivo de *fixtures*, donde se definieron los datos de prueba a utilizar. Para cada modelo, se crearon datos suficientes como para cubrir las clases de equivalencia a probar, los cuales son compartidos con las pruebas de controladores. Se separó además el código fuente de las pruebas en contextos, utilizando la gema *Shoulda*, de esta forma, se lograron modularizar los distintos tipos de pruebas dentro de un mismo archivo. Definir contextos tiene como ventaja, además de una mayor legibilidad, que se pueden compartir variables entre todas las pruebas de un mismo contexto, donde estas variables se definen en un método *setup*, y luego de cada prueba sus cambios son descartados, manteniendo no solo cada test independiente entre sí, sino también reduciendo el uso de recursos y los tiempos de ejecución.

Pruebas unitarias

Para las pruebas unitarias de modelos se separaron las pruebas en contextos específicos para las asociaciones, las validaciones de campos, los listados y los métodos, de esta forma, se organiza mejor el código.

Para cada método a validar, se crearon casos de prueba a partir del estudio de clases de equivalencia.

La tabla muestra los casos estudiados para la corrección de campos interactivos.

⁶En controladores y modelos desarrollados

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/admin_controller.rb	100.00 %	76	37	37	0	1.35
app/controllers/answers_controller.rb	100.00 %	308	133	133	0	3.07
app/controllers/application_controller.rb	100.00 %	31	10	10	0	6.20
app/controllers/concerns/publications_controller_concern.rb	100.00 %	94	31	31	0	3.97
app/controllers/grades_controller.rb	100.00 %	14	6	6	0	1.17
app/controllers/pages_controller.rb	100.00 %	8	3	3	0	1.00
app/controllers/publications_controller.rb	100.00 %	256	104	104	0	2.94
app/controllers/redirect_controller.rb	100.00 %	23	11	11	0	1.64
app/controllers/topics_controller.rb	100.00 %	105	58	58	0	1.74
app/controllers/users_controller.rb	100.00 %	127	63	63	0	2.73
app/helpers/application_helper.rb	100.00 %	12	7	7	0	23.86
app/models/admin.rb	100.00 %	5	1	1	0	1.00
app/models/answer.rb	100.00 %	58	31	31	0	26.52
app/models/difficulty.rb	100.00 %	13	5	5	0	15.40
app/models/grade.rb	100.00 %	37	16	16	0	14.00
app/models/publication.rb	100.00 %	170	65	65	0	13.77
app/models/teacher.rb	100.00 %	5	1	1	0	1.00
app/models/topic.rb	100.00 %	28	13	13	0	2.31
test/controllers/answers_controller_test.rb	100.00 %	256	164	164	0	1.56
test/controllers/grades_controller_test.rb	100.00 %	21	11	11	0	1.09
test/controllers/pages_controller_test.rb	100.00 %	22	12	12	0	1.00
test/controllers/publications_controller_test.rb	100.00 %	320	167	167	0	1.87
test/controllers/redirect_controller_test.rb	100.00 %	25	17	17	0	1.00
test/controllers/topics_controller_test.rb	100.00 %	126	79	79	0	1.27
test/controllers/users_controller_test.rb	100.00 %	114	67	67	0	1.45
test/models/answer_test.rb	100.00 %	70	45	45	0	1.47
test/models/difficulty_test.rb	100.00 %	22	14	14	0	1.00
test/models/grade_test.rb	100.00 %	53	32	32	0	1.38
test/models/publication_test.rb	100.00 %	252	88	88	0	1.52
test/models/topic_test.rb	100.00 %	40	27	27	0	1.15
test/models/user_test.rb	100.00 %	56	42	42	0	1.14

Figura 2.7: Cobertura de código en modelos y controladores

Cuadro 2.1: Casos de prueba para las preguntas de completar

Casos de prueba		
Pregunta:	Respuesta Estudiante	Resultado
Abierta (Sin solución definida)	Vacía	Sin corregir
Abierta (Sin solución definida)	Con contenido	Sin corregir
Con tildes	Sin tildes	Correcta
Sin tildes	Con tildes extras	Correcta
Con espacios excesivos	Sin espacios	Correcta
Sin espacios	Con espacios extras	Correcta
Con solución definida	Respuesta correcta	Correcta
Con solución definida	Respuesta incorrecta	Incorrecta
Con solución definida	Vacía	Incorrecta

Pruebas de controladores

Para las pruebas de controlador se escribieron pruebas de rutas, se escribieron contextos basados en los perfiles administrador, usuario y visitante. En cada uno de estos contextos se escribieron pruebas que validaran las funcionalidades que estos tipos de usuarios deberían poder realizar y también se validó que no pudieran realizar acciones que no les correspondían.

En otras palabras, para las funcionalidades propias de los administradores, se validó en su contexto que pudieran realizarlas, y en los contextos de docente y visitante, que no pudieran llevar a cabo estas tareas que no les correspondían. Análogamente, para las tareas de docentes se validó que se pudieran realizar en el contexto docente y se controló que no pudieran hacerla los demás contextos. Se realizó lo mismo para los visitantes.

Para cada una de las acciones a validar que si se podían llevar a cabo, se probaron clases de equivalencia para asegurarse que estaban cubiertos todos los casos.

Calidad de código

Para controlar la calidad de código se utilizó la gema *Rubocop*⁷, la cual es un analizador de código estático. Se utilizó la configuración por defecto, que utiliza muchas de las reglas definidas en la guía de estilos de Ruby, y se configuraron excepciones en algunos métodos donde seguirlas, dificultaría la lectura o cohesión, por ejemplo, se desactivó el control de largo máximo de método o clase en los test, ya que implicarían dividirlo en muchos archivos, perdiendo cohesión.

A modo de demostrar la calidad del código alcanzada en el proyecto, se expresan algunas de las reglas más importantes que incluye la guía de estilos [12]:

- Se mantuvo el largo de línea máximo de ochenta caracteres.
- Se utilizaron nombres de variables en inglés.
- No se superaron las 100 líneas de código por clase.
- No se superaron las diez líneas de código por método.
- No se superó el límite de asignaciones, condicionales y flujos por método.

Por otro lado, se siguieron las reglas de convención de Ruby On Rails, también exigidas por Rubocop, entre ellas:

- Se nombraron los archivos HTML de forma que coincidan con el nombre de los métodos del controlador correspondiente.
- Se nombraron las clases y los controladores siguiendo el formato de Camel-Case en singular
- Se nombraron las variables y los atributos en formato snake_case
- Los métodos del tipo booleano terminan en `?`
- Los métodos destructivos o que arrojan excepciones terminan en `!`
- Las gemas deben estar listadas en orden alfabético

Al seguir estas guías (junto con todas las exigidas por Rubocop) no solo se respetó las convenciones del lenguaje sino también se logró un código de calidad, mantenible y legible para futuros desarrolladores.

⁷<https://github.com/rubocop/rubocop>

2.7. Búsqueda de publicaciones

La búsqueda de publicaciones en Emmy hace uso de las funcionalidades de búsqueda de texto completo de PostgreSQL y el uso de ts vector. Este tipo de datos está relacionado con los lexemas, a lo cual se lo configuró para usar el lenguaje español como base.

Con esta configuración, al realizar la búsqueda las palabras se llevan a su raíz lingüística, los plurales y las conjugaciones se transforman en su palabra raíz, mientras que palabras muy utilizadas, como por ejemplo *a*, *y*, *o*, son ignoradas, para evitar falsos positivos.

Así por ejemplo, si una persona busca “*hallé los números primos*”, van a aparecer en los resultados publicaciones que tengan esas palabras en cualquier forma de conjugación, por ejemplo “*hallar el número primo*”. Esto da una gran versatilidad y poder a las búsquedas de publicaciones, al mismo que las penalizaciones de rendimiento son mucho menores a implementaciones más simples, como las alcanzables con clausuras *LIKE/ILIKE*.

Se agregaron también índices de búsqueda utilizando *GIN* y algoritmos concurrentes para las publicaciones.

Se implementó además una búsqueda ponderada por atributo. Al momento de realizar una consulta, el mayor peso lo tienen las ocurrencias en el título, seguido por las ocurrencias de la descripción corta, y por último el contenido. Esto se hace así porque una ocurrencia de la palabra *primos* en el título, es un mejor indicador del contenido de la publicación que la misma ocurrencia en el cuerpo del mismo.



Figura 2.8: Búsqueda de números primos

La Figura 2.8 muestra la búsqueda *hallaron los numeros PRIMOS*⁸ en el sitio, por lo cual, la implementación encuentra la raíz lingüística de *hallaron* (*hallar*), transforma los plurales en singulares e ignora las mayúsculas, realizando la búsqueda por *“hallar número primo”*. Encuentra tres publicaciones, las cuales pondera su peso según dónde aparece la ocurrencia, las cuales son, en el título, en la descripción corta y en el cuerpo respectivamente.

Si esta misma búsqueda se hubiera implementado con soluciones más simples, como las de *LIKE/ILIKE* ya mencionadas, no se hubieran obtenido resultados, ya que la conjugación de *hallar*, el plural de las palabras, el tilde de *números* y las mayúsculas de *PRIMOS* hubieran sido incompatibles con los campos de la publicación. Debe aclararse además, que los mismos resultados se hubieran obtenido para cualquier permutación de las palabras buscadas, tanto en la publicación como en el texto de búsqueda, algo imposible de hacer con las alternativas (ya que buscan subcadenas). Además de todo esto, tampoco se hubiera ponderado el resultado y el costo de ejecución hubiera sido mayor.

En conclusión, se implementó una búsqueda de texto completo que tiene en cuenta las conjugaciones del lenguaje, que simplifica las búsquedas por parte de los usuarios, evita falsos positivos, provee unos resultados ricos y ponderados según donde aparezcan las palabras consultadas, al mismo tiempo que ofrece grandes ventajas de rendimientos, gracias no solo a las capacidades de búsqueda en sí del motor de base de datos y a la implementación de memoria cache de esta. Si a esto se le agrega la paginación que implementa la gema *Will-paginate*, la cual solo extrae la base de datos la cantidad de elementos a mostrar en pantalla [20], se tiene una búsqueda muy completa, inteligente, rica y de muy bajo consumo frente a otras alternativas más simples.

2.8. Integridad de datos, validación y seguridad

Formularios

Para proteger contra ataques indeseados, cuando un usuario está frente a un formulario HTML, por ejemplo al crear una publicación o responder a una, Rails crea un token de autenticidad aleatorio, lo almacena en la sesión y lo embebe de

⁸Observar la falta del tilde y la mayúscula.

manera oculta en este formulario. En el momento que el usuario envía este formulario, Rails valida el token guardado en la sesión con el enviado en el formulario HTML.

Validación de campos

Las validaciones de los campos se realizan a nivel de cliente utilizando las nuevas funcionalidades de HTML 5 y a nivel de servidor utilizando funcionalidades de Ruby on Rails.

En el caso de las validaciones HTML 5 se utilizan los atributos requerido (*required*), de largo mínimo y máximo (*minlength*, *maxlength*), de valor máximo y mínimo (*min*, *max*), de tip (*type*) útil para la validación de correos y patrones de *regex* (*pattern*), el cual se usó para validar contraseñas.

Algunas validaciones están respaldadas por controles JavaScript, para el ingresar el nombre de usuario por ejemplo, el sitio no inserta espacios en el campo, mientras que además por HTML5 y a nivel de modelo se agregan controles de largo.

Además de implementar validaciones en los modelos, se cuenta con verificaciones de unicidad y existencia para los campos en la base de datos. Este diseño en múltiples capas mejora la integridad de los datos y protege contra ataques externos.

Como medida de seguridad, todos los controladores, según la acción a realizar, controlan la sesión del usuario y además filtran los parámetros recibidos, además de proteger de manera nativa contra todo tipos de ataques malintencionados, como por ejemplo los de inyección de SQL.

Para lograr un sistema más robusto, todas las validaciones de unicidad se realizan tanto a nivel de base de datos, a nivel de modelos y a nivel de vistas. Las unicidades a nivel de cadena de caracteres se controlan ignorando las mayúsculas. El resto de las validaciones (como los largos de los campos o su obligatoriedad), al pertenecer la lógica de negocios y ser más susceptibles a cambios, se realizan solo a nivel de vistas y de modelo.

Sanitización de contenidos

El editor *TinyMCE* por defecto protege frente a la introducción de código malicioso, encapsulando todo el contenido del usuario en etiquetas de párrafo HTML.

Sin embargo, aunque el editor de texto filtra los contenidos, nada evita un ataque en el cual se sustituya el contenido de este por otro código malicioso (ya sea editando el código fuente HTML o realizando una petición HTTP independiente). No pudiendo depender solo de la buena intención del usuario, y de los filtros a nivel cliente que el editor provee, fue necesario realizar controles de contenido del lado del servidor.

Por defecto, Rails protege contra todo tipo de ataques de inyección de código y los *sanitiza* al momento de manipularlos. Este filtro sin embargo no era útil al mostrar el contenido de las publicaciones, ya que el editor introduce código HTML para darle estilo a la publicación, y es necesario mostrar este código. Para resolver este problema, se utilizó la gema *sanitize*⁹, la cual define por defecto una lista blanca de atributos permitidos.

Se definió la sanitización para filtrar el contenido introducido por los usuarios y que este sea luego introducido junto con el resto del código HTML del sitio, sin que este pueda ser indebidamente utilizado.

Además de la configuración por defecto, se estudiaron las etiquetas producidas por el editor de texto enriquecido y se configuró la sanitización de la gema para permitir ciertas etiquetas con ciertos atributos.

Para el contenido de las publicaciones escritas y mostradas a los docentes, se utilizó la configuración por defecto *Relajada*, la cual incluye imágenes, tablas y estilos considerados seguros. Los links se limitan a *FTP*, *HTTP*, *HTTPS*, y protocolos *MAILTO*. Las imágenes se limitan a *HTTP* y *HTTPS*. El Listado 2.1 muestra la configuración utilizada para ambos campos.

Para poder utilizar todas las funcionalidades del editor, se agregaron además elementos *input*, *select* y *option* para poder introducir campos interactivos, así como *iframes* para poder embeber vídeos.

De las etiquetas *input* se permiten los atributos *id*, tipo (*type*), texto de muestra (*placeholder*) y el tipo de datos HTML5 que almacena las respuestas correctas (*data-soluciones*).

⁹<https://github.com/rgrove/sanitize>

En cuanto a las múltiple opción, se permite el atributo *id* de las etiquetas *select*, y los atributos valor (*value*), mostrar desactivado (*disabled*), y marcar seleccionado (*selected*) de las etiquetas *options* correspondientes a cada una de las opciones.

De los vídeos embebidos se permite la etiqueta *iframe* y los atributos de origen (*src*), de alto y ancho (*height*, *width*) y se permite la pantalla completa (*allowfullscreen*).

De las tablas (*table*) se permite definir un borde (*border*).

A la configuración por defecto de imágenes (etiqueta *img*) se la agregan los atributos de texto alternativo (*alt*), de título (*title*) y de origen *src* siendo los orígenes permitidos *http*, *https* y *data* (ya que las imágenes se almacenan en Base 64).

Para el contenido mostrado a los estudiantes, el cual es una conversión del contenido anterior, quitándole las respuestas correctas y convirtiendo los *inputs* del tipo múltiple opción en etiquetas *select*, también se realizó una sanitización, ya que es el cliente quien debe realizar la conversión y luego enviarlo al servidor.

```
1 Sanitize.fragment(content,
2   Sanitize::Config.merge(Sanitize::Config::RELAXED,
3     elements: Sanitize::Config::RELAXED[:elements]
4     + %w[input select option code iframe],
5     attributes: {
6       'input' => %w[id type placeholder data-soluciones],
7       'select' => ['id'],
8       'option' => %w[value disabled selected],
9       'iframe' => %w[height src width allowfullscreen],
10      'table' => ['border']
11    },
12    add_attributes: {
13      'img' => %w[alt title]
14    },
15    protocols: {
16      'a' => { 'href' => %w[ftp http https mailto] },
17      'iframe' => { 'src' => %w[http https] },
18      'img' => { 'src' => ['http', 'https', 'data', :relative] }
19    })
```

Listado 2.1: Definición de etiquetas y atributos permitidos

Con las respuestas de los estudiantes, como estos no tienen registro en el sitio, se fue más restrictivo con la sanitización y se utilizó la configuración *Relajada* del a gema.

No se permite el ingreso de ninguna etiqueta ni ningún atributo no permitido

por estas configuraciones descritas, siendo este filtrado tanto a nivel de Frontend (por TinyMCE) como a nivel de Backend (por la gema). Mientras que para el resto de todos los campos introducidos por los usuarios en la plataforma, se utiliza la sanitización por defecto de Ruby on Rails.

Contraseñas

Basándose en los fundamentos de los requisitos de contraseña listados en NIST SP 800-63-3 [4], el sitio requiere un largo entre 8 y 64 caracteres. También se pide que la contraseña contenga al menos una minúscula, una mayúscula y un símbolo. Las contraseñas se guardan encriptadas en la base de datos, utilizando OpenBSD bcrypt [23], al ser la implementación por defecto en Devise.

Esta validación se hace tanto en el lado del cliente (mediante validaciones HTML5, línea 2 del Listado 2.2) como del lado del servidor (línea 3 del Listado 2.3). Cualquiera de los mensajes de error se puede sustituir por etiquetas en el caso de implementar la internacionalización.

```
1 <input class="form-control"
2   autocomplete="Contraseña nueva"
3   pattern= "^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[#?!@$%^&*~]).{8,64}"
4   title='Debe tener largo entre 8-64, mayúsculas, minúsculas y símbolos'
5   type="password" name="user[password]" id="user_password"
6 >
```

Listado 2.2: Validación de contraseña del lado del cliente

```
1 def password_complexity
2   return if password.blank? || password
3   =~ /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[#?!@$%^&*~]).{8,64}$/
4
5   errors.add :password, 'debe tener un largo entre 8-64. Tener mayúsculas,
6   ↪ minúsculas y al menos un símbolo.'
7 end
```

Listado 2.3: Validación de contraseña del lado del servidor

2.9. Endpoints

Se definieron endpoints en el enrutador de Rails, anidando los recursos donde correspondía. Siguiendo la convención de Rails, a partir de las rutas se pueden inferir el controlador y el método correspondiente. Así por ejemplo, el endpoint de

la línea 4 del Listado 2.6 corresponde a un listado de publicaciones relacionadas con el tema *topic_id*, mientras que el endpoint de la línea 2 del Listado 2.8 corresponde con un listado de respuestas de la publicación *publication_id* del autor *user_id*.

Tener un enrutamiento de esta forma no solo jerarquiza la navegación sino que mantiene la calidad de código.

Como los controladores de Rails pueden configurarse para atender llamados no solo HTML sino también JSON, se podría implementar a futuro otro Frontend web, o una aplicación móvil, reutilizando toda la lógica ya implementada, es decir, alcanzaría con codificar las respuestas en formato JSON, funcionalidad ya nativa de Rails.

Como descrito en la Sección 2.6 realizaron pruebas de ruteo para validar el que el acceso estuviera disponible.

Generales

Estos definen las rutas de entrada al sitio (Listado 2.4)

```
1 GET /
2 GET /students
```

Listado 2.4: Endpoints generales

Usuarios

Definen las rutas de docentes y administradores (Listado 2.5). Esto incluye el inicio y cierre de sesión, así como la creación, modificación y eliminación de usuarios por parte de administradores.

```

1 GET    /users/sign_in
2 POST   /users/sign_in
3 DELETE /users/sign_out
4 GET    /users/password/new
5 GET    /users/password/edit
6 PATCH  /users/password
7 PUT    /users/password
8 POST   /users/password
9 GET    /users/cancel
10 GET   /users/sign_up
11 GET   /users/edit
12 PATCH /users
13 PUT   /users
14 DELETE /users
15 POST  /users
16 GET   /users
17 GET   /users/:id
18 POST  /admin/:admin_id/users/destroy_multiple
19 GET   /admin/:admin_id/teachers
20 GET   /admin/:admin_id/admins
21 POST  /admin/:admin_id/users/destroy_multiple
22 GET   /admin/:id/passwords_index
23 PUT   /admin/:id/passwords_update
24 POST  /admin/:id/passwords_update
25 GET   /admin/:id

```

Listado 2.5: Endpoints de usuarios

Temas y grados

En estos endpoints se encuentran el Listado de temas y grados (Listado 2.6), así como la modificación, Listado, borrado y creación de nuevos por parte de los administradores (líneas 3 a 9).

```

1 GET    /topics
2 GET    /grades
3 GET    /admin/:admin_id/topics
4 POST   /admin/:admin_id/topics/destroy_multiple
5 GET    /admin/:admin_id/topics
6 GET    /admin/:admin_id/topics/new
7 GET    /admin/:admin_id/topics/:id/edit
8 PATCH  /admin/:admin_id/topics/:id
9 PUT    /admin/:admin_id/topics/:id

```

Listado 2.6: Endpoints de temas y grados

Publicaciones

Los endpoints de publicaciones (Listado 2.7) incluyen el Listado o visualización de estas a partir de un tema (línea 1 y 2), un grado (líneas 3 y 4) o un docente (línea 5 y 6), así como el acceso general (líneas 7 a 14) y el acceso exclusivo para

administradores (líneas 15 y 16).

```
1 GET /topics/:topic_id/publications
2 GET /topics/:topic_id/publications/:id
3 GET /grades/:grade_id/publications
4 GET /grades/:grade_id/publications/:id
5 GET /users/:user_id/publications
6 GET /users/:user_id/publications/:id
7 GET /publications
8 POST /publications
9 GET /publications/new
10 GET /publications/:id/edit
11 GET /publications/:id
12 PATCH /publications/:id
13 PUT /publications/:id
14 DELETE /publications/:id
15 GET /admin/:admin_id/publications
16 POST /admin/:admin_id/publications/destroy_multiple
```

Listado 2.7: Endpoints de publicaciones

Respuestas

Define el endpoints de las respuestas (Listado 2.8), a las cuales se puede acceder a través de la publicación y el usuario (líneas 1 a 8), a través de la publicación en sí (líneas 9 a 14) o a través del código secreto (líneas 15 y 16).

```
1 POST /users/:user_id/publications/:publication_id/answers/destroy_multiple
2 GET /users/:user_id/publications/:publication_id/answers
3 POST /users/:user_id/publications/:publication_id/answers
4 GET /users/:user_id/publications/:publication_id/answers/new
5 GET /users/:user_id/publications/:publication_id/answers/:id/edit
6 GET /users/:user_id/publications/:publication_id/answers/:id
7 PATCH /users/:user_id/publications/:publication_id/answers/:id
8 PUT /users/:user_id/publications/:publication_id/answers/:id
9 POST /publications/:publication_id/answers
10 GET /publications/:publication_id/answers/new
11 GET /publications/:publication_id/answers/:id/edit
12 GET /publications/:publication_id/answers/:id
13 PATCH /publications/:publication_id/answers/:id
14 PUT /publications/:publication_id/answers/:id
15 GET /:recovery_key
16 GET /answer_from_key
```

Listado 2.8: Endpoint de respuestas

2.10. Frontend

El diseño del Frontend está diseñado en HTML, CSS y JavaScript utilizando JQuery *Bootstrap*. Se separó el código HTML del código JavaScript y CSS, de forma que facilite el desarrollo y lograr una aplicación modular.

2.10.1. Tecnologías y herramientas utilizadas

Bootstrap

Se utilizó Bootstrap 4 como base para el diseño del sitio. Esto tenía la ventaja de traer clases y estilos predefinidos, además de tener un soporte completo para dispositivos móviles y estar diseñado de manera *responsive*. Se consideró utilizar la última versión de Bootstrap (versión 5) pero se descartó por encontrarse en beta. Los iconos presentes en el sitio provienen de Bootstrap Icons ¹⁰ y son utilizados a través de un *CDN*.

Se separaron los archivos CSS según su procedencia, se tiene una archivo *bootstrap.css* que se modificó para lograr el estilo de la plataforma, pero en él solamente se encuentran las clases definidas por Bootstrap, el resto de los estilos se encuentra en archivos separados, de forma que si se desea cambiar de versión de Bootstrap o utilizar una plantilla de terceros, alcance con intercambiar el archivo.

reCAPTCHA

Se utilizó la gema de reCAPTCHA¹¹, simplificando el uso de reCAPTCHA de Google, el cual utiliza un motor avanzado de análisis de riesgo para prevenir el uso malicioso del sitio [19]. Se configuró el sitio para utilizar la versión 2, la cual lleva consigo la resolución de un desafío y la versión 3, la cual utiliza inteligencia artificial que se basa en el comportamiento del visitante en el sitio.

Google ofrece además, una consola de administración, la cual ofrece estadísticas de uso, como muestra la Figura 2.10.

reCAPTCHA trabaja con claves públicas y privadas, las públicas son visibles en el código HTML, mientras que las privadas se cargan como variable de entorno

¹⁰<https://icons.getbootstrap.com/>


¹¹<https://github.com/ambethia/recaptcha>

Iniciar sesión


Usuario

Contraseña

Recordar



I'm not a robot



reCAPTCHA
Privacy - Terms

INICIAR SESIÓN

Figura 2.9: Inicio de sesión con reCAPTCHA

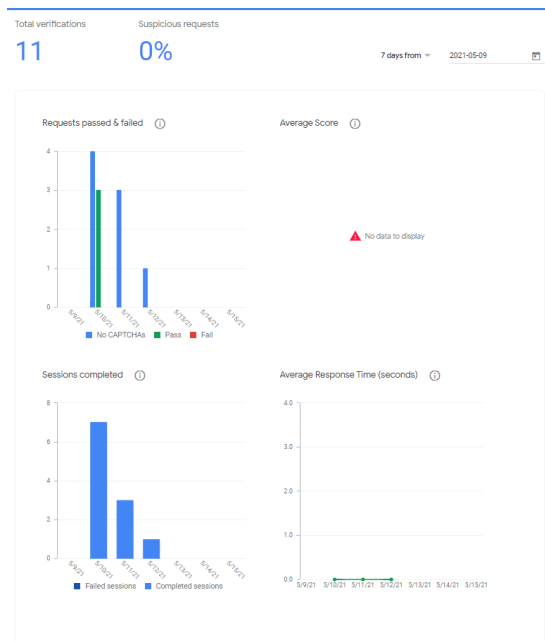


Figura 2.10: Consola de administración de Google

en el servidor. Esto las protege incluso de aquellos que tengan acceso al código fuente de la aplicación.

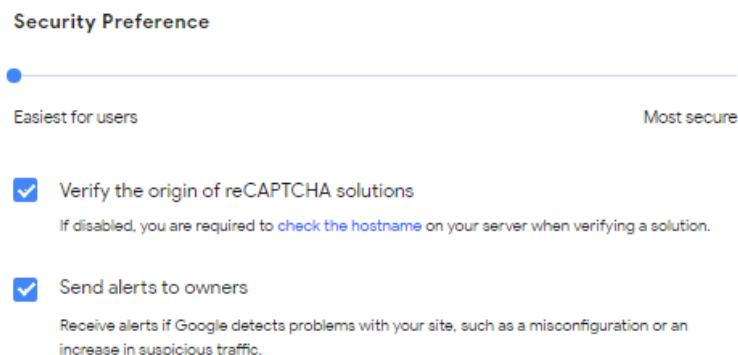


Figura 2.11: Configuración del motor v2

Para aumentar la seguridad, se configuró desde la consola de administración en el sitio de Google, permitiendo el uso de las claves solo si la petición viene desde el sitio oficial (esto evita que otros sitios suplanten la identidad o envíen su validación desde otro lugar). En esta consola se configuró también la dificultad de los desafíos de la versión 2 para ser los más simples posibles, de forma que molestaran lo menos posible a los usuarios, estas dos configuraciones se muestran en la Figura 2.11, mientras que el umbral de fallo de la versión 3 es parametrizable por código y se dejó por defecto al 50 %.

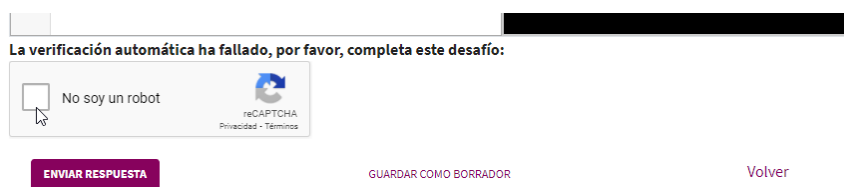



Figura 2.12: Al fallar la validación V3, se dispone de la validación v2

Se utilizó reCAPTCHA en el envío de respuestas de estudiantes, se les presenta la versión 3 (Figura 2.13) y en caso de que falle, se les muestra un aviso y se lanza la versión anterior (Figura 2.12). Se tuvo que implementar mediante JavaScript el relleno de las respuestas de campos de completar cuando fallaba la validación y se recargaba el sitio.

Eratóstenes fue un matemático, astrónomo y geógrafo griego, que desarrolló la Criba de Eratóstenes. La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número natural n dado.

Algoritmo:

Primero se forma una tabla con todos los números naturales comprendidos entre 2 y n y se van tachando los números que no son primos de la siguiente manera: Comenzando por el 2, se tachan todos sus múltiplos; comenzando de nuevo, cuando se encuentra un número que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente. El proceso termina cuando el cuadrado del siguiente número confirmado como primo es mayor que n . Se pide implementar la criba de Eratóstenes para un N arbitrario y completar el siguiente cuadro (separados con coma, sin espacios entre medio):



$n=10$	2, 3, 5, 7
$n=20$	cuadro de texto

Docente: Docente de pruebas
Institución: Matefun

Tema principal: Números primos
Dificultad: Experto
Grado: Universitario
Fecha: vie. 07 may. 2021
Con corrección automática
Ver más publicaciones del autor

Privacy - Tema

Figura 2.13: Al visitar una publicación, aparece el logo de reCAPTCHA V3 en la esquina inferior derecha

Para el inicio de sesión se utilizó solamente la versión 2 (Figura 2.9), ya que hay poca interactividad con la página y la versión 3 solía fallar la mayoría de las veces.

JavaScript

Se utilizó JavaScript y JQuery como lenguajes de programación de Frontend, por ser el primero, el estándar en desarrollo web desde hace muchos años y el segundo por los años que tiene de desarrollo y uso.

A pesar de que MateFun estuviera hecho con un Frontend en Angular, no se decidió utilizar este lenguaje ya que se desconocía por completo su funcionamiento, y debido a que implementarlo junto con Ruby On Rails precisa de una instalación y configuración inicial, que además varía con cada versión de este último, a diferencia de JavaScript, que tiene soporte por defecto y viene con la instalación de este Framework.

Webpacker

Se utilizó Webpacker el cual es una Gema que utiliza *Webpack V5* para empaquetar código JavaScript y CSS en manteniendo dos formatos, uno conveniente para los navegadores web y los desarrolladores, lo que trae un alto rendimiento con facilidades de desarrollo.

Turbolinks

Se utilizó Turbolinks el cual es una gema que intercepta todos los clics a enlaces en el sitio y reemplaza su petición HTTP por un llamado Ajax, para luego recrear solamente la partes de la pantalla que han cambiado respecto a la anterior. Esto trae consigo un mejor rendimiento, menor uso de recursos y mejores tiempos de respuesta, dando una experiencia de uso similar a las alcanzables con los Frontend más modernos [18].

Yarn

Se utilizó *Yarn* como manejador de paquetes de JavaScript. Tiene la ventaja frente a *NPM* de ser más rápido y de generar un archivo *yarn.lock*, que sirve para control de versiones.

2.11. Editor de código e interprete

Se logró reutilizar el editor de código y la consola del sitio MateFun, este consta de librerías ya establecidas de JavaScript. El código se envía al servidor para ser procesado y luego se recibe para devolvérselo al usuario.

Para hacer uso del intérprete se utilizaron las librerías de JavaScript *JQConsole*¹² y el editor de código *CodeMirror*¹³. Se eligieron estas ya que fueron las utilizadas en el proyecto de grado anterior [21] y permitían adaptarlas rápidamente al proyecto, además de dar compatibilidad. También se reutilizaron las reglas sintácticas de *CodeMirror* correspondientes al lenguaje MateFun. La Figura 2.14 muestra como luce el entorno.

¹²<https://github.com/replit-archive/jq-console>

¹³<https://codemirror.net/>

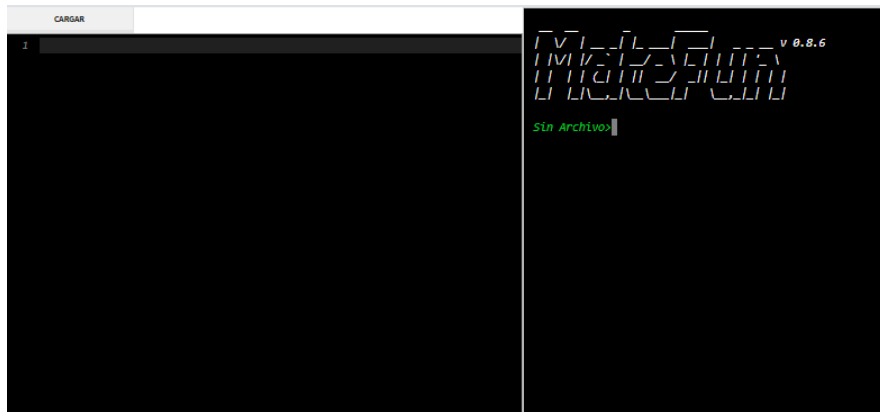


Figura 2.14: intérprete de MateFun

A pesar de que el proyecto de MateFun tenía desarrollado su Frontend en Angular, por desconocer el lenguaje y por dificultades al reutilizar el código en el proyecto, en parte porque utilizar esa tecnología sobre Ruby On Rails llevaba varias configuraciones adicionales (a diferencia de JavaScript, la cual está perfectamente incorporada en el flujo de trabajo), dependiendo de la versión no solo de Ruby On Rails sino también de Angular, se optó por convertir el código desde Typescript a JavaScript, e implementar todos los llamados, reutilizando gran parte del código, esta conversión se realizó automáticamente con conversores de código. Esto último no fue sin antes intentar sin éxito durante un par de semanas la conexión entre el Framework y Angular.

También se intentó traer el intérprete de MateFun al servidor local, para no depender de la conexión al otro sitio, pero esto requería toda la administración de hilos y la redirección de la salida desde el intérprete hacia el Frontend.

Al iniciar el entorno, de la Figura 2.14, se implementaron utilizando Ajax, funciones en JavaScript que primero se comunican con el endpoint de inicio de sesión de un servidor externo de MateFun. Para la conexión, se utiliza la cuenta de invitado (Figura 2.15 punto 1) ya que Emmy almacena los códigos fuentes, y haber utilizado una cuenta de usuario para crear y borrar códigos hubiera agregado dificultades innecesarias. Se optó por guardar el código MateFun ingresado por los usuarios completamente en la base de datos de la plataforma, junto con el contenido de las publicaciones y los ejercicios.

Una vez que se inicia sesión, se consigue el token (Figura 2.15 punto 2). Con

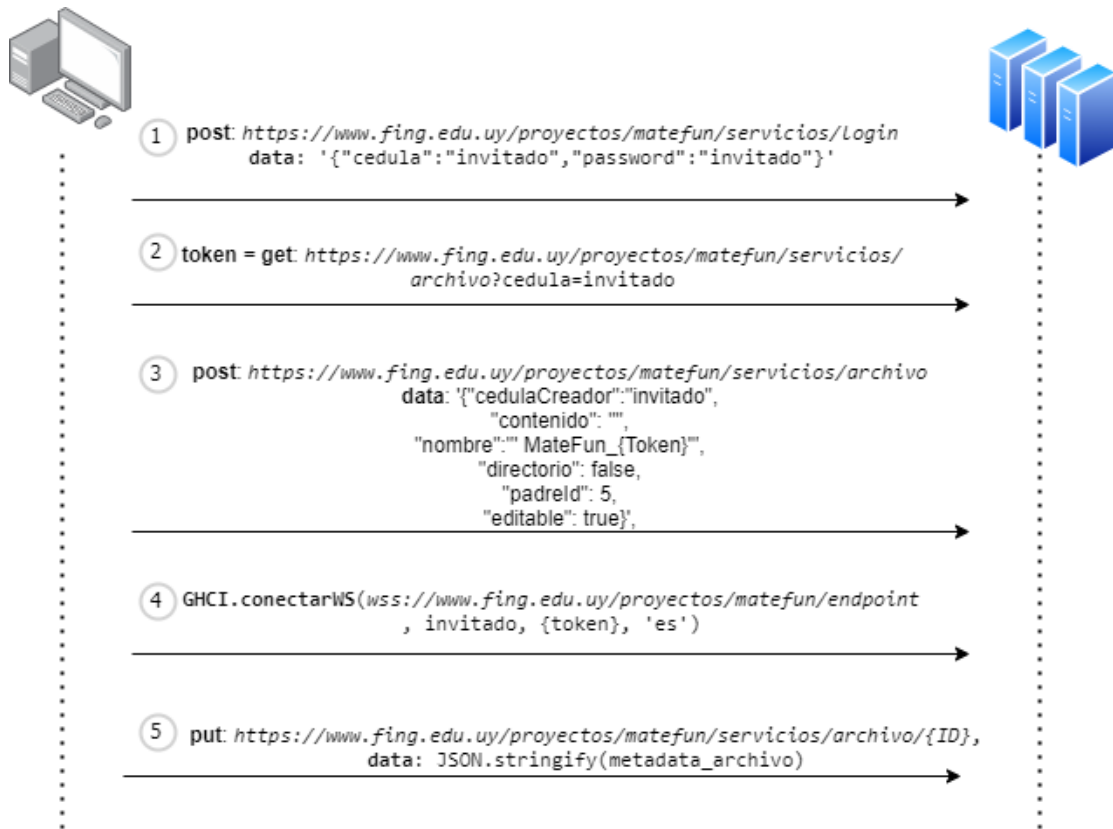


Figura 2.15: Diagrama de conexión con los servidores de MateFun

este dato se manda crear un archivo, cuyo nombre resulta de la concatenación de la cadena *Matefun_* y el token de acceso provisto por el servidor, siendo este único y administrado por MateFun (Figura 2.15 punto 3).

Luego de tener el archivo creado, se invocan los servicios *GHCI* de la implementación de MateFun (Figura 2.15 punto 4), este se encarga de la gestión de la consola y el intérprete, por lo que no se tuvo que implementar dichas funcionalidades. Para ejecutar tanto comandos MateFun como código enviado al servidor, se tiene la librería *JQConsole* con una conexión abierta WSS de MateFun, la cual permanece en este estado hasta abandonar la página con el editor.

Una vez creado el archivo, se ingresa código en el editor *Codemirror* (el cual reconoce el lenguaje gracias a la reutilización de la configuración) y al momento de presionar el botón de carga, se envía su contenido al servidor (Figura 2.15 punto 5).

Respecto a la implementación, no se lograron traer detalles como el indicador de número de línea, iconos de advertencia y error que se muestran junto al número de línea, ni sugerencias en cuanto al uso de variables, por ejemplo.

2.12. Gráficas

Se logró reutilizar el ambiente de gráficas 2D y 3D del sitio MateFun. El código se envía al servidor para ser procesado y luego se reciben los metadatos de la figura para transformarlos en imágenes 2D o 3D al usuario con librerías JavaScript.

Al igual que la implementación anterior, se intentó durante otro par de semanas incorporar el Frontend de Angular, pero por dificultades técnicas, se optó por convertir de manera automatizada todo el código escrito en Typescript a JavaScript mientras se reutilizó toda la lógica asociada.

Se implementaron los llamados a las funciones con JavaScript y JQuery, tanto para el JQuery como la consola se reutilizaron las librerías (y las configuraciones ya realizadas) del proyecto anterior de MateFun, esto incluye el Canvas de HTML5 y la librería graph3D ¹⁴, teniendo que implementar solamente las invocaciones a las funciones y la iniciación de las clases. Figura 2.16 muestra cómo luce la implementación.

¹⁴<https://github.com/ifagian/graph3D>

Se configuró el servicio que se comunica con el servidor para realizar llamados a las clases de gráficas en 2D y 3D cuando el resultado del servidor eran respuestas del tipo correspondiente.

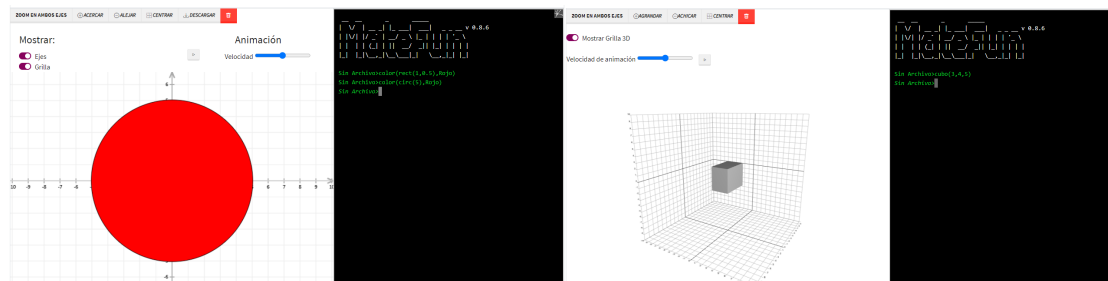


Figura 2.16: Implementación del entorno de gráficas 2D y 3D en Emmy.

No se pudo reproducir en un ambiente de desarrollo el servidor MateFun, por lo que se dependía de la instalación en producción, sin embargo, al momento de implementar el soporte de gráficas, el servidor estuvo caído y se tuvo que estudiar el código fuente y hacer una ingeniería inversa para entender cómo funcionaba y como se veían las gráficas.

Se logró finalmente un funcionamiento básico, se permite solo un gráfico a la vez y al graficar funciones en dominios discretos, las gráficas se continúan como si fueran en dominios reales. Por otro lado, si se logró el zoom y movimientos de gráficas, así como las animaciones. Por último, se logró una interfaz ligeramente diferente y más simple a la del sitio MateFun.

2.13. Editor de texto enriquecido

Como editor de texto enriquecido se consideraron *Action Text*¹⁵, nativo de Ruby on Rails, Quill¹⁶ y TinyMce¹⁷.

Se eligió TinyMce por la cantidad de funcionalidades disponibles por defecto (a diferencia de Action Text y Quill), buena documentación (que incluía varias configuraciones por defecto del editor) y uso extendido (más que *Action Text*, que está disponible por defecto solo para Ruby On Rails).

¹⁵https://edgeguides.rubyonrails.org/action_text_overview.html

¹⁶<https://quilljs.com/>

¹⁷<https://www.tiny.cloud/>

TinyMce es un editor de texto enriquecido que genera código HTML de manera transparente al usuario a partir del contenido ingresado en un editor amigable que recuerda a las interfaces antiguas de Microsoft Word y sus alternativas como OpenOffice y LibreOffice (ver Figura 2.17), lo cual se consideró como favorable para reducir la curva de aprendizaje de la plataforma, sobre todo para los docentes.

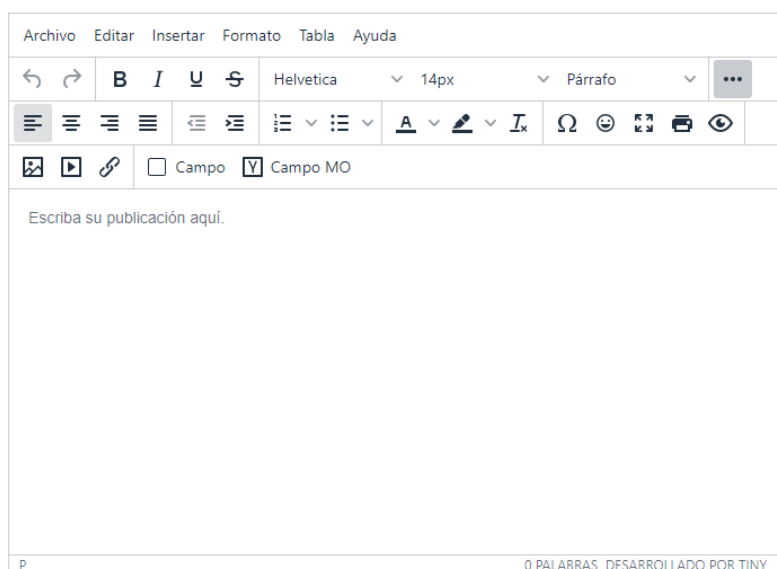


Figura 2.17: Vista del editor con los controles desarrolladores específicamente para Emmy.

Al ser un editor de texto con HTML de fondo, se pueden crear todo tipo de contenidos en completa sintonía con el sitio web, además, todos los estilos que se definieron para el sitio funcionan, para el contenido generado por los docentes. Por ejemplo se definieron los títulos del sitio (tags HTML *h*) de un color determinado, y todos los títulos que ingrese el docente en sus publicaciones van a reutilizar estos estilos, sin intervención alguna de usuarios ni desarrolladores.

Al utilizar HTML y no una sintaxis propia, se puede reemplazar este editor de texto por otro que trabaje de manera similar y todos los contenidos generados serían compatibles.

2.14. Ejercicios interactivos

El sitio cuenta con dos tipos de campos interactivos para la publicación de ejercicios. Estos son el campo de múltiple opción y el campo de completar, los cuales se gestionan enteramente del lado del cliente, sin intervención del servidor.

Para su implementación se tuvo que partir de ejemplos muy simples disponibles en la documentación y sitios de Internet, ya que no se pudo encontrar ninguna implementación previa de este tipo de campos en este editor de texto.

Generación: Punto de vista docente

Desde el punto de vista docente, este puede hacer uso de los campos interactivos al momento de escribir una publicación, Figura 2.18 muestra las dos formas de utilizarlos; yendo al menú insertar o tocando directamente su icono en la barra de tareas.

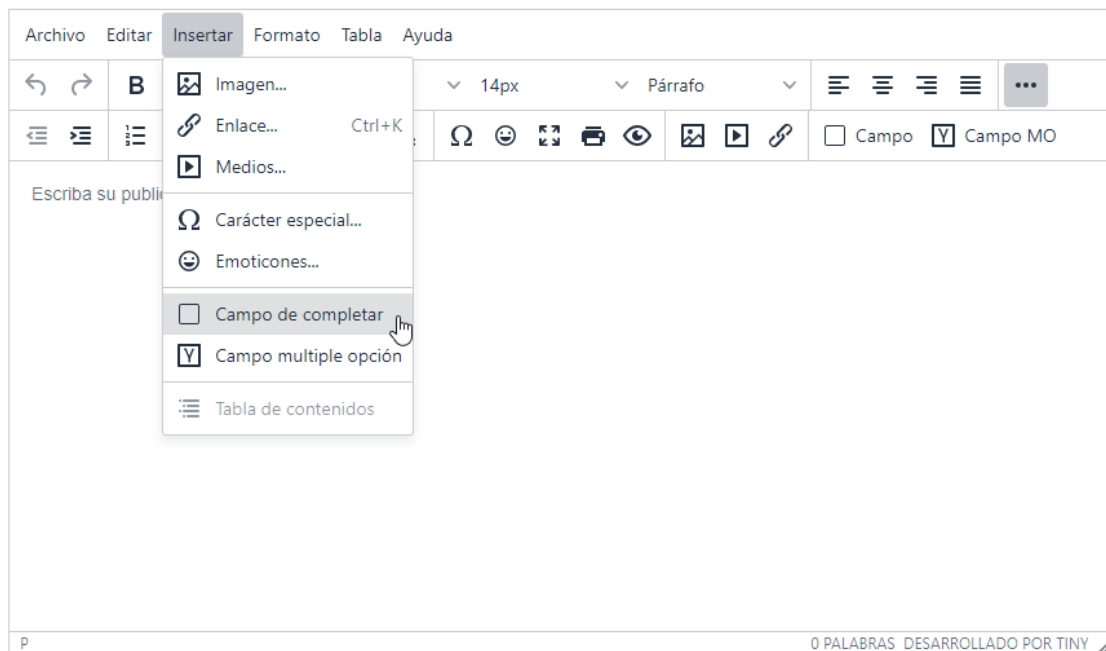


Figura 2.18: Editor de texto TinyMce con los campos de completar y múltiple opción

Estos campos fueron implementados utilizando JavaScript y siguiendo la docu-

mentación de TinyMce para controles personalizados [3] [16]. Al cargar la página, se configuró el editor para agregar los iconos en la barra de tareas y bajo el menú insertar.

Al hacerle clic a cualquiera de estos controles, se crea un cuadro de texto nativo de TinyMce, donde el docente puede escribir las respuestas. Cuando este confirma (como TinyMCE trabaja con HTML de fondo), se crea un cuadro de completar (un elemento *input* de HTML), como muestra la Figura 2.19, en la posición del cursor en el contenido del editor y se le genera un identificador alfanumérico aleatorio. Haciendo uso de las propiedades *data-* de HTML 5, se embeben las respuestas. Para evitar problemas de visualización, se quitan los tildes ingresados.

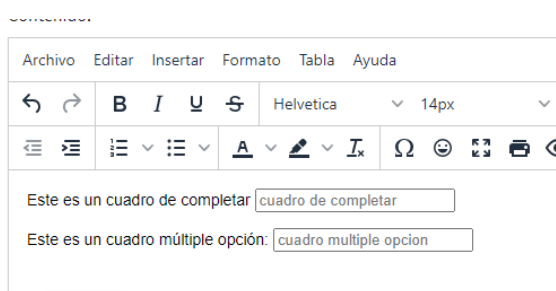


Figura 2.19: Vista de los campos de múltiple opción y de completar en el editor

En cualquier momento, el docente puede posicionar el cursor dentro del cuadro, y volviendo a elegir la opción correspondiente (campo de completar o múltiple opción) se vuelve a mostrar el cuadro de texto y se cargan las respuestas.

Es posible también transformar un cuadro de completar en un cuadro de múltiple opción y viceversa conservando las respuestas. Para esto, simplemente se posiciona el cursor en el *input* y se selecciona el campo interactivo correspondiente. El editor está programado para transformar automáticamente el cuadro de completar.

Nota: *Figura 2.19 muestra como ambos campos lucen iguales, a pesar de que el campo de múltiple opción se transformará luego en otro elemento que mostrará todas las opciones posibles al estudiante, esto se debe a una restricción de implementación, ya que el editor TinyMCE no permite seleccionar elementos HTML de manera fácil, y cuando se implementó el campo de múltiple opción como un *-select-HTML* (que es el elemento que ven los estudiantes) la única forma que tenían los docentes de seleccionarlo para cambiarles las respuestas, era arrastrando el cursor y *-pintando-* el elemento (aunque no cambiaba de color). Debido a esta limitación,*

se optó por presentar ambos campos como *-inputs-*, ya que para seleccionarlos bastaba con poner el cursor dentro del campo. Para evitar confusiones, se agregó un *-placeholder- HTML* sobre el campo con los textos *-cuadro de completa-* y *-cuadro múltiple opción-* respectivamente.

Campo de completar Un campo de completar es un campo abierto donde los estudiantes deben ingresar la respuesta, y para insertarlo alcanza con tocar la opción correspondiente y en el cuadro emergente escribir todas las respuestas correctas, una por renglón, como muestra la Figura 2.20. Luego de confirmar el ingreso, se crea un elemento *input* HTML sobre la posición del cursor.

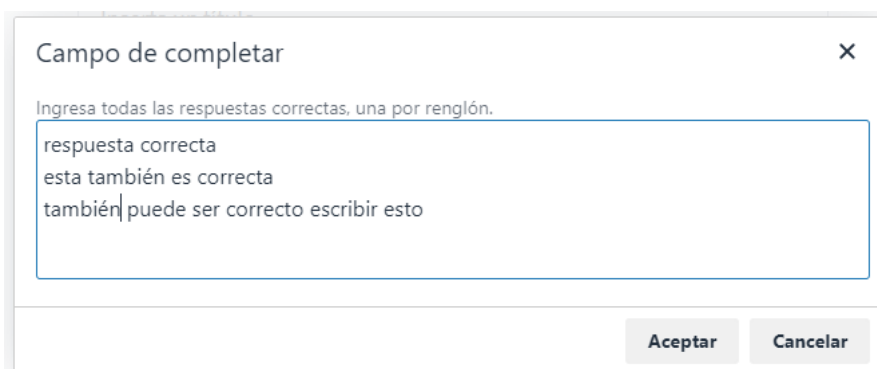


Figura 2.20: Vista del campo de diálogo para ingresar las respuestas de un campo de completar

El Listado 2.9 muestra cómo queda construido el elemento *input*, del tipo *text*. Primero, se genera un identificador aleatorio alfanumérico de diez dígitos, lo que da un total de 37^{10} identificadores distintos.

Aunque se puede editar fácilmente el código JavaScript para aumentar la cantidad de caracteres, se consideró que teniendo en cuenta que ninguna publicación iba a tener una cantidad considerable de campos de completar, y que estos identificadores son invisibles a los usuarios, el largo elegido era suficiente. Se genera también un *placeholder*, el cual ayuda al docente a diferenciar este campo del múltiple opción.

Por último, se genera, utilizando el atributo *data-* de HTML5 un *data-soluciones*, que contiene todas las respuestas correctas, separadas por un salto de línea.

```
1 <input id="matefun-campo-6mjhl48nfu"  
2   type="text"  
3   placeholder="cuadro de completar"
```

```

4 data-soluciones="respuesta correcta
5 esta tambien es correcta
6 tambien puede ser correcto escribir esto"
7 >

```

Listado 2.9: Input HTML generado a partir de la Figura 2.20

Campo de múltiple opción Un campo de múltiple opción es un campo cerrado donde los estudiantes deben elegir una respuesta (o no elegir ninguna) de todas las disponibles, para insertarlo alcanza con seleccionar la opción correspondiente y luego listar todas las respuestas, correctas e incorrectas, como muestra la Figura 2.21. Luego de confirmar el ingreso de todas las respuestas, se crea un elemento *input* HTML sobre la posición del cursor.

Para simplificar el desarrollo de la funcionalidad, se decidió que todas las respuestas correctas debían empezar con un caracter @.

The image shows a dialog box with the title "Campo de múltiple opción" and a close button (X). Inside the dialog, there is a text input field with a placeholder: "Ingresar una opción por renglón. Las respuestas correctas empiezan con el caracter '@'". Below the input field, there is a list of four options: "Primera respuesta incorrecta", "@ Respuesta correcta", "Segunda incorrecta", and "@ Esta también es correcta". At the bottom right of the dialog, there are two buttons: "Aceptar" and "Cancelar".

Figura 2.21: Vista del campo de diálogo para ingresar las respuestas de un campo de múltiple opción

El Listado 2.9 muestra cómo queda construido el elemento *input* del tipo personalizado *mo*. Primero, al igual que con el campo de completar, se genera un identificador aleatorio alfanumérico de diez dígitos. Se genera también un *placeholder*, el cual ayuda al docente a diferenciar este campo del múltiple opción.

Por último, se genera, utilizando el atributo *data-* de HTML5, un *data-soluciones*, que contiene todas las respuestas que ha ingresado el docente correspondientes, una por salto de línea, manteniendo el símbolo especial @. El editor ignora además los espacios antes y después del caracter, por lo cual el docente no debe preocuparse por si hay espacios o no entre su respuesta y el @.

```

1 <input id="matefun-campo-3bj4e7rrib"
2   type="mo"
3   placeholder="cuadro multiple opcion"
4   data-soluciones="Primera respuesta incorrecta
5   @ Respuesta correcta
6   Segunda incorrecta
7   @ Esta tambien es correcta
8 ">

```

Listado 2.10: Input HTML generado a partir del 2.21

Editar el campo Si en cualquier momento, el docente quiere editar el cuadro interactivo, simplemente debe posicionar el cursor dentro de él y seleccionar el tipo de campo en el editor de texto. Al hacer eso, el editor de texto lee las respuestas del atributo *data-soluciones* y las carga en el cuadro de diálogo. Al momento de confirmar, actualiza las respuestas.

Esto también es válido si por ejemplo se desea cambiar el tipo de campo, por ejemplo, si se selecciona un campo de completar en el contenido de la publicación y se elige del menú la opción campo de múltiple opción (o viceversa), en este caso el editor se da cuenta que se quiere cambiar el tipo por otro y hace las actualizaciones internas correspondientes, conservando todas las respuestas, todo de manera transparente para el docente.

Transición de vista docente a vista estudiante

Al enviar la publicación, Emmy guarda todo el código HTML generado, esto incluye los campos interactivos (con su solución!). Dar acceso a este código fuente HTML sin procesar a los estudiantes, sería darles la posibilidad de poder inspeccionar el código del sitio y leer las respuestas correctas directamente desde allí. Para evitar este problema, de toda publicación se guardan dos versiones, una con la vista docente y otra con la vista estudiante.

Al momento de enviar la publicación, se implementó a nivel de cliente, funciones JavaScript que analizan todo el contenido HTML ingresado por docente, capturan las respuestas de cada campo interactivo y las guardan en un *array* Json, asociando el ID con el conjunto de respuestas.

Se genera también una nueva versión de la publicación para estudiantes, donde se quita el atributo *data-soluciones* junto con su contenido, y se transforma el campo *input* del tipo múltiple opción en un elemento *selects*, quitando el *data-*

soluciones y transformando cada renglón de su contenido en un *option* (quitándoles el símbolo de @), se le agrega además una opción vacía, (que se deja disponible para cuando el estudiante no quiere responder), como muestra el Listado 2.13, se listan todas las opciones, correctas e incorrectas, de manera indistinguible.

Se genera así una versión estudiante que no contiene soluciones en su código fuente y se envían al servidor las dos versiones (docente y estudiante) junto con otro atributo *soluciones*. Cada vez que se actualiza una publicación, se generan de nuevo las soluciones y la versión estudiante.

El Listado 2.11 muestra cómo se almacenan las respuestas correctas correspondientes a los campos ingresados en Figura 2.20 y Figura 2.21. Para los campos de múltiple opción solo se capturan las respuestas correctas (se les quita además el @) y se ignoran las incorrectas, las cuales solo quedan almacenadas en la versión docente.

```
1 [
2   ["matefun-campo-6mjhl48nfu",
3     [
4       "respuesta correcta",
5       "esta tambien es correcta",
6       "tambien puede ser correcto escribir esto"
7     ]
8   ],
9   ["matefun-campo-3bj4e7rrib",
10    [
11      "respuesta correcta",
12      "esta tambien es correcta",
13      "tambien puede ser correcto escribir esto"
14    ]
15  ]
16 ]
```

Listado 2.11: Array Json de soluciones

Generación: Punto de vista estudiante

Como ya mencionado, a los estudiantes se les muestra una versión diferente que a los docentes, ya que esta no contiene información acerca de las respuestas correctas, y tiene el elemento de *input* que representa un campo de múltiple opción transformado en un elemento *select*.

Campo de texto

El campo de texto del Listado 2.9 queda, luego de la transformación, sin infor-

mación acerca de las soluciones, como muestra el Listado 2.12

```
1 <input id="matefun-campo-6mjhl48nfu"  
2   type="text"  
3   placeholder="cuadro de completar"  
4 >
```

Listado 2.12: Código HTML de un campo de completar generado para los estudiantes

Campo de múltiple opción

El campo de múltiple opción, del Listado 2.10, queda transformado en un campo *select* de HTML, donde se le ha quitado toda información acerca de las respuestas correctas, como muestra el Listado 2.13.

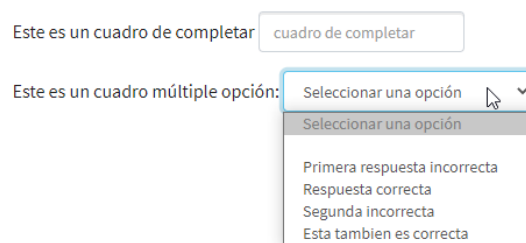


Figura 2.22: Vista de los campos interactivos por parte del estudiante

```
1 <select id="matefun-campo-3bj4e7rrib">  
2   <option value="" selected="" disabled="">Seleccionar una opcion</option>  
3   <option value=""></option>  
4   <option value="Primera respuesta incorrecta">  
5     Primera respuesta incorrecta  
6   </option>  
7   <option value=" Respuesta correcta">  
8     Respuesta correcta  
9   </option>  
10  <option value="Segunda incorrecta">  
11    Segunda incorrecta  
12  </option>  
13  <option value=" Esta tambien es correcta">  
14    Esta tambien es correcta  
15  </option>  
16 </select>
```

Listado 2.13: Código HTML generado para los estudiantes

Finalmente, Figura 2.22 muestra como lucen los campos desde el punto de vista estudiante, Notar como no se distinguen las respuestas correctas de las incorrectas.

Corrección de campos

Una vez que el estudiante completa los campos y envía la respuesta, se extraen del lado del cliente con código JavaScript las respuestas a cada uno de los campos y su identificador en un array Json en un nuevo atributo, y se envían junto con el texto de la respuesta, como muestra el Listado 2.14.

```
1 [
2   [
3     "matefun-campo-6mjhl48nfu",
4     "no se la respuesta"
5   ],
6   [
7     "matefun-campo-3bj4e7rrrib",
8     "Esta tambien es correcta"
9   ]
10 ]
```

Listado 2.14: Array Json de respuestas del estudiante.

Esto se guarda en el servidor para ser utilizado al cargar la respuesta del estudiante. Si esta fue enviada como borrador, al momento de volver a editarla, se ejecuta código JavaScript que lee el array Json de respuestas, y para cada una de ellas, utiliza selectores JQuery y el campo id almacenado para completar con la respuesta enviada.

Para el Listado 2.14, se buscará en el contenido de la publicación dos campos con identificador *matefun-campo-6mjhl48nfu* y *matefun-campo-3bj4e7rrrib*, y se completará el texto con *no sé la respuesta* para el primer campo (de completar) y se seleccionará la opción correspondiente del múltiple opción, (elemento *select*) al valor *Esta también es correcta*.

En el caso de que el docente haya borrado un campo interactivo, o eliminado la opción seleccionada por el estudiante de un campo múltiple opción, esta respuesta será ignorada y se cargarán las disponibles.

Al momento de visualizar la respuesta, si el docente tiene activada la corrección automática por parte de estudiantes, estos verán la corrección a cada campo, y tanto docentes como estudiantes verán la misma corrección, como muestra la Figura 2.24, en cambio, si está deshabilitada, se cargarán los campos pero no se

mostrará la corrección, como muestra la Figura 2.23.

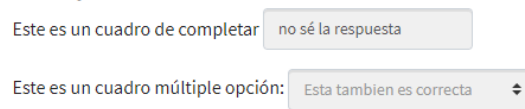


Figura 2.23: Visualización de los campos interactivos sin corregir, correspondientes a los listados 2.10 y 2.9

A diferencia del resto de las funcionalidades de campos interactivos, por seguridad y para proteger las soluciones a los ejercicios, las correcciones se realizan del lado del servidor. Como los estudiantes nunca descargan desde el servidor las soluciones, no tienen nunca acceso a las respuestas correctas, solo a la corrección de lo que enviaron.

Al momento de corregir una respuesta, se comparan los arreglos Json de cada una de las respuestas enviadas con cada una de las soluciones posibles a los campos¹⁸, los cuales se comparan a partir de coincidir de los identificadores.

Esta comparación se realiza en el modelo de respuestas, siguiendo el criterio de asignación de responsabilidades *Expert (tipo 1)* de GRASP [9]. Al momento de hacer la comparación, se ignoran todos los espacios en blanco (utilizando expresiones regulares), las mayúsculas (utilizando métodos de Ruby) y realizando la transliteración del contenido (Utilizando la gema de internacionalización de Rails [7])[17].

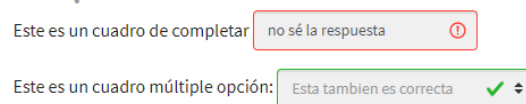


Figura 2.24: Corrección de los campos interactivos correspondientes a los listados 2.10 y 2.9

Para cada campo, se genera un Array Json con el resultado de la corrección (ver Listado 2.15) y se envía al cliente para que utilizando JavaScript cargue las respuestas y marque su corrección. Las marcas de correcto e incorrecto se insertan agregándole las clases *is-invalid* e *is-valid* de Bootstrap 4 mediante JQuery.

¹⁸Recordar que solo se almacenan respuestas correctas en el arreglo Json de soluciones.

```

1 [
2   [
3     "matefun-campo-6mjhl48nfu",
4     "no se la respuesta",
5     "nok"
6   ],
7   [
8     "matefun-campo-3bj4e7rrib",
9     " Esta tambien es correcta",
10    "ok"
11  ]
12 ]

```

Listado 2.15: Array Json de respuestas corregidas, marcando *ok* y *nok*, correspondientes a las enviadas en las figuras 2.23 y 2.24

Resumen

Tanto la generación de campos para docentes como estudiantes, como la extracción y carga de respuestas se hacen del lado del cliente, sin intervención del servidor, esto evita consumir recursos y retrasos innecesarios a causa de la comunicación entre estos, mientras que por seguridad y para proteger las soluciones, la comparación se hace del lado del servidor.

A partir del ingreso de los campos interactivos por parte del docente (Listado 2.16), siguiendo lo ingresado en Figura 2.21 y Figura 2.20, se genera del lado del cliente el código HTML mostrado en el Listado 2.16.

```

1 <p>Este es un cuadro de completar
2 <input id="matefun-campo-6mjhl48nfu" type="text" placeholder="cuadro de completar"
   ↪ data-soluciones="respuesta correcta
3 esta tambien es correcta
4 tambien puede ser correcto escribir esto">
5 </p>
6 <p>Este es un cuadro multiple opcion:
7 <input id="matefun-campo-3bj4e7rrib" type="mo" placeholder="cuadro multiple opcion
   ↪ " data-soluciones="Primera respuesta incorrecta
8 @ Respuesta correcta
9 Segunda incorrecta
10 @ Esta tambien es correcta
11 "></p>

```

Listado 2.16: Código HTML generado por parte del docente

Esto genera, al momento de enviar la publicación, un array de soluciones como muestra el Listado 2.11, y un código HTML que es mostrado al estudiante como el que muestra el Listado 2.17, donde no se le han quitado las respuestas.

```

1 <p>Este es un cuadro de completar
2 <input id="matefun-campo-6mjhl48nfu" type="text" placeholder="cuadro de completar"
   ↪ ></p>
3 <p>Este es un cuadro multiple opcion:
4 <select id="matefun-campo-3bj4e7rrib">
5   <option value="" selected="" disabled="">Seleccionar una opcion</option>
6   <option value=""></option>
7   <option value="Primera respuesta incorrecta">Primera respuesta incorrecta</
   ↪ option>
8   <option value=" Respuesta correcta"> Respuesta correcta</option>
9   <option value="Segunda incorrecta">Segunda incorrecta</option>
10  <option value=" Esta tambien es correcta"> Esta tambien es correcta</option>
11 </select>
12 </p>

```

Listado 2.17: Código HTML generado para los estudiantes

Las vistas docentes y estudiante se pueden comparar viendo las figuras 2.19 y 2.22.

Cuando el estudiante decide enviar la respuesta a una publicación, se envía junto con esta un array Json de con la respuesta de los campos interactivos, como muestra el Listado 2.14. El cual es usado también al momento de cargar una respuesta para editar.

En el caso de visualizar una respuesta a una publicación sin corrección automática, los estudiantes ven los campos sin corrección, como muestra la Figura 2.23, en caso contrario, ven lo mismo que los docentes (Figura 2.24).

Al momento de corregir, el servidor compara el arreglo Json del Listado de soluciones (Listado 2.11) con el arreglo Json de respuestas (Listado 2.14) y genera un nuevo arreglo con la corrección (Listado 2.15) el cual se carga y se muestra con código JavaScript y JQuery, dando lugar a la corrección mostrada en Figura 2.24.

2.15. Almacenamiento de imágenes

Se optó por almacenar las imágenes de las publicaciones en formato *Base64*. Esto es hecho automáticamente por TinyMce.

A nivel de base de datos, se almacena el contenido de la publicación, su código HTML correspondiente y las imágenes en una misma columna del tipo *text*. Para los docentes, el máximo largo combinado (texto, HTML e imágenes) del contenido de una publicación no puede superar el MB, mientras que para las respuestas, como no está habilitada la subida de imágenes, se dejó en 16 kB.

TinyMce también permite definir un endpoint de carga de imágenes [6], y se puede implementar utilizando la gema de ActiveSupport el almacenamiento de imágenes en disco o en un servicio de terceros (Como Amazon S3, Google Cloud Storage, o Microsoft Azure Storage) [1].

Sin embargo, se optó por la primer opción ya que se almacenaba todo el contenido de la publicación en el mismo atributo, la asociación y el borrado de imágenes al crear o eliminar una publicación no quería ninguna implementación adicional.

A los usuarios registrados se les permite subir imágenes al sitio, mientras que los visitantes pueden adjuntar imágenes a través de un enlace (debiendo estar almacenadas en otro sitio), se diseñó así para evitar un uso malintencionado del almacenamiento del sitio.

2.16. Fechas y zonas horarias

La gema de ActiveRecord de Rails almacena las fechas bajo la zona horaria UTC [2]. Para separar la presentación del almacenamiento de los datos, se mantuvo al configuración por defecto y mediante el uso de las librerías se convierten a la zona horaria del usuario.

Se utilizaron las librerías JavaScript *Jstz*¹⁹, *Moment*²⁰ y *Moment-timezone*²¹.

Para escribir las fechas en el sitio, se implementaron *partials* con la lógica de las fechas, los cuales son paramétricos, por lo cual cada vez que se quiere mostrar una fecha, alcanza con invocar el método correspondiente.

El Listado 2.18 muestra un llamado donde se pasa como parámetro la fecha de actualización de una respuesta, y el formato deseado (sino se hubiera especificado uno, el método tiene configurado un estilo por defecto).

```
1 <%=
2   render partial: 'shared/datetime/datetime',
3   locals: { datetime: @answer.updated_at, format: 'ddd DD MMM YYYY, (h:mm
4   ↪ A)' }
5 %>
```

Listado 2.18: Invocación de fechas (archivo .rb)

¹⁹<https://github.com/iainsinnott/jstz>

²⁰<https://momentjs.com/>

²¹<https://momentjs.com/timezone/>

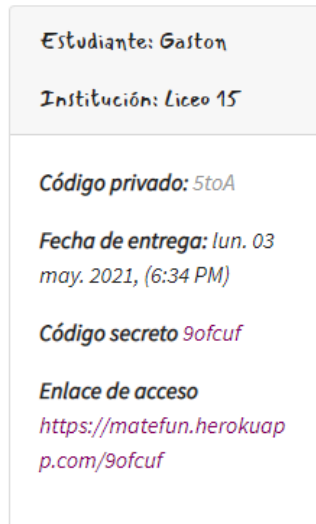


Figura 2.25: Ejemplo de fecha en Emmy

Cuando el servidor envía el código HTML, este contiene la fecha tal como es almacenado en la base de datos (Listado 2.19), y le agregan las clases *date-time* *invalid*.

```
1 <p class="date-time invalid" data-format="ddd DD MMM YYYY, (h:mm A)">  
2   2021-05-03T21:34:00+00:00  
3 </p>
```

Listado 2.19: Fecha enviada por el servidor (archivo html)

Cuando termina de cargarse la página en el navegador del cliente, con las librerías mencionadas se detectan todas las fechas mostradas en pantalla (en base a las clases) y se determina la zona horaria del visitante, reemplazando el formato por defecto del servidor (Listado 2.19) por uno establecido por el desarrollador (Listado 2.18), quitando la clase *invalid*, dando como resultado el formato mostrado en el Listado 2.20 y en la Figura 2.25, todo sin intervención del servidor.

```
1 <p class="date-time" data-format="ddd DD MMM YYYY, (h:mm A)">  
2   lun. 03 may. 2021, (6:34 PM)  
3 </p>
```

Listado 2.20: Fecha convertida en la zona horaria del cliente

Aunque no se implementó la elección de zona horaria ni su almacenamiento para cada usuario (mediante la configuración en su perfil), se configuró *Jstz* para

consultar la sesión del usuario en busca de una zona horaria, de forma que cuando se desee implementar esta funcionalidad, no haya que realizar esta configuración.

2.17. Paginación de resultados

Al presentar resultados de búsqueda y listados, estos se paginan con la gema *Will-Paginate* ²², la cual ofrece soporte para estilos Bootstrap. Esta gema está basada en los comandos *LIMIT* y *OFFSET* de SQL [20].

De esta forma, cuando se solicita información solo se consulta a la base de datos por la cantidad de registros necesarias para mostrarle al usuario en la página actual, evitando traer registros innecesarios.

2.18. Código Secreto

El sitio no está diseñado para tener estudiantes registrados, todas las respuestas a publicaciones tienen un código secreto asociado de seis dígitos, único en la plataforma, el cual sirve para recuperar y continuar (en caso de haberla guardado como borrador) con la respuesta, todas las veces que se quiera, hasta el momento de enviarla. Una vez hecho esto, la respuesta se puede seguir consultando, pero sin opción a modificar, este código se genera por única vez y permanece invariable.

El código secreto está hecho con la gema *SecureRandom*, en base 36, por lo cual está compuesta de caracteres alfanuméricos, esta librería es usada para generar claves de sesión HTTP [13], por lo cual cumple con ciertos requisitos de aleatoriedad y seguridad.

Se eligió un largo de seis ya que mantiene una buena relación entre cantidad de combinaciones y facilidad para ser manejada por los usuarios, (resultaría poco práctico por ejemplo, dar un código de 32 caracteres), además de que la memoria a corto plazo de los seres humanos puede recordar hasta siete elementos [14].

Con este largo elegido se tienen $2,176782336 \times 10^9$ combinaciones, las cuales se acabarían luego de 60 años ²³.

²²https://github.com/mislav/will_paginate

²³Si se supone 100 respuestas por día ($2176782336/100/365$) sin haber borrado ninguna en ese periodo.

La unicidad del código se controla tanto a nivel de base de datos (con la implementación de índices, que además aceleran la recuperación de la respuesta) como a nivel de modelos. Este código se almacena en la misma tabla que la respuesta, y al eliminarla, se destruye el mismo, por lo cual estos pueden ser reutilizados, mejorando la vida útil de esta implementación.

2.19. Internacionalización

Rails soporta internacionalización con su gema *I18n* [10], y su uso es bastante sencillo. Alcanza con definir un archivo YML y traducir las etiquetas correspondientes, las cuales luego se utilizan en las vistas HTML. Se han traducido los modelos y algunos de sus atributos ²⁴.

Estos se muestran en el Listado 2.21.

```
1 # config/locales/es.yml
2 es:
3   activerecord:
4     models:
5       topic: "Tema"
6       publication: "publicacion"
7       answer: "Respuesta"
8       user: "Usuario"
9       difficulty: "Dificultad"
10      grade: "Grado"
11     attributes:
12       publication:
13         content: "Contenido de la publicacion"
14         title: "Titulo de la publicacion"
15         short_description: "descripcion corta"
16       topic:
17         name: "Nombre"
18         description: "descripcion"
19       answer:
20         rich_text: "Contenido de la respuesta"
21         visitor_name: "Nombre"
22         visitor_institution: "institucion"
23       user:
24         email: "Email"
25         username: "Nombre de usuario"
26         name: "Nombre"
27         institution: "institucion"
28   recaptcha:
29     errors:
```

²⁴Por convención los nombres de los modelos y atributos van en inglés.

```

30     verification_failed: "Ha fallado la verificacion de reCAPTCHA, por favor
↪ intente de nuevo."
31     recaptcha_unreachable: "Oops, fallamos en validar tu reCAPTCHA, por favor
↪ intenta de nuevo."

```

Listado 2.21: Traducciones definidas

2.20. Breadcrumbs

Se utilizó la gema `Gretel`²⁵ para definir los breadcrumbs del sitio, estos son del tipo *localización*, el cual además se comporta como uno dinámico [8] en la mayoría de los casos.

Su configuración consiste en definir un archivo de configuración con todos los caminos y luego invocarlos desde la vista. Para determinar el camino, el sitio se basa en la URL anidada, los listados 2.22, 2.23, 2.24 muestran cómo dependiendo del camino que haya realizado el usuario para llegar a la publicación (los cuales generan URL distintas) es el contenido de la miga de pan que se muestra en el sitio.

```

1 https://matefun.herokuapp.com/users/7/publications/47
2
3 Emmy > Acceso Estudiantes > Docentes > Docente de pruebas >
4 Introduccion a la grafica

```

Listado 2.22: Acceso desde el Listado de las publicaciones del docente

```

1 https://emmy-fing.herokuapp.com/publications/47
2
3 Emmy > Acceso Estudiantes > Publicaciones > Introduccion a la grafica

```

Listado 2.23: Acceso a la publicación mediante enlace directo o el buscador

```

1 https://emmy-fing.herokuapp.com/topics/107/publications/47
2
3 Emmy > Acceso Estudiantes > Temas > Funciones > Introduccion a la grafica

```

Listado 2.24: Acceso a la publicación a través del Listado de temas

Al haber definido los recursos como anidados (Sección 2.9) se generan automáticamente direcciones anidadas, las cuales son accedidas a través de la navegación en el sitio, esto favoreció y simplificó la implementación de los breadcrumbs.

²⁵<https://github.com/kzkn/gretel>

El Listado 2.25 muestra un fragmento de la definición del archivo de miga de pan. El texto a mostrar se puede definir como un String (como en el caso de *Publicaciones*) el cual también se puede sustituir por etiquetas (lo cual sería necesario si se quiere un sitio multilinguaje) o también se puede parametrizar y mostrar un atributo (Como en el caso de mostrar el título de la publicación, *publication.title*).

```

1 ...
2 crumb :publications do
3   parent :student
4   link 'Publicaciones', publications_path
5 end
6 crumb :publication do |publication|
7   parent :publications
8   link publication.title, publication_path(publication)
9 end
10 crumb :topic_publication do |publication|
11   parent publication.topic
12   link publication.title, publication_path(publication)
13 end
14 ...
15 crumb :user_publication do |publication|
16   parent publication.user
17   link publication.title, publication_path(publication)
18 end
19 ...

```

Listado 2.25: Fragmento de definición de breadcrumbs

2.21. Listas desplegables autocompletables

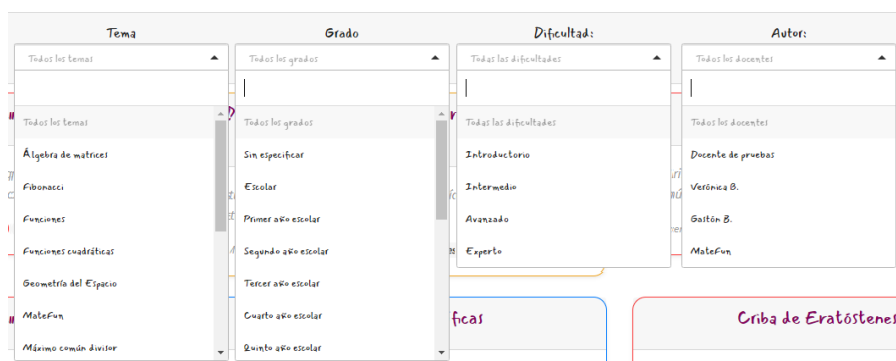


Figura 2.26: Filtros de publicaciones activados

Para los filtros, se definieron listas desplegables autocompletables, las cuales ayudan a seleccionar la opción deseada.

Esta implementación se llevó a cabo con la librería JavaScript *Choices.js*²⁶. Esta librería trae consigo una definición de estilos, las cuales se mantienen en un archivo separado a Bootstrap y al archivo con los estilos de Emmy, de manera que si se desea cambiar la librería, alcanza con borrar el archivo *.css* asociado.

Figura 2.26 muestra todos los filtros posibles del buscador de aplicaciones, y las listas desplegadas con las opciones de cada uno de ellos.

Figura 2.27 muestra como al escribir una subcadena, las opciones se filtran, facilitando la elección. Esto es especialmente útil con la selección de temas y docentes, ya que las dificultades y años escolares son finitos y conocidos. Hay que aclarar además, que en el filtrado por tema, solo se muestran aquellos que tienen publicaciones en estado publicado, eso evita que el visitante pierda tiempo filtrando por temas que no van a dar resultados.

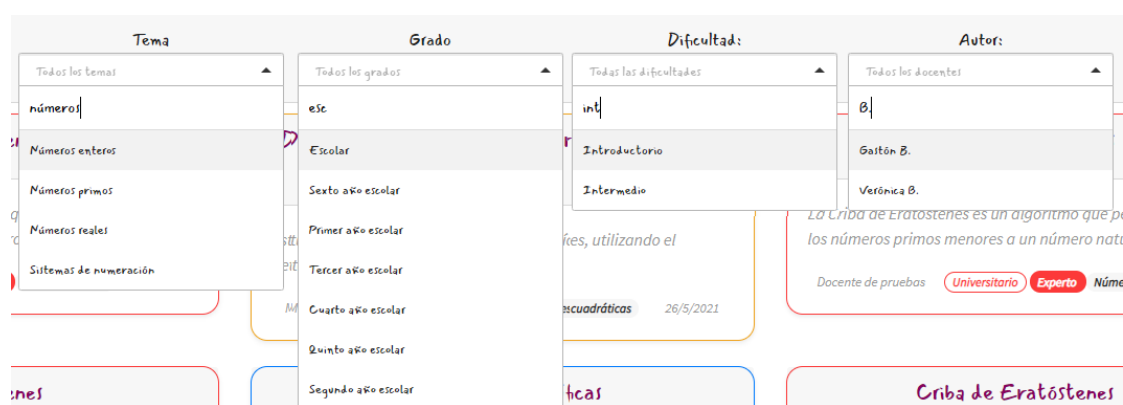


Figura 2.27: Filtros de publicaciones activados

La utilización de la librería es bastante sencilla, Listado 2.26 muestra la invocación de la librería para el listado de temas, el cual consiste en crear un objeto de tipo *choices*. Manteniendo la separación entre HTML y JavaScript, en el Listado 2.27 se muestra el listado de temas, para el cual alcanza con asignarle el mismo ID que se referencia en el código JavaScript. Si en un futuro se desea reemplazar la librería o quitarla, no se necesita modificar el código HTML.

²⁶<https://github.com/Choices-js/Choices>

```

1 var dropDownSelects = new Choices("#dropdown-choice-select-topic", {
2   searchFields: ["label"],
3   noResultsText: "No se encontraron resultados",
4   noChoicesText: "No hay opciones para mostrar",
5 });

```

Listado 2.26: Invocación de la librería

```

1 <%= f.collection_select :topic_id,
2   Topic.with_published_publications,
3   :id,
4   :name,
5   {
6     include_blank: 'Todos los temas' ,
7     selected: selected_options["topic_id"]
8   },
9   {
10    id: 'dropdown-choice-select-topic',
11    class: "form-filter",
12    onchange: "this.form.submit()"
13  }
14 %>

```

Listado 2.27: Listado de temas

Glosario

API Interfaz de programación de aplicaciones. 3

Backend Parte del software con la cual procesa la entrada desde el frontend y no interactúa con los usuarios. 3, 5, 6, 36

Base 64 En codificación, es un grupo de esquemas de codificación de binario a texto ASCII, utilizando un esquema de numeración de base 64.. 35

Breadcrumb Técnica de navegación utilizada en la interfaz de usuario para indicar su ruta de navegación. 65

Commit Confirmar un conjunto de cambios temporales de forma permanente. 4

Concern Extensiones de módulo a más alto nivel. 24, 25

CRUD Crear, leer, actualizar y borrar. 22, 23, 25

CSS Hoja de estilos en cascada. 40, 44

código privado Identificador acordado entre el docente y sus estudiantes, su utilidad es facilitar la identificación de soluciones. Un estudiante al enviar su solución puede ingresar opcionalmente un código, el docente al acceder al listado de soluciones puede filtrarlas, obteniendo así las soluciones de un grupo en particular.. 17, 19

código secreto Identificador único en la plataforma, es proporcionado por esta al estudiante cuando responde una publicación. Con este código el estudiante puede volver a cargar su borrador para continuar el trabajo o visualizar la respuesta una vez enviada.. 18, 19, 21, 23, 24, 39, 63

DRY Don't repeat yourself (No repitas). 8

Fat model, Skinny Controller Se basa en mantener la lógica en los modelos y limitar a los controladores a realizar los llamados. 21

Framework Entorno de trabajo que agrega al lenguaje base funcionalidades con el fin de reutilizar código, para simplificar y acelerar el desarrollo. 4–6, 8, 43, 45

Frontend Parte del software con la cual interactúan los usuarios. 3, 5, 6, 26, 27, 36, 37, 40, 43–45, 47

gema Programas y librerías del lenguaje Ruby. 8, 13, 14, 17, 18, 24, 25, 27, 30, 32, 34–36, 40, 44, 58, 61, 63–65

HTML Lenguaje de marcado de hipertexto. 15–17, 27, 30, 32–34, 37, 40, 49, 51–54, 59, 60, 62, 64, 67

Lexema Unidad de significado léxico que se relaciona con un conjunto de palabras a través de la flexión lingüística. En bases de datos son palabras que han sido normalizadas para unificar las variantes de una misma palabra. 31

Merge-commit Operación en el código fuente que une dos líneas de código. 4

Merge-request Peticiones para unir una línea de código a otra. 3, 4, 27

modelo Un modelo es una clase de Ruby que representa una tabla en la base de datos. 15, 21–25, 27, 33, 58

MVC Modelo, vista, controlador. 5, 8, 19

MVP Mínimo producto viable. 6, 26

NIST Instituto Nacional de Estándares y Tecnología. 36

refactorización Reestructurar código existente sin alterar su funcionalidad, para reducir su extensión, complejidad, aumentar tu mantenimiento y su legibilidad. 8

sanitización Limpieza y filtrado de elementos para evitar un uso indeseado o malicioso. 34

STI Herencia en única tabla. 13

ts vector En PostgreSQL, es una lista ordenada de lexemas. . 6, 8, 31

visitantes Personas sin sesión iniciada. 17, 19, 22, 25, 26, 61

vista Representación final de una petición. 8, 21, 25, 33, 60, 64, 65

WSS Prefijo para indicar un protocolo de websocket seguro. 47

YML Formato de serialización de datos. 64

Bibliografía

- [1] Active storage overview. https://edgeguides.rubyonrails.org/active_storage_overview.html. Accedido: 05-2021.
- [2] ActiveSupport::timezone. <https://api.rubyonrails.org/classes/ActiveSupport/TimeZone.html>. Accedido: 05-2021.
- [3] Create a custom toolbar button. <https://www.tiny.cloud/docs-4x/advanced/creating-a-custom-button/>. Accedido: 05-2021.
- [4] Digital identity guidelines. <https://pages.nist.gov/800-63-3/sp800-63b.html>. Accedido: 05-2021.
- [5] Flujo de trabajo de gitflow. <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow#:~:text=El%20flujo%20de%20trabajo%20Gitflow,pr%C3%A1cticas%20de%20implementaci%C3%B3n%20de%20DevOps.&text=El%20flujo%20de%20trabajo%20Gitflow%20define%20un%20modelo%20de%20creaci%C3%B3n,publicaci%C3%B3n%20del%20proyecto%20como%20fundamento>. Accedido: 07-2021.
- [6] Image and file options. <https://www.tiny.cloud/docs/configure/file-image-upload/>. Accedido: 05-2021.
- [7] Method: I18n.transliterate. <https://www.rubydoc.info/gems/activesupport/2.3.17/I18n.transliterate>. Accedido: 05-2021.
- [8] Miga de pan (informática). [https://es.wikipedia.org/wiki/Miga_de_pan_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Miga_de_pan_(inform%C3%A1tica)). Accedido: 05-2021.

- [9] Programación avanzada, criterio de asignación de responsabilidades. <https://www.fing.edu.uy/tecnoinf/mvd/cursos/progavan/material/teo/pavan-teorico12-disenio-grasp.pdf>. Accedido: 05-2021.
- [10] Rails internationalization (i18n) api. <https://guides.rubyonrails.org/i18n.html>. Accedido: 05-2021.
- [11] Ruby on rails. https://en.wikipedia.org/wiki/Ruby_on_Rails. Accedido: 05-2021.
- [12] The ruby style guide. <https://rubystyle.guide/>. Accedido: 05-2021.
- [13] Securerandom. <https://ruby-doc.org/stdlib-2.5.1/libdoc/securerandom/rdoc/SecureRandom.html>. Accedido: 05-2021.
- [14] Short-term memory: Up to 7 items, but highly volatile. https://thebrain.mcgill.ca/flash/capsules/experience_jaune03.html#:~:text=Short%2DTerm%20Memory%3A%20Up%20to%207%20Items%2C%20But%20Highly%20Volatile&text=Many%20psychology%20experiments%20have%20shown,2%2C%20depending%20on%20the%20individual. Accedido: 05-2021.
- [15] Tables and indexes. <https://www.postgresql.org/docs/9.5/textsearch-tables.html>. Accedido: 05-2021.
- [16] Toolbar buttons. <https://www.tiny.cloud/docs/ui-components/toolbarbuttons/>. Accedido: 05-2021.
- [17] Transliteración. <https://es.wikipedia.org/wiki/Transliteraci%C3%B3n>. Accedido: 05-2021.
- [18] Turbolinks - github. <https://github.com/turbolinks/turbolinks>. Accedido: 05-2021.
- [19] What is recaptcha. <https://www.google.com/recaptcha/about/>. Accedido: 05-2021.
- [20] will paginate documentation. https://github.com/mislav/will_paginate/wiki. Accedido: 05-2021.

- [21] Gonzalo Cameto et al. Documento de arquitectura y diseño.
- [22] Full-text search in postgresql in milliseconds.
- [23] Niels Provos and David Mazier. A futureadaptable password schem.