

HERRAMIENTA WEB PARA LA ENSEÑANZA DE LA PROGRAMACIÓN EN NIÑOS Y JÓVENES: "MI SUPERPODER ES LA PROGRAMACIÓN"

ERIKA JISSEL GUTIÉRREZ BELTRÁN

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface, en alcances y calidad, todos los requisitos que demanda un Trabajo de Grado de Maestría.



JUAN CARLOS MARTÍNEZ ARIAS  
Director Trabajo de Grado



LUISA FERNANDA RINCÓN PÉREZ  
Jurado



LILIANA ALARCÓN ALZATE  
Jurado

Aprobado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali, para optar el título de Magister en INGENIERIA DE SOFTWARE.



HERNÁN CAMILO ROCHA NIÑO Ph. D.  
Decano Facultad de Ingeniería y Ciencias



JUAN CARLOS MARTÍNEZ ARIAS  
Director Posgrados de Ingeniería y Ciencias



**Acta de Correcciones al Documento de Trabajo de Grado**

**Santiago de Cali, 3 de marzo de 2023**

**Autor: Erika Jissel Gutiérrez Beltrán**

**Título del Trabajo de Grado: HERRAMIENTA WEB PARA LA ENSEÑANZA DE LA PROGRAMACIÓN EN NIÑOS Y JÓVENES “MI SUPERPODER ES LA PROGRAMACIÓN”**

**Director:**

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuarán, como consta en el Acta de Evaluación correspondiente.

---

Juan Carlos Martínez Arias  
Director del Trabajo de Grado



Pontificia Universidad  
**JAVERIANA**  
Cali

**Maestría en Ingeniería de software**  
**Facultad de Ingeniería y Ciencias**

### **Datos Actuales**

**Nombre completo:** Erika Jissel Gutiérrez Beltrán

**Dirección:** Calle 51 #83-96, Casa 155, unidad residencial Olivenza

**Teléfono de contacto:** 3187381607

**Correo electrónico:** erika.giselle.gb@hotmail.com

**Profesión:** Ingeniera multimedia

**Universidad:** Universidad de San Buenaventura - Cali

**Empresa:** Mercado Libre

**Cargo:** Ingeniera de software



Pontificia Universidad  
**JAVERIANA**  
Cali

**Maestría en Ingeniería de software**  
**Facultad de Ingeniería y Ciencias**

## **ABSTRACT**

Programming, as a skill, has been considered to be the central pillar in developing a better understanding and affinity for technology, assisting to develop additional capabilities, such as logical and critical thinking in problem solving, creativity, as well as providing all who practice it with systematic reasoning. Due to the advantages found in programming, many internationally recognized institutions and universities work daily to teach programming to children and youth of all ages. Thanks to this, it is possible to find many platforms to teach coding and informatics. However, in Colombia, many children and young adults cannot access these resources due to the costly licenses associated with some tools as well as a lack of access to the necessary devices to connect to the educational platforms available online. This results in an inability for Colombian children and young adults to improve their abilities and strengthen their education. Additionally, many platforms dedicated to teaching programming are focused on teaching the code rather than logic that is foundational to developing strong computational thinking.

This project presents the prototype of a tool for teaching programming to children and young adults, based on the initial principles of computing, it offers a new, holistic methodology to teach programming for free to children and young adults in vulnerable sectors, thus giving them alternatives for their education.



## **RESUMEN**

La enseñanza de la programación cada día toma más fuerza debido a la revolución tecnológica que se está viviendo en la actualidad. El uso de dispositivos tecnológicos ha mejorado la calidad de vida de las personas, facilitando la comunicación y el acceso a la información. Sin embargo, muchos siguen desaprovechando las posibilidades que esta les ofrece, ya que son mayormente consumidores en vez de creadores. La programación ha sido considerada como uno de los pilares fundamentales para tener una mejor integración y afinidad con la tecnología. Además, otorga habilidades adicionales como lo es el desarrollo del pensamiento lógico y crítico para la solución de problemas, creatividad y razonamiento sistemático a todos los que la practican. A partir de las ventajas encontradas en la programación, diversas instituciones y universidades reconocidas en el mundo, trabajan día a día para enseñar programación a los niños y jóvenes. Gracias a esto es posible encontrar múltiples plataformas para la enseñanza del código y la informática. En Colombia existen muchos niños y jóvenes con pocas posibilidades de acceder a este tipo de recursos que mejoran sus habilidades y fortalecen su educación, ya sea por los costos que las herramientas puedan representar, o por no contar con los dispositivos necesarios para conectarse a las páginas web disponibles en internet. Adicionalmente, gran parte de las plataformas centradas en la enseñanza de la programación, se enfocan en el código y no en las habilidades lógicas necesarias para desarrollar el pensamiento computacional. Por medio del desarrollo de este proyecto, se crea el prototipo de una plataforma para la enseñanza de la programación en niños y jóvenes teniendo como base los principios iniciales de la computación, donde más allá de la plataforma, sea posible convertir todo el proceso que ofrecerá la herramienta en una metodología para la enseñanza, siendo de acceso libre, con el fin de proporcionar a los niños y jóvenes de sectores vulnerables más alternativas para su formación.



FICHA RESUMEN  
TRABAJO DE GRADO DE MAESTRÍA

**TITULO:** HERRAMIENTA WEB PARA LA ENSEÑANZA DE LA PROGRAMACIÓN EN NIÑOS Y JÓVENES “MI SUPERPODER ES LA PROGRAMACIÓN”

**1. ÁREA DE INVESTIGACIÓN:** Aplicado

**2. ESTUDIANTE:** Erika Jissel Gutiérrez Beltrán

**3. CORREO ELECTRÓNICO:** erika.giselle.gb@hotmail.com

**4. DIRECTOR:** Juan Carlos Martínez Arias

**5. PALABRAS CLAVE:** Ingeniería de software, software educacional, software en la nube, tecnologías web, programación para niños

**6. FECHA DE INICIO:** 01 de Febrero de 2021

**7. FECHA ENTREGA:** 13 de Enero de 2023

**8. RESUMEN:** La enseñanza de la programación cada día toma más fuerza debido a la revolución tecnológica que se está viviendo en la actualidad. El uso de dispositivos tecnológicos ha mejorado la calidad de vida de las personas, facilitando la comunicación y el acceso a la información. Sin embargo, muchos siguen desaprovechando las posibilidades que esta les ofrece, ya que son mayormente consumidores en vez de creadores. La programación ha sido considerada como uno de los pilares fundamentales para tener una mejor integración y afinidad con la tecnología. Además, otorga habilidades adicionales como lo es el desarrollo del pensamiento lógico y crítico para la solución de problemas, creatividad y razonamiento sistemático a todos los que la practican. A partir de las ventajas encontradas en la programación, diversas instituciones y universidades reconocidas en el mundo, trabajan día a día para enseñar programación a los niños y jóvenes. Gracias a esto es posible encontrar múltiples plataformas para la enseñanza del código y la informática. En Colombia existen muchos niños y jóvenes con pocas posibilidades de acceder a este tipo de recursos que mejoran sus habilidades y fortalecen su educación, ya sea por los costos que las herramientas puedan representar, o por no contar con los dispositivos necesarios para conectarse a las páginas web disponibles en internet. Adicionalmente, gran parte de las plataformas centradas en la enseñanza de la programación, se enfocan en el código y no en las habilidades lógicas necesarias para desarrollar el pensamiento computacional. Por medio del desarrollo de este proyecto, se crea el prototipo de una plataforma para la enseñanza de la programación en niños y jóvenes teniendo como base los principios iniciales de la computación, donde más allá de la plataforma, sea posible convertir todo el proceso que ofrecerá la herramienta en una metodología para la enseñanza, siendo de acceso libre, con el fin de proporcionar a los niños y jóvenes de sectores vulnerables más alternativas para su formación.



Pontificia Universidad  
**JAVERIANA**  
Cali

**HERRAMIENTA WEB PARA LA ENSEÑANZA DE LA  
PROGRAMACIÓN EN NIÑOS Y JÓVENES  
“MI SUPERPODER ES LA PROGRAMACIÓN”**

*Erika Jissel Gutiérrez Beltrán*  
Código 1144181235

*Trabajo de grado para optar al título de  
Magister en Ingeniería de Software*

Director  
Juan Carlos Martínez Arias

FACULTAD DE INGENIERÍA Y CIENCIAS  
MAESTRÍA EN INGENIERÍA DE SOFTWARE  
SANTIAGO DE CALI, *ENERO DE 2023*

## AGRADECIMIENTOS

*Gracias a cada una de las personas que me acompañaron en este proceso, que confiaron en mí y me brindaron su ayuda y apoyo.*

*Quiero agradecer en especial a mi profesor Juan Carlos Martínez por ser mi tutor, por acompañarme en este proceso, por sus sugerencias, valiosos aportes, apoyarme para hacer posible y alcanzar esta meta.*

*Gracias a los profesores por sus valiosas enseñanzas. Gracias a los padres y niños que hicieron parte del proceso de pruebas, que con su motivación me inspiran a seguir con este proyecto para transformar vidas a través de la enseñanza.*

*Gracias a cada uno de los profesores, estudiantes y amigos que probaron la herramienta y que con sus aportes fue posible mejorar el prototipo final.*

## TABLA DE CONTENIDO

|   |           |
|---|-----------|
| <b>INTRODUCCIÓN</b>   | <b>10</b> |
| <b>1. DEFINICIÓN DEL PROBLEMA</b>   | <b>12</b> |
| 1.1 PLANTEAMIENTO DEL PROBLEMA  | 12        |
| 1.2 FORMULACIÓN DEL PROBLEMA  | 14        |
| <b>2. OBJETIVOS DEL PROYECTO</b>  | <b>15</b> |
| 2.1 OBJETIVO GENERAL  | 15        |
| 2.2 OBJETIVOS ESPECÍFICOS   | 15        |
| <b>3. ALCANCE</b>   | <b>16</b> |
| <b>4. MARCO TEÓRICO</b>   | <b>17</b> |
| 4.1 Identificación del estado actual de las herramientas existentes para la enseñanza de la programación en niños y jóvenes | 18        |
| 4.2 Trabajos de investigación relacionados a la enseñanza de la programación  | 26        |
| 4.3 Impacto de la enseñanza de la computación y programación en niños y jóvenes   | 27        |
| 4.5 Usabilidad de las herramientas actuales para la enseñanza de la computación   | 29        |
| <b>5. OBTENCIÓN Y ANÁLISIS DE REQUISITOS</b>  | <b>30</b> |
| 5.1 Captura de información previa para la construcción de los requisitos  | 30        |
| 5.2 Definición de requisitos  | 33        |
| 5.2.1 Requisitos funcionales  | 37        |
| 5.2.2 Requisitos no funcionales   | 41        |
| 5.2.3 Restricciones de diseño   | 43        |
| 5.3 Validación de requisitos  | 43        |
| 5.4 Construcción de sprints para el desarrollo del proyecto   | 45        |
| <b>6. DISEÑO</b>  | <b>46</b> |
| 6.1 Análisis de patrones de arquitectura de software  | 46        |
| 6.1.1 Patrón de arquitectura en capas   | 46        |
| 6.1.2 Patrón de arquitectura microservicios   | 48        |
| 6.1.3 Estilo de arquitectura modelo-vista-controlador   | 49        |
| 6.1.4 Estilo de arquitectura orientado a servicios (SOA)  | 50        |
| 6.1.5 Patrones de arquitectura sin servidor (Serverless)  | 50        |
| 6.2 Patrones de arquitectura para videojuegos   | 54        |
| 6.3 Definición de la arquitectura a utilizar en el desarrollo de la herramienta   | 55        |
| 6.4 Diseño de base de datos   | 64        |
| 6.5 Definición de la tecnología a utilizar  | 69        |
| 6.6 Bocetos, escenarios e interfaces  | 70        |
| <b>7. DESARROLLO</b>  | <b>75</b> |
| 7.1 Construcción del prototipo de una herramienta web para la enseñanza de la programación                                  | 75        |

|  |            |
|--|------------|
| <b>8. PRUEBAS</b>  | <b>77</b>  |
| <b>9. CONCLUSIONES Y TRABAJOS FUTUROS</b>                      | <b>90</b>  |
| 9.1 Conclusiones   | 90         |
| 9.2 Trabajos Futuros   | 93         |
| <b>REFERENCIAS</b>   | <b>95</b>  |
| <b>ANEXOS</b>  | <b>100</b> |
| 1. Documento de especificación de requisitos de la herramienta | 100        |
| 2. Game Design Document - Documento de diseño de juego         | 131        |
| 3. Cuestionario de pruebas de la herramienta                   | 154        |
| 4. Políticas de privacidad                                     | 162        |
| 5. Interfaces de alta fidelidad, diseño UX                     | 166        |

## LISTADO DE TABLAS

|   | Pág |
|---|-----|
| <b>Tabla 1:</b> Extracción de características de las herramientas más utilizadas, para la construcción y definición de requisitos | 25  |
| <b>Tabla 2:</b> Descripción de las herramientas para aprender a programar más utilizadas  | 31  |
| <b>Tabla 3:</b> Definición de requisitos funcionales del rol del profesor y estudiante  | 38  |
| <b>Tabla 4:</b> Definición de requisitos no funcionales de la herramienta web para la enseñanza de la programación                | 41  |
| <b>Tabla 5:</b> Comparación de las arquitecturas, análisis de ventajas y desventajas  | 55  |
| <b>Tabla 6:</b> Extracción de características del prototipo de la herramienta Mi Superpoderes la Programación                     | 92  |

## LISTADO DE IMÁGENES

|  | Pág |
|--|-----|
| <b>Figura 1:</b> Imagen del entorno de desarrollo web de scratch y ejemplo de programación en bloques  | 19  |
| <b>Figura 2:</b> Ambiente de programación para pilas bloques y muestra del primer ejercicio  | 20  |
| <b>Figura 3:</b> Entorno de programación en Codemonkey y ejemplo de realización de cada uno de los niveles usando código   | 21  |
| <b>Figura 4:</b> Ambiente de programación en Code.org, hora del código Angry Birds   | 22  |
| <b>Figura 5:</b> Ambiente de programación en Tynker y proyecto de ejemplo en la versión gratuita   | 23  |
| <b>Figura 6:</b> Videojuego Rabbids Coding en uno de los niveles, el cual es solucionado por bloques   | 24  |
| <b>Figura 7:</b> Diagrama de caso de uso desde el rol del profesor. Se muestran las funcionalidades con las que contará el usuario   | 34  |
| <b>Figura 8:</b> Diagrama de caso de uso desde el rol del estudiante. Se muestran las funcionalidades con las que contará el usuario   | 35  |
| <b>Figura 9:</b> Diagrama de actividad sobre el desarrollo de los juegos o actividades de la herramienta. Se muestra la interacción en determinados casos y respuesta a las acciones ejecutadas                | 36  |
| <b>Figura 10:</b> Diagrama de actividad sobre el módulo de ayuda con el que contará la herramienta   | 36  |
| <b>Figura 11:</b> Diagrama de actividad sobre el módulo de personalización del avatar. Esta sección se utilizará como incentivo para el usuario o estrategia de gamificación                                   | 37  |
| <b>Figura 12:</b> Interfaz de usuario tomada desde el rol del profesor, se visualiza la creación de estudiantes para una clase   | 44  |
| <b>Figura 13:</b> Interfaz de usuario tomada desde el rol del estudiante, se visualizan las clases disponibles o asignadas a un estudiante y el mapa de progreso de una clase en particular y espacio de juego | 44  |
| <b>Figura 14:</b> Tablero de trello creado para la gestión del desarrollo de los requisitos del proyecto   | 45  |
| <b>Figura 15:</b> Tablero de trello creado para la gestión de la documentación y seguimiento general del proyecto  | 45  |
| <b>Figura 16:</b> Representación de las capas base de una arquitectura por capas   | 47  |
| <b>Figura 17:</b> Representación de una arquitectura por microservicios  | 48  |
| <b>Figura 18:</b> Diagrama de secuencia del modelo vista controlador   | 49  |
| <b>Figura 19:</b> Diagrama de arquitectura para el patrón servicio web sencillo en AWS   | 51  |
| <b>Figura 20:</b> Patrón de arquitectura para aplicación web en tres niveles sobre AWS   | 52  |
| <b>Figura 21:</b> Patrón de arquitectura puerta de enlace API con múltiples puerta de enlace API, cada microservicio tiene su propia API   | 53  |

|   |    |
|---|----|
| <b>Figura 22:</b> Patrón de arquitectura puerta de enlace API con API única, microservicios conectados a la misma puerta de enlace            | 54 |
| <b>Figura 23:</b> Diseño de arquitectura para la herramienta web y el sitio informativo utilizando docker en una instancia de EC2             | 59 |
| <b>Figura 24:</b> Diseño de arquitectura para sitio web informativo diseñado en WORDPRESS   | 60 |
| <b>Figura 25:</b> Diseño de arquitectura para la herramienta web utilizando API Gateway y Lambdas   | 61 |
| <b>Figura 26:</b> Diseño de arquitectura final para el sitio web estático e informativo y la herramienta para la enseñanza de la programación | 62 |
| <b>Figura 27:</b> Costo mensual de los servicios utilizados por región (Ohio) en AWS  | 64 |
| <b>Figura 28:</b> Gráfica de costos mensuales generados por los servicios utilizados mensualmente en AWS                                      | 64 |
| <b>Figura 29:</b> Primera propuesta para la base de datos de la herramienta   | 65 |
| <b>Figura 30:</b> Esquema de base de datos NoSQL a ser implementado en DynamoDB   | 66 |
| <b>Figura 31:</b> Esquema de base de datos NoSQL a ser implementado en MongoDB o DocumentDB   | 68 |
| <b>Figura 32:</b> Diseño UX para pantalla de login cuando se ha validado que el usuario es de tipo Docente                                    | 71 |
| <b>Figura 33:</b> Diseño UX para pantalla de inicio, después del login. Aquí el profesor contará con las acciones que se muestran en pantalla | 71 |
| <b>Figura 34:</b> Diseño UX para pantalla Ver Estudiantes, para visualizar los estudiantes asociados a una clase                              | 72 |
| <b>Figura 35:</b> Diseño UX para pantalla de login cuando se ha validado que el usuario es de tipo Estudiante                                 | 72 |
| <b>Figura 36:</b> Diseño UX para pantalla con módulos asignados a un estudiante   | 73 |
| <b>Figura 37:</b> Diseño UX para pantalla de progreso e interacción con el avatar   | 73 |
| <b>Figura 38:</b> Diseño UX para determinar sección de ayuda que facilitará el contacto directo con el profesor                               | 74 |
| <b>Figura 39:</b> Diseño UX para pantalla de juego, creación de código y ejecución del mismo  | 74 |
| <b>Figura 40:</b> Diagrama de despliegue de herramienta web y sitio web informativo   | 76 |
| <b>Figura 41:</b> Ejercicio para la identificación de elementos y separación utilizando conjuntos   | 79 |
| <b>Figura 42:</b> Imagen de referencia para la construcción del algoritmo para describir el paso a paso de lavarse los dientes                | 80 |
| <b>Figura 43:</b> Ejercicio de identificación del elemento faltante de la secuencia   | 81 |

|  |    |
|--|----|
| <b>Figura 44:</b> Ejercicio de traducción de figuras a lenguaje natural presente en el taller de conocimientos previos   | 81 |
| <b>Figura 45:</b> Ejercicio para la creación del camino desde el inicio hasta el personaje                               | 82 |
| <b>Figura 46:</b> Avatar de apoyo el que se muestra en la herramienta web  | 83 |
| <b>Figura 47:</b> Ejercicio de secuencia de pasos a convertir en diagrama de flujo                                       | 83 |
| <b>Figura 48:</b> Diagrama de flujo creado por uno de los niños participantes de las pruebas                             | 84 |
| <b>Figura 49:</b> Estudiante haciendo uso de la herramienta web Mi Superpoder es la Programación                         | 85 |
| <b>Figura 50:</b> Gráfica resultado de las respuestas para la pregunta asociada a facilidad de uso de la herramienta     | 86 |
| <b>Figura 51:</b> Gráfica resultado de las respuestas para la pregunta asociada a la apariencia visual de la herramienta | 87 |
| <b>Figura 52:</b> Gráfica resultado de las respuestas para la pregunta asociada a qué le agregarías a la herramienta     | 88 |
| <b>Figura 53:</b> Gráfica resultado de las respuestas para la pregunta asociada a dispositivos utilizados                | 88 |

## LISTADO DE ANEXOS

|   | Pág |
|---|-----|
| <b>Anexo 1:</b> Documento de especificación de requisitos de la herramienta | 97  |
| <b>Anexo 2:</b> Documento de diseño de juego - Game Design Document         | 128 |
| <b>Anexo 3:</b> Cuestionario de pruebas de la herramienta                   | 151 |
| <b>Anexo 4:</b> Políticas de privacidad                                     | 159 |
| <b>Anexo 5:</b> Interfaces de alta fidelidad, diseño UX                     | 163 |

## INTRODUCCIÓN

Gran parte de las escuelas y colegios de Colombia están transformando sus planes de estudio con el fin de involucrar a la computación y tecnología, ya que se reconoce su uso en la realización de diversas actividades cotidianas del ser humano, llegando a ser los dispositivos electrónicos e internet considerados como imprescindibles. Toda esa revolución, ha generado una gran demanda en campos de tecnología y actividades de investigación y desarrollo (I+D) [1], motivo por el cual se busca promover la enseñanza de la programación para inclinar y despertar el interés de los niños y jóvenes en áreas relacionadas a la informática y el software.

En Colombia, a pesar de que el gobierno resalta la inversión en tecnología y transformación digital en el área de la educación, en el país todavía existen sedes educativas que no cuentan con infraestructura para la enseñanza de la tecnología e informática. En el caso de las escuelas oficiales solamente el 33.9% de 40.276 instituciones públicas cuentan con bienes y servicios TIC. Estos recursos en gran parte de las escuelas, son utilizados al menos una vez por semana [2].

En el país es posible encontrar modelos educativos que aún no han tomado el paso de evolucionar, teniendo poco interés en proporcionar contenidos de gran impacto para el desarrollo personal y profesional de los niños y jóvenes. Las actividades de enseñanza se centran en ciertas áreas del conocimiento, las cuales tienen baja relevancia en el desarrollo del estudiante [3], y además no tienen campo de acción en la tecnología e informática.

Con el fin de promover conocimientos en tecnología a niños y jóvenes, diversas entidades públicas y privadas se han enfocado en enseñar programación a través de cursos extracurriculares, algunos de ellos siendo de pago. También las escuelas y colegios han decidido integrar herramientas libres para la enseñanza del código. A pesar de los esfuerzos, gran parte de los niños y jóvenes que asisten a este tipo de actividades no cuentan con los recursos en casa para desempeñar los ejercicios de práctica, lo que lleva al abandono.

A lo anterior se suma, que en Colombia los niños y jóvenes prefieren realizar actividades de ocio cuando disponen de dispositivos electrónicos e internet [4], desaprovechando el potencial de ser creadores de tecnología más allá de consumidores. Algunos de ellos consideran que el internet está basado en las redes sociales, plataformas de streaming y juegos en línea, ya sea por el desconocimiento o poco interés en aprender sobre conceptos de tecnología e informática en el tiempo libre.

A partir de las falencias encontradas en las metodologías, disponibilidad de recursos electrónicos, costos de los talleres particulares y desaprovechamiento del tiempo libre por

parte de los niños y jóvenes, nace el proyecto Mi Superpoder es la Programación, con el fin de impartir cursos de computación, tecnología, informática y programación, para integrar nuevos conocimientos que puedan ser utilizados en su día a día y en el futuro. En los talleres se hace uso de diversas herramientas ejecutables desde la web, las cuales pueden ser libres o de pago, con el fin de enseñar desde los fundamentos, secuencias y lógica, hasta la solución de problemas por medio de código.

A través de dichos talleres, se busca dejar en evidencia los beneficios de aprender programación desde cortas edades, como lo son el desarrollo de la lógica computacional para la solución de problemas de la cotidianidad hasta la creatividad [5], preparando a los niños y jóvenes para el futuro.

Considerando que se requiere hacer uso de múltiples herramientas para llevar a cabo los talleres, y resaltando que gran parte de los niños y jóvenes del país acceden a internet móvil para desarrollar las tareas en casa desde un celular; en este proyecto de grado se desarrolló el prototipo de una herramienta web para la enseñanza de la programación en niños y jóvenes, la cual se ajusta a los dispositivos disponibles para su acceso. Esta herramienta es posible adquirirla de manera gratuita, y se enfoca en incrementar las habilidades desde lo fundamental hasta lo más avanzado desde una única herramienta, sin necesidad de incluir elementos adicionales para lograr una formación completa en el área de la computación, tecnología, informática y programación.

La herramienta construida se puede utilizar en ambiente escolar o de manera independiente, siendo ejecutable desde un celular o computador, además de llamativa para los niños y jóvenes. Esto con el fin de captar su interés y atención en esta área, promoviendo el buen uso del tiempo libre y la posibilidad de aprender con los recursos disponibles en el entorno. La herramienta es el recurso a utilizar en los talleres del programa *Mi Superpoder es la Programación*, para desarrollar e impartir sus clases.

## 1. DEFINICIÓN DEL PROBLEMA

### 1.1 PLANTEAMIENTO DEL PROBLEMA

Cada día se hace más evidente el hecho de una revolución tecnológica la cual exige personas capacitadas y formadas en esta área, principalmente en la programación. Por esta razón, instituciones públicas y privadas suman esfuerzos, con el fin de enfocar a los niños y jóvenes del país en aprender contenidos relacionados, que les permita adquirir habilidades en la tecnología y programación. Además, algunos investigadores en esta área, han determinado que a través del aprendizaje de la programación es posible fortalecer y adquirir conocimientos relacionados a las matemáticas y computación, los cuales mejoran la creatividad, el razonamiento sistemático y el trabajo colaborativo [6].

La creación de herramientas para vincular a los niños y jóvenes con la tecnología y programación viene desde los años 80 con la aparición de Logo [7]. Estas herramientas tienen como propósito enseñar programación y motivar el desarrollo lógico y computacional de los niños y jóvenes, a los cuales se les considera nativos digitales por nacer en una época donde la tecnología está en auge [8]. En los países considerados como desarrollados, la acogida de la tecnología ha sido numerosa y las escuelas tienen en sus planes de estudio a la programación como un curso de base. Sin embargo, en Colombia aprender programación aún es considerado un pasatiempo.

A pesar de los esfuerzos por vincular a los niños y jóvenes del país con la tecnología, se han identificado diversos factores que pueden afectar directamente el uso de dichas herramientas e interés en la programación, como lo son: 1) no contar con un computador en casa, 2) no tener acceso a internet; 3) costos de algunos talleres y herramientas; 4) poco interés en los contenidos asociados a la informática y programación, 5) pérdida de la motivación por frustración; y 6) otros componentes vinculados al factor socio-cultural.

Algunos colegios en Colombia, han decidido implementar herramientas como Code.org y Scratch las cuales son gratuitas y tienen algunos ejercicios predefinidos, lo que facilita la enseñanza del código. Con su uso se espera el incremento de las habilidades de lógica y programación en los niños y jóvenes. Sin embargo, estudios como el realizado por la Universidad Baskent en Turquía [9] o el instituto de ciencias Weizmann [10], determinaron que no hay diferencias significativas entre los niños y jóvenes que hacen uso o no de la herramienta, ya que no causó incremento en el desarrollo de habilidades para resolución de problemas y de pensamiento reflexivo como se esperaba, además se evidenció que aunque hay facilidad de entendimiento en algunos conceptos, en otros no influyó en lo absoluto.

Las herramientas actuales buscan enseñar a los niños y jóvenes a codificar, sin tener en cuenta el paso previo, el cual consiste en el desarrollo de la lógica y la abstracción de problemas. Además de otros componentes de utilidad, los cuales otorgan fundamentos y bases sólidas al futuro programador. Se ha identificado, que la mayoría de herramientas diseñadas para niños y jóvenes funciona por bloques, dejando de lado en muchas ocasiones la limpieza del código, por lo que es posible encontrar en una solución, problemas de estructuración y orden lógico en programas construidos en bloques [11]. Estos problemas deben ser corregidos a tiempo, para que al momento de hacer la transición a un lenguaje de programación formal, no se implementen soluciones con los mismos problemas.

A diario nos encontramos con nuevas herramientas ya sean gratuitas o de pago, que siguen basándose en la enseñanza de conceptos de programación a través de juegos por bloques, dejando de lado algunas bases iniciales y fundamentales. Las nuevas invenciones para fortalecer la educación en esta área debe evolucionar, enseñar no solo conceptos asociados netamente al código, sino que además se aprenda de conceptos de computación e informática, previos a la creación de código. La enseñanza de la programación debe enfocarse a los niños y jóvenes a no ser solo codificadores, sino a tener conocimientos en lógica y computación que les facilite la creación de aplicaciones.

Es importante resaltar que la mayoría de herramientas para enseñar programación, están dirigidas para ser utilizadas en un computador. En Colombia, gran parte de la población accede a recursos y contenidos tecnológicos por medio de sus dispositivos celulares, lo cual es una limitante para su uso, ya que la mayor parte de personas se conecta desde su celular [4], utilizando los servicios de internet por paquetes de datos que le ofrece el operador.

Por medio de la experiencia adquirida en los talleres dictados a través del proyecto *Mi Superpoder es la Programación*, fue posible determinar que las problemáticas claves de las herramientas existentes se basan principalmente en la poca posibilidad de acceder desde un dispositivo móvil, recurso con el cual la gran mayoría de niños y jóvenes del país cuenta; además de las pocas posibilidades en formación de bases computacionales fundamentales, previo al paso de un lenguaje de programación formal. Gran parte de los niños y jóvenes que asisten a los talleres, desconocen de los temas iniciales de la informática como lo son las partes de un computador y su funcionamiento, lo cual no es posible aprenderlo desde las herramientas existentes.

En este proyecto de grado se buscó, por medio del desarrollo de una herramienta web, cubrir las necesidades identificadas al hacer uso de otras herramientas para llevar a cabo las clases de computación y programación. Esto con el fin de crear actividades que permitan mejorar las habilidades lógicas antes y durante el proceso de codificación. Siendo completamente gratuita, además de accesible desde cualquier dispositivo principalmente computadores y celulares, enfocándose principalmente en fortalecer los componentes

iniciales de la computación, necesarios para alcanzar etapas avanzadas de la programación donde se maneje código formal.

## **1.2 FORMULACIÓN DEL PROBLEMA**

- **¿Cómo enseñar a los niños y jóvenes en los talleres de *Mi Superpoder es la Programación*, los conceptos necesarios para aprender programación desde lo fundamental hasta lo más avanzado, sin requerir de múltiples herramientas y teniendo la posibilidad de acceder desde cualquier dispositivo ?**

## **2. OBJETIVOS DEL PROYECTO**

### **2.1 OBJETIVO GENERAL**

Desarrollar el prototipo de una herramienta web para la enseñanza de la programación en niños y jóvenes, donde sea posible aprender desde los conceptos básicos fundamentales hasta los más avanzados, accediendo también desde dispositivos móviles, sin requerir del uso de múltiples herramientas.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Identificar el estado actual de las herramientas gratuitas y de pago existentes para la enseñanza de la programación en niños y jóvenes, con el fin de identificar sus debilidades y establecer los requisitos funcionales y no funcionales del prototipo de la herramienta a construir.
- Analizar al menos 3 patrones de arquitectura de software y definir la arquitectura final a utilizar en el desarrollo del prototipo de la herramienta.
- Construir el prototipo de una herramienta web para la enseñanza de la programación en niños y jóvenes, la cual cuente con un módulo inicial donde sea posible aprender sobre las primeras bases de la computación (Secuencia y diagramas de flujo).
- Evaluar la herramienta con mínimo 10 niños de edades entre los 6 y 12 años, con capacidades lectoras, en un ambiente completamente virtual, con el fin de evaluar la funcionalidad, experiencia de usuario y motivación en el uso del prototipo por parte de los niños.

### **3. ALCANCE**

Para alcanzar exitosamente los objetivos de este proyecto, la herramienta a construir para la enseñanza de la programación en niños y jóvenes, deberá contar con al menos un nivel donde sea posible por medio de diagramas de flujo, pseudocódigo y conceptos básicos de programación, dar instrucciones y movimientos a un personaje para alcanzar un objetivo con éxito.

La herramienta podrá accederse desde un navegador web y deberá contar con dos perfiles uno de ellos necesario para la asignación de actividades (perfil profesor) y uno el cual permita a los niños realizar sus actividades (perfil estudiante).

Para llevar a cabo la evaluación de la herramienta, se considera una muestra de 10 niños de edades entre los 6 y 12 años, con capacidades lectoras e interés en la programación, con el fin de obtener información sobre la experiencia de usuario, facilidad de los ejercicios a realizar y la motivación que esta genera.

Se debe considerar que inicialmente la herramienta será un prototipo de evaluación por lo tanto no será 100% operacional.

#### 4. MARCO TEÓRICO

La enseñanza de la programación se ha vuelto tendencia y cada vez más instituciones educativas y universidades reconocidas a nivel mundial promueven su aprendizaje. A través de la programación se ha identificado que poseer conocimientos en esta área, proporciona habilidades para la resolución de problemas del día a día. Por esta razón se han creado diversas estrategias, metodologías y herramientas para la enseñanza de todo lo relacionado con la programación dirigido especialmente para niños y jóvenes, donde sea más divertido y fácil aprender.

Teniendo en cuenta las problemáticas de los niños y jóvenes de Colombia descritas previamente en el [capítulo 1](#), donde se habla principalmente de las limitaciones para acceder a recursos tecnológicos desde sus escuelas o el hogar, se realiza un análisis de las herramientas más utilizadas para la enseñanza de la programación hoy en día. Esto con el fin de identificar el estado actual, características y necesidades que no logran cubrir, en países poco desarrollados en el área de la tecnología como lo es Colombia. Además, se destaca la importancia de considerar a la población vulnerable en el ámbito educativo, el cual es un grupo poblacional excluido del sistema educativo por razones culturales o socioeconómicas<sup>1</sup>.

Para comprender las necesidades de la herramienta a construir, no solo se analizan las herramientas más utilizadas, sino que además se describen componentes claves e importantes en trabajos relacionados. Se resalta también el impacto de la pandemia vivida en 2020, que obligó a modificar el modelo educativo y tuvo un impacto significativo en las escuelas que carecen del componente tecnológico.

Para garantizar un análisis efectivo de las herramientas para la enseñanza de la programación, se deben reconocer las necesidades y formas en que los niños y jóvenes se identifican. En Colombia, se establece que los niños y niñas tienen edades comprendidas entre los 0 y 12 años, mientras que los jóvenes se encuentran entre los 12 y 18 años<sup>2</sup>.

---

<sup>1</sup> Población vulnerable, MINEDUCACIÓN: <https://www.mineducacion.gov.co/1621/article-82770.html>

<sup>2</sup> Concepto unificado niñez y adolescencia, ICBF: [https://www.icbf.gov.co/cargues/avance/docs/concepto\\_icbf\\_0027891\\_2010.htm](https://www.icbf.gov.co/cargues/avance/docs/concepto_icbf_0027891_2010.htm)

#### 4.1 Identificación del estado actual de las herramientas existentes para la enseñanza de la programación en niños y jóvenes

Existen diversas herramientas para la enseñanza de la programación en niños y jóvenes, algunas de ellas son gratuitas y están pensadas para ejecutarse principalmente en un computador. Algunos investigadores han evaluado la efectividad de las herramientas, a través de factores como la motivación, interés, desarrollo de las habilidades lógicas y creativas para la solución de problemas computacionales.

Las herramientas que se nombran a continuación, son algunas de las más utilizadas y conocidas. Además, han sido objeto de estudio en múltiples investigaciones, adicionalmente, algunas de ellas son las herramientas usadas en los talleres de *Mi Superpoder es la Programación*, estas son: 1) Scratch; 2) Pilas Bloques; 3) Codemonkey; 4) Code.org; y 5) Tynker. En el análisis se tiene en cuenta el videojuego Rabbids Coding, el cual está enfocado en la enseñanza de la programación por medio de bloques, puesto que su dinámica, modo de interacción y arte conceptual servirán de referencia de la herramienta web creada.

##### Scratch

Es un lenguaje de programación desarrollado por el MIT Media Lab<sup>3</sup>, el cual permite por medio de componentes visuales como bloques, manipular escenarios, dar comportamientos a personajes y modificar sonidos o videos. Cuenta con su propia herramienta de edición de código, y está diseñado para ser intuitivo y de fácil uso, permitiendo a personas sin experiencia en programación utilizarla. Pertenece a la categoría de software libre, ya que puede ser modificado y utilizado libremente sin necesidad de realizar pagos de membresías [16]. Es ampliamente reconocido entre los niños y jóvenes, ya que diversas escuelas y colegios a nivel mundial, hacen uso del entorno y lenguaje de programación .

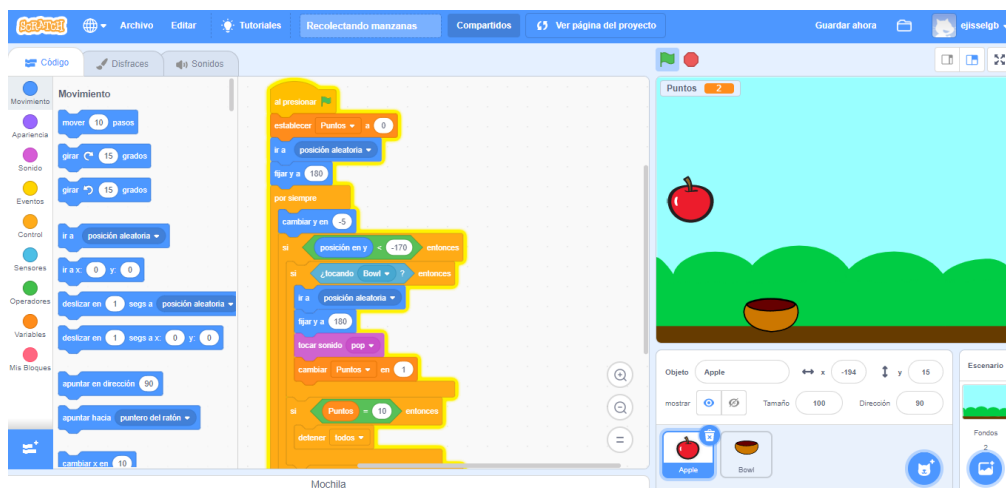
A pesar de ser muy utilizado para enseñar las bases de la programación, la primera impresión causada en los usuarios novatos es que puede ser compleja de utilizar, ya que cuenta con múltiples tipos de bloques para realizar diferentes tipos de acciones, con las cuales el usuario no se encuentra familiarizado. Adicionalmente, a medida en que mejoran los conocimientos, se debe recurrir a lenguajes de programación más avanzados, para aumentar de nivel o desarrollar programas de mayor complejidad y detalle.

El entorno de programación no tienen la posibilidad de ser accedido fácilmente desde un celular, y aunque cuenta con aplicación móvil, no es compatible con todos los dispositivos y

---

<sup>3</sup> Herramienta Scratch: <https://scratch.mit.edu/>

sistemas operativos. Requiere de Java para ser ejecutado de manera local en un computador y si los programas son muy complejos, es posible que se generen archivos de gran tamaño. Algunos de los textos de la interfaz no se encuentran completamente traducidos al idioma seleccionado del usuario, al igual que la documentación. Un ejemplo del entorno de programación y el uso del lenguaje en Scratch se muestra en la Figura 1.



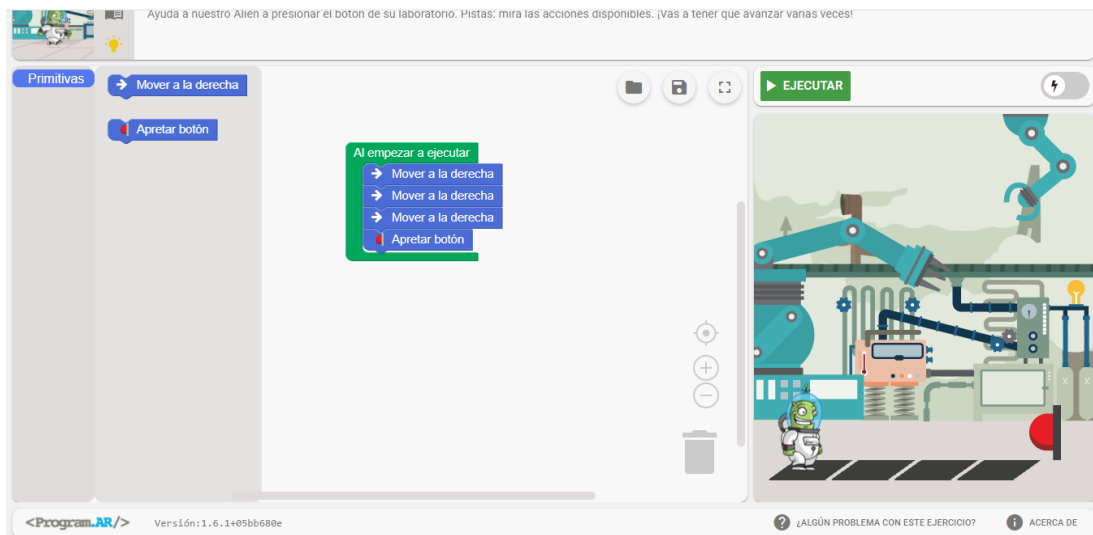
**Figura 1:** Imagen del entorno de desarrollo web de scratch y ejemplo de programación en bloques, tomado de <https://scratch.mit.edu/>

## Pilas Bloques

Es una herramienta para la enseñanza de la programación por medio de bloques, la cual hace parte del programa educativo argentino Program.Ar<sup>4</sup>. Este proyecto busca que los niños aprendan todo sobre programación en sus escuelas y hogares. La herramienta Pilas bloques cuenta con su propio entorno de programación, incluyendo al menos 40 actividades que abordan diferentes conceptos de programación, que van desde la instrucción más básica a la más compleja [17]. Este proyecto y la herramienta, están enfocados en niños que se encuentran cursando grados de primaria.

Pilas bloques tiene todos sus ejercicios prediseñados, y no brinda la posibilidad de que los niños puedan crear nuevas actividades haciendo uso de bloques. Es necesario hacer uso de otras herramientas para crear código en bloques de manera personalizada. Un ejemplo de los ejercicios que tiene la herramienta lo podemos ver en la Figura 2, donde el niño debe utilizar bloques para lograr la acción esperada.

<sup>4</sup> Programar en casa, proyecto argentino que busca promover la enseñanza de la programación a través de bloques <http://pilasbloques.program.ar>



**Figura 2:** Ambiente de programación para pilas bloques y muestra del primer ejercicio, tomado de <http://pilasbloques.program.ar/online/#/desafio/1>

Program.Ar evolucionó a causa de la pandemia de COVID 19 creando la iniciativa Programar en Casa, donde se integraron más actividades que permiten comprender cómo funciona la tecnología. Cuenta con documentación propia la cual es clara y entendible, siendo esta en español. La herramienta Pilas Bloques hace uso del API Pilas Engine web<sup>5</sup>.

## Codemonkey

Es un entorno para el aprendizaje de la programación basado en juegos para niños y jóvenes, donde no se requiere experiencia previa para ser utilizada<sup>6</sup>. La herramienta promete que los usuarios, podrán desenvolverse en el mundo de la programación sin problemas al finalizar todos los niveles que se ofrecen. Se pueden encontrar diversos cursos en los niveles de principiante, intermedio y avanzado, contando también con módulos de capacitación para profesores en la misma herramienta.

En los cursos que ofrece esta plataforma es posible encontrar programación por bloques y código basado en CoffeeScript<sup>7</sup> o Python, proporcionando una experiencia más cercana a lo que sería programar en código formal. Sin embargo, debe pagarse una licencia de manera anual para ser utilizada. Adicionalmente, el idioma oficial de la herramienta es el inglés por lo que cuenta con algunos textos de la interfaz y documentación como guías en pdf y videos sin traducir. En la Figura 3 se muestra el entorno de programación de Codemonkey.

<sup>5</sup> API de la herramienta Pilas Engine Web: <https://app.pilas-engine.com.ar/#/api>

<sup>6</sup> Codemonkey, herramienta para la enseñanza de la programación: <https://www.codemonkey.com/>

<sup>7</sup> What is CoffeeScript? <https://help.codemonkey.com/en/articles/3593792-what-is-coffeescript>



**Figura 3:** Entorno de programación en Codemonkey y ejemplo de realización de cada uno de los niveles usando código, tomado de <https://app.codemonkey.com/challenges/16>

## Code.org

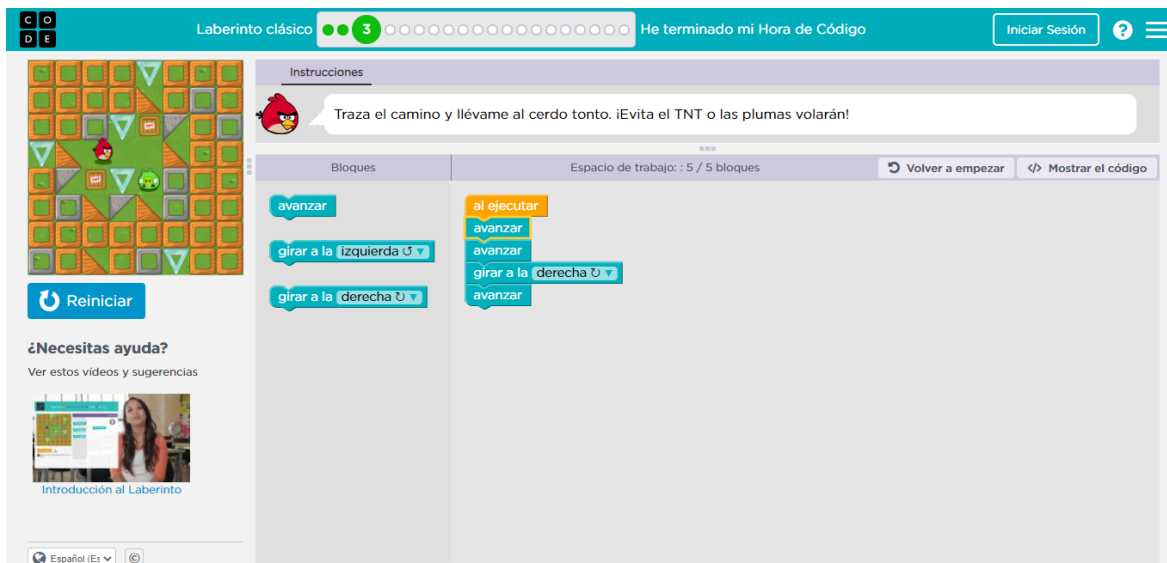
Es una organización sin ánimo de lucro, enfocada en llevar la programación a todas las escuelas del mundo. Ofrece por medio de su sitio web cursos enfocados en la computación básica, lo cual se aprende por medio de juegos a superar con bloques<sup>8</sup>. Es de libre acceso y es financiada por grandes empresas como Amazon, Facebook y Google.

Cuenta con su propio entorno de programación embebido en el sitio web y además los niveles a realizar basan sus historias y personajes en películas o videojuegos conocidos, lo que llama la atención de los niños y jóvenes. Code.org es el promotor de la hora del código, el cual busca por medio de una clase de introducción de una hora en computación, demostrar que todo el mundo sin importar su edad, puede aprender a programar.

Aunque algunas de sus actividades están prediseñadas, es posible crear proyectos personalizados, teniendo una estructura muy similar a Scratch. La herramienta cuenta con algunas desventajas relacionadas a que gran parte de su documentación se encuentra en el idioma original inglés y al momento de crear una cuenta no es posible identificar si el rol será de estudiante o profesor, lo que podría ser un poco confuso para los usuarios que ingresan por primera vez. El entorno de la herramienta se observa en la Figura 4.

---

<sup>8</sup> Sobre Code.org y propósito en la enseñanza de la programación <https://code.org/international/about>



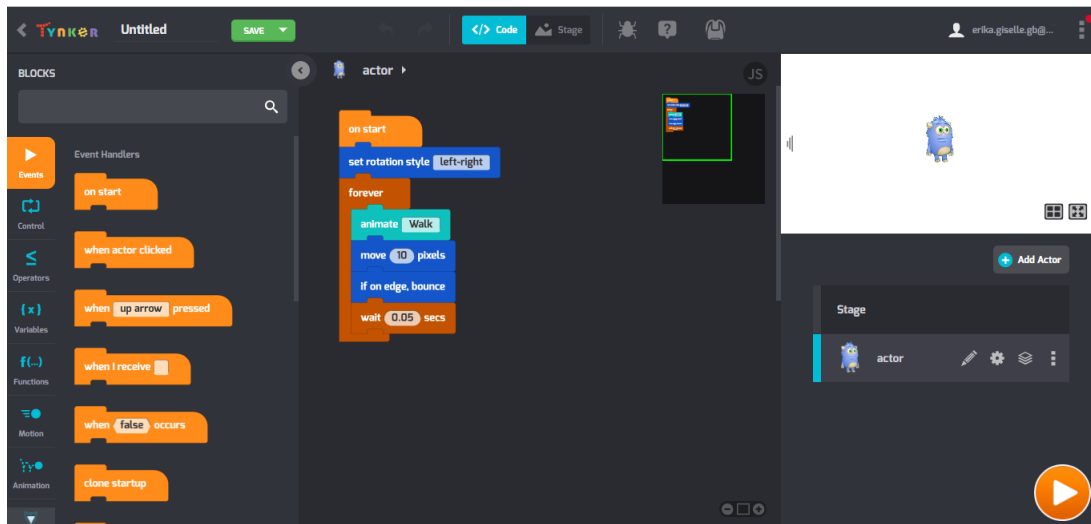
**Figura 4:** Ambiente de programación en Code.org, hora del código Angry Birds, tomado de <https://studio.code.org/hoc/1>

## Tynker

Es una herramienta que ofrece un plan de estudio completo para niños y jóvenes de primaria y secundaria, contando con una versión dirigida tanto al uso doméstico como escolar<sup>9</sup>. Está basado en juegos conocidos, como por ejemplo minecraft y la forma de crear código es a partir de bloques, muy similar a scratch. Propone el uso de código como Javascript, HTML, CSS y Python en los niveles más avanzados [18]. La herramienta es licenciada, el cobro se hace mes a mes, trimestral o en un solo pago, que permitirá el uso indefinido de la misma. Solo unos pocos ejercicios son gratuitos, para acceder a todos los recursos es necesario adquirir la licencia.

Esta plataforma está completamente en inglés, siendo de difícil uso para los niños y jóvenes de Colombia. Adicionalmente debido a la cantidad de información de la interfaz, para usuarios principiantes, la navegación y entendimiento de las actividades a realizar puede ser difícil. Por esta razón Tynker es considerada como una herramienta para niños y jóvenes con un poco más de experiencia en codificación en comparación con las herramientas descritas anteriormente. El entorno de programación de Tynker se muestra en la Figura 5.

<sup>9</sup> Sitio web oficial de Tynker, <https://www.tynker.com/>



**Figura 5:** Ambiente de programación en Tynker y proyecto de ejemplo en la versión gratuita, tomado de <https://www.tynker.com/>

## Rabbids Coding

Es un videojuego para niños enfocado en la enseñanza de la programación, el cual está basado en lenguaje por bloques [19]. El videojuego se enfoca en enseñar conceptos básicos de codificación por medio de una historia y objetivos a resolver dentro del juego, a partir de instrucciones que permitirán superar una misión exitosamente.

A pesar de ser un videojuego gratuito, puede ser complicada su instalación en usuarios poco expertos, debido a que se requiere de un cliente (Uplay - Ubisoft) para ser descargado. El videojuego también se encuentra disponible para celulares, lo que hace más fácil el proceso de instalación. Además no requiere de internet para ser utilizado.

Cuenta con una interfaz clara e intuitiva, sin embargo, no cuenta con idioma español, lo que puede dificultar el entendimiento de las instrucciones a la mayoría de los usuarios del país.



**Figura 6:** Videojuego Rabbids Coding en uno de los niveles, el cual es solucionado por bloques, figura tomada del videojuego [19]

El análisis de las herramientas existentes para aprender a programar permitió encontrar y determinar las características faltantes que se debían tener en cuenta en la herramienta construida, con el fin de cubrir las necesidades presentadas por los niños y jóvenes al momento de aprender temas relacionados a la programación. En la Tabla 1, se muestra a mayor detalle la información obtenida, la cual fue clave en la definición de los requisitos de la herramienta.

Para determinar si una herramienta cumple con todas las características esperadas, se utilizan símbolos para indicar el nivel de cumplimiento. Por ejemplo, el símbolo  se utiliza para indicar que la característica se cumple en su totalidad, para indicar si la característica se cumple parcialmente o está incompleta se utiliza el símbolo /X, y si finalmente no cumple se utiliza el símbolo .

**Tabla 1:** Extracción de características de las herramientas más utilizadas, para la construcción y definición de requisitos

| Herramienta    | Características |                          |            |                |  |                                     |                            |          |
|----------------|-----------------|--------------------------|------------|----------------|--|-------------------------------------|----------------------------|----------|
|                | Ambiente web    | Portabilidad (app móvil) | Responsive | Idioma Español | Ejercicios de algoritmos (previos a la programación) | Posibilidad de incrementar el nivel | Código diferente a bloques | Gratuita |
| Scratch        | ✓               | ✓                        | ✓/X        | ✓/X            | X  | ✓/X                                 | X                          | ✓        |
| Pilas Bloques  | ✓               | X                        | ✓          | ✓              | X  | X                                   | X                          | ✓        |
| Codemonkey     | ✓               | X                        | ✓/X        | ✓/X            | ✓/X  | ✓                                   | ✓                          | X        |
| Code.org       | ✓               | X                        | ✓          | ✓/X            | X  | ✓/X                                 | ✓/X                        | ✓        |
| Tynker         | ✓               | ✓                        | ✓ / X      | ✓/X            | X  | ✓                                   | ✓                          | X        |
| Rabbids Coding | X               | ✓                        | X          | X              | ✓/X  | X                                   | X                          | ✓        |

Algunas herramientas cuentan con traducción al idioma Español. Sin embargo, algunos textos o documentación de apoyo se encuentran en su idioma original Inglés, contando con una traducción incompleta o parcial al idioma de los usuarios. Se considera que el idioma es importante ya que la mayoría de los niños y jóvenes no se desenvuelven correctamente en una segunda lengua o no tienen ningún conocimiento, lo que podría dificultar el entendimiento de las guías y material de apoyo. De igual manera ocurre con criterios como el responsive, donde la mayoría de las plataformas no se adaptan correctamente al dispositivo siendo complicado y poco amigable realizar las actividades.

La mayoría de las herramientas analizadas no cuentan con un módulo de entrenamiento previo a la codificación, en donde se fortalezca la lógica y habilidades matemáticas. Esta característica es necesaria, debido a que enseñar únicamente a codificar no proporciona las habilidades requeridas para ser un programador. La diferencia está en que un codificador hace uso de la tecnología para resolver una problemática particular escribiendo código, mientras que un programador crea y da soluciones innovadoras a través de algoritmos complejos y sofisticados<sup>10</sup>.

<sup>10</sup> What's the difference between a Programmer, Coder, Developer and Software Engineer?  
<https://www.linkedin.com/pulse/whats-difference-between-programmer-coder-developer-software-hasan/>

El ítem de análisis relacionado al incremento de nivel, permite determinar si en la herramienta utilizada es posible crear código personalizado o adquirir conocimientos complejos sin necesidad de irse a otra herramienta.

#### **4.2 Trabajos de investigación relacionados a la enseñanza de la programación**

La enseñanza de la programación en niños y jóvenes se ha implementado como herramienta didáctica para aprender conceptos de matemáticas. Esta estrategia es utilizada, ya que se ha demostrado que los procesos secuenciales aportan a otras áreas del conocimiento a parte de la tecnología [20]. Las herramientas y lenguajes más utilizados, para enseñar los inicios de la programación y fortalecer conocimientos en matemáticas y geometría son los basados en bloques, un ejemplo de esto es Scratch.

El uso de videojuegos para la enseñanza de la programación cada día toma más fuerza, ya que la gamificación y la interacción, además de la posibilidad de visualizar el conjunto de comandos e instrucciones dadas en un personaje que realiza eventos, atrae más la atención de los niños, en comparación con la enseñanza tradicional de lenguajes de programación reales y principios de algoritmia [21].

En Colombia, el auge de la tecnología ha llevado a instituciones públicas y privadas a promover la enseñanza de la programación. En el trabajo *“Analizar por qué se debe enseñar la lógica de programación a niños y jóvenes en los hogares de la ciudad de Medellín”*, se resaltan los principales motivos de aprender programación para fortalecer otras áreas del conocimiento, ya que proporciona pensamiento analítico y de resolución de problemas. Sin embargo, no se deja de lado, que cada herramienta existente para la enseñanza de código tiene un objetivo particular, por lo que se debe interactuar con más de una y analizar previamente la más acorde según la necesidad del estudiante, experiencia con la tecnología y concepto de interés a enseñar [22].

Las herramientas más utilizadas son las gratuitas o de código abierto, basadas en videojuegos que facilitan su acceso desde un navegador y no requieren importantes especificaciones técnicas. En el trabajo *“La enseñanza-aprendizaje del pensamiento computacional en edades tempranas: una revisión del estado del arte”*, se nombran otras herramientas didácticas para la enseñanza de la programación, las cuales no hacen uso de un computador, como por ejemplo el libro Hello Ruby de Linda Liukas, donde se tienen actividades asociadas principalmente al desarrollo de la lógica por medio de secuencias [23]. También se nombra el uso de robots, para la enseñanza de la programación, ya que permiten por medio de un software definir el conjunto de comandos a seguir.

### 4.3 Impacto de la enseñanza de la computación y programación en niños y jóvenes

La programación posee características de interés para los investigadores, por esta razón es posible encontrar diversos autores que han enfocado sus esfuerzos en el estudio y análisis del impacto de la programación desde edades tempranas. En el caso del trabajo de investigación *“Developing computational thinking with educational technologies for young learners”* [24], se resalta el uso de tecnologías para incentivar el desarrollo del pensamiento computacional, la cual es una forma de crear habilidades para la resolución de problemas de la cotidianidad de manera eficiente.

Se considera que la programación es un elemento clave dentro del pensamiento computacional, ya que permite apoyar y cultivar tareas cognitivas asociadas a la computación, las cuales pueden ser realizadas no solo a nivel de un computador sino también con ayuda de personas [25]. Por esta razón, diversas herramientas están orientando su enfoque en el desarrollo del pensamiento computacional.

Se identificó que a través del pensamiento computacional los niños y jóvenes están capacitados para expresar ideas haciendo uso de un lenguaje de programación en concreto. Además de que el pensamiento computacional en la enseñanza de la tecnología, puede tener un impacto en diversas áreas diferentes a la informática como por ejemplo la salud, educación y ciencias, lo que fortalece no solo la solución de problemas a partir de conceptos lógicos sino también desde la imaginación [26].

Las herramientas y tecnologías que buscan desarrollar el pensamiento computacional se enfocan no solo en el componente de programación sino también en el componente motriz, por medio de piezas de rompecabezas que permiten realizar un ensamblaje, como es el caso de los robots LEGO. A través de las herramientas existentes hoy en día, se busca fortalecer el conocimiento de los niños, para evitar un uso indiscriminado de la tecnología sin tener un objetivo claro, ya que puede ocasionar un impacto en la capacidad intelectual no muy positivo [27].

En la enseñanza de la computación y programación aparecen conceptos como el aprendizaje significativo, donde se señala que los nuevos conocimientos aprendidos se relacionan con los previamente adquiridos, particularmente en la práctica de la programación. Este concepto es estudiado por tres universidades de Chile en el trabajo *“Experiencias prácticas con el uso del lenguaje de programación Scratch para desarrollar el pensamiento algorítmico de estudiantes en Chile”* [28]. Esto cobra importancia, debido a que las personas aprenden mejor y disfrutan más cuando trabajan en proyectos con significado personal [29]. Este equipo propone el uso del lenguaje y herramienta Scratch para la enseñanza de la programación, ya que cada niño o joven puede crear y diseñar un proyecto en base a sus intereses, gustos y conocimientos.

Los lenguajes de programación y las diversas herramientas enfocadas en enseñar a los niños y jóvenes sobre código, parten del juego como base, lo que es significativo ya que la cantidad de deserciones puede ser menor. El juego causa mayor interés y motivación en el aprendizaje [30]. Lo anterior se describe en el trabajo “Aprendizaje basado en juegos”, donde se describen los beneficios otorgados por el aprendizaje a través de videojuegos, lo cual se logra si el contenido académico pasa desapercibido, donde tiene mayor relevancia la historia del juego, aprendiendo por medio de técnicas de gamificación, desarrollando no solo el pensamiento computacional sino también el aprendizaje significativo.

#### **4.4 Transformación de la educación en tiempos de pandemia**

La tecnología adaptó una nueva forma de vivir e interactuar con el mundo, nos encontramos con una migración del entorno tradicional en donde se necesita la interacción directa, a una virtual usando el internet como medio de comunicación. En lo que concierne a los estudiantes, el uso de internet puede hacerles asumir un rol más activo en el proceso de adquisición de conocimientos, ya que otorga autonomía, lo que crea una red activa donde los estudiantes pueden ser receptores y generadores de saberes [31].

El internet ha permitido que el ser humano rompa el esquema tradicional de comunicación, ya que no solo es un emisor, sino que también está en constante aprendizaje dependiendo de la situación social en la que se encuentre [31]. La generación actual se caracteriza por el uso de tecnologías sofisticadas y las habilidades comunicativas logradas a través de dispositivos electrónicos [32].

La educación virtual pasó de ser algo lejano a una alternativa, ya que facilita el acceso al conocimiento, mitigando obstáculos para aprender, debido a que la educación presencial no puede llegar fácilmente a todos los lugares. Además, busca proporcionar espacios de formación novedosos e interactivos, con el fin de transmitir toda una perspectiva pedagógica, que cause un efecto positivo en el desarrollo de las personas, comunidad y entorno [33].

Sin embargo, en Colombia sigue siendo un obstáculo debido a que la pandemia dejó en evidencia la falta de recursos para educación que poseen los niños y jóvenes del país. Muchos de ellos asistieron a clases virtuales desde sus dispositivos, pero el uso de algunas herramientas y la descarga de archivos enviados por los profesores, dificultaron el proceso de aprendizaje.

#### 4.5 Usabilidad de las herramientas actuales para la enseñanza de la computación

La usabilidad es el concepto que está asociado a la facilidad de uso de una herramienta sin importar la experiencia del usuario [34]. En la actualidad, se ha optado por hacer uso de diversas herramientas para la enseñanza en las escuelas y colegios en Colombia, por lo que el término cada vez toma más fuerza.

A pesar de que existen diversos sitios web pensados en la enseñanza de la programación e informática, algunos son complejos de manejar, haciendo que los niños y jóvenes descarten su uso, ya que pierden el interés en utilizarla, pues si no se cuenta con una manera de reducir los errores que pueda cometer el usuario, o si las tareas son poco claras permitiendo que pueda realizar unas no deseadas, el tiempo de visita en el sitio será muy corto [35].

Para mejorar la usabilidad de un sitio web Nielsen propone 10 principios basados en 1) visibilidad del sistema; 2) lenguaje común entre el sistema y el usuario; 3) libertad y control por parte del usuario; 4) consistencia y estándar; 5) prevención de errores; 6) reconocer es mejor que recordar; 7) flexibilidad y eficiencia de uso; 8) diseño minimalista; 9) permitir al usuario solucionar el error; y 10) Ayuda y documentación [36]. Con el uso de estos principios, las herramientas no solo web sino también de escritorio, pensadas en la educación para niños y jóvenes podrían mejorar significativamente y ser más llamativas para ellos.

Es importante considerar las buenas prácticas para lograr un buen producto tecnológico, que cumpla su objetivo teniendo en cuenta las necesidades del usuario, pero que a su vez pueda ser evaluado para conocer el cumplimiento en la efectividad, eficiencia y satisfacción generada en el usuario [37]. Se debe tener en cuenta el público objetivo, ya que la mayoría de diseñadores al momento de definir las interfaces y los flujos de interacción, consideran que desarrollar tecnología para niños sigue las mismas pautas que en adultos, sin tener en cuenta que los gustos y necesidades son distintos por la edad y entorno en el que se desenvuelven [35].

## 5. OBTENCIÓN Y ANÁLISIS DE REQUISITOS

### 5.1 Captura de información previa para la construcción de los requisitos

Para llevar a cabo la obtención de los requisitos a implementar en la herramienta web para la enseñanza de la programación, se realizaron actividades para la captura de información. Estas se basaron en aplicar un conjunto de preguntas a 15 niños de edades entre los 5 y 14 años, con el fin de conocer sus necesidades respecto a herramientas para aprender sobre programación e informática, satisfacción con las herramientas existentes si es que ya habían hecho uso de ellas previamente y problemas identificados en las que utilizan habitualmente. También se realizó un análisis de las herramientas más utilizadas donde se describen sus funcionalidades, además de la definición de las dinámicas de los ejercicios con las que estas cuentan, las cuales servirán como base de la herramienta a desarrollar.

La convocatoria para la captura de información con los niños y jóvenes se realizó a través de las redes sociales oficiales del proyecto social Mi Superpoder es la Programación y vía correo electrónico, donde se invitó a estudiantes que habían participado previamente de talleres de programación.

En los resultados de las encuestas realizadas a los niños y jóvenes se identificó que algunos consideran que tienen conocimientos en programación, ya que hacen uso del computador o dispositivos móviles. Sin embargo, al momento de dar una definición más puntual del término, la mayoría expresó que la programación no va más allá de dar instrucciones o interactuar con un computador.

La mayoría de los niños y jóvenes indicó que en su escuela o colegio se dicta la materia de sistemas e informática, pero las herramientas vistas están fuertemente relacionadas a la ofimática (word, excel, powerpoint). Solamente 3 de los encuestados indicaron ver materias de programación, pero solo 1 indicó que la herramienta utilizada es el programa de Lego. Los demás niños, consideraron nuevamente que la programación es la interacción con el computador.

A pesar de que los niños no tienen programación en sus currículos, gran parte de los encuestados han participado de clases particulares de programación y han utilizado herramientas como: Codemonkey, Codecombat, Scratch y Codeorg. Algunos han considerado la consola del navegador como herramienta de programación.

De los encuestados, algunos niños y jóvenes han pagado por acceder a clases de programación o hacer uso de las herramientas. Los costos son considerados como

intermedios ni muy bajos, ni muy elevados, siendo posible hacer parte de clases particulares o aprender en casa en el tiempo libre.

A nivel de satisfacción de uso de las herramientas, lo que más llama la atención de los niños y jóvenes, es que las actividades a realizar tienen dinámicas de juego entretenidas y son de fácil uso y entendimiento. Sin embargo, consideran que falta más variedad en los ejercicios, complejidad cuando se avanza de nivel, más posibilidades de personalización y codificación de manera independiente. Algunos definen que las herramientas utilizadas están incompletas, ya que no es posible hacer un proceso de transición, pues se requiere pasar a otras herramientas o tecnologías más avanzadas, para cubrir las necesidades expresadas.

En la revisión de las plataformas existentes para identificar componentes a mejorar, estas fueron descargadas y utilizadas, se analizaron criterios como facilidad de uso, variedad en los ejercicios, costos y documentación de apoyo. Los resultados pueden visualizarse a través de la tabla 2:

**Tabla 2:** Descripción de las herramientas para aprender a programar más utilizadas

|                             |   |
|-----------------------------|---|
| <p><b>Scratch</b></p>       | <p>Es una herramienta para aprender a programar en bloques, visualmente agradable y de fácil interacción. Sin embargo, no todos los componentes de codificación y apoyo están en el idioma nativo del usuario. Contiene múltiples elementos que pueden utilizarse para codificar, aunque el usuario no tenga experiencia, lo que puede parecer complejo para muchos niños y jóvenes en su primera interacción. La web no es completamente responsive, obligando al usuario a tener herramientas adicionales como lo son ScratchJR para el celular. Una ventaja es que la herramienta cuenta con ejecutables para ambiente local en sistemas operativos Windows, Mac y Linux, funcionando sin necesidad de internet.</p> |
| <p><b>Pilas Bloques</b></p> | <p>Esta herramienta cuenta con múltiples ejercicios progresivos, que van aumentando de nivel cada vez que el usuario supera sus objetivos con éxito. Además, es un sitio web responsive. Al finalizar completamente los ejercicios, el usuario debe buscar herramientas adicionales para complementar su formación o mejorar su</p>   |

|                          |   |
|--------------------------|---|
|                          | <p>habilidades. Las actividades se enfocan únicamente en enseñar a codificar en bloques, olvidando que la programación depende de otras competencias previas.</p>   |
| <p><b>CodeMonkey</b></p> | <p>Es una herramienta web para aprender a programar, la cual permite avanzar de código básico en bloques a código real en CoffeeScript o Python. A pesar de permitir incrementar los conocimientos de codificación en una misma plataforma, no cuenta con un módulo de conceptos básicos y teóricos, previos a la etapa de escribir código. Para su uso es necesario el pago de una licencia anual, lo que la hace difícil de obtener para algunos niños y jóvenes de Colombia.</p>   |
| <p><b>Codeorg</b></p>    | <p>Es un sitio web que cuenta con múltiples ejercicios para llevar a cabo la enseñanza de la programación en bloques. Aunque permite realizar actividades personalizadas al finalizar los módulos de capacitación, la documentación no está en el idioma nativo del usuario o las explicaciones se pueden encontrar en texto plano, lo que puede ser difícil para niños que inician en este mundo con bases débiles en la lectura. Posee mucha información en su interfaz, lo que puede saturar al usuario y hacer más complejas las búsquedas. Se requiere de una plataforma más avanzada para acceder a código real o contenido más avanzado.</p> |
| <p><b>Tynker</b></p>     | <p>Es una herramienta de pago, por lo que muchos niños y jóvenes de Colombia no podrían acceder a ella. Se paga de manera mensual, anual o cada 3 años. Adicionalmente sus recursos (tutoriales de apoyo, ejercicios) están en inglés. El contenido de la interfaz es numeroso lo que puede confundir al usuario. Esta herramienta cuenta con la posibilidad de acceder a código real o ejercicios más avanzados, pero no es fácil encontrar documentación de apoyo. Tiene una versión gratuita muy limitada en</p>   |

|                       |   |
|-----------------------|---|
|                       | funcionalidades.  |
| <b>Rabbids Coding</b> | Es un videojuego basado en programación por bloques, multiplataforma y gratuito. A pesar de ser entretenido, interactivo y claro en su objetivo, las instrucciones pueden llegar a no ser muy fáciles de entender por parte del usuario, puesto que están en inglés. Requiere de un cliente de Ubisoft para ser instalado, lo que podría dificultar la descarga en computador por parte de los usuarios |

Posteriormente a la revisión de las herramientas, se establecen algunos criterios a consideración por parte del autor que debe tener el prototipo de la herramienta a desarrollar, para cubrir las necesidades identificadas por los niños y jóvenes encuestados. Estos son los siguientes: 1) facilidad de uso; 2) variedad en los ejercicios; 3) amplia documentación; 4) llamativa, dinámica e interacción y 5) gratuita y de libre acceso.

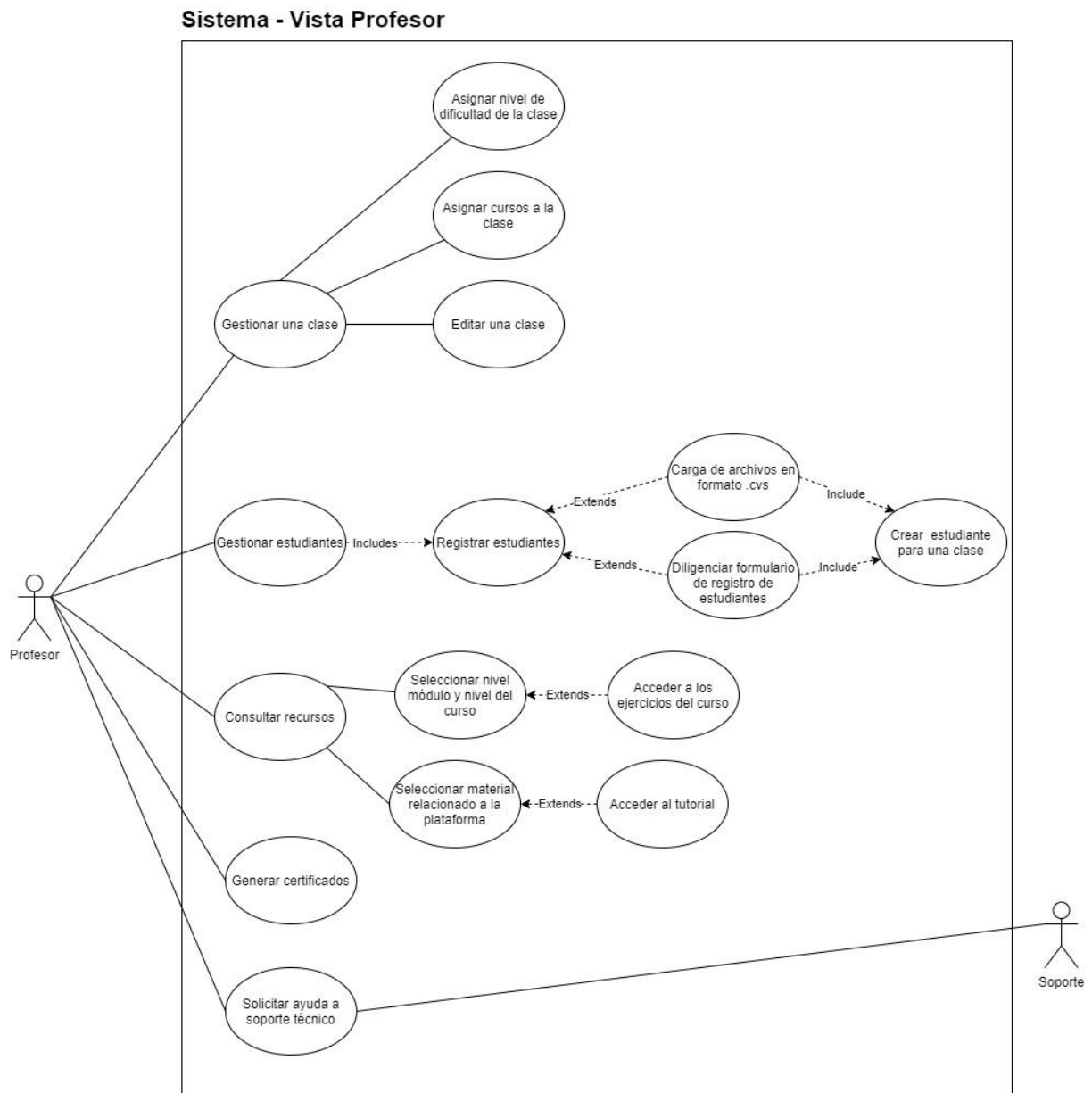
Para llevar a cabo el desarrollo del prototipo de la herramienta, inicialmente se identificaron las necesidades con las que cuentan los usuarios (niños y jóvenes). A partir de los elementos encontrados, se definieron los requisitos funcionales donde se describe lo que hará el sistema o su comportamiento, y requisitos no funcionales, donde se hace énfasis en el diseño y lineamientos para la implementación.

A continuación se describen los requisitos funcionales y no funcionales utilizados en el desarrollo y pruebas de la herramienta.

## **5.2 Definición de requisitos**

Teniendo en cuenta la fase de investigación previa, se definen un conjunto de requisitos funcionales y no funcionales que indican la manera en se que se comporta la herramienta construida. En los requisitos se definieron las dependencias de estos, reglas para que la funcionalidad de los requisitos se cumpla correctamente.

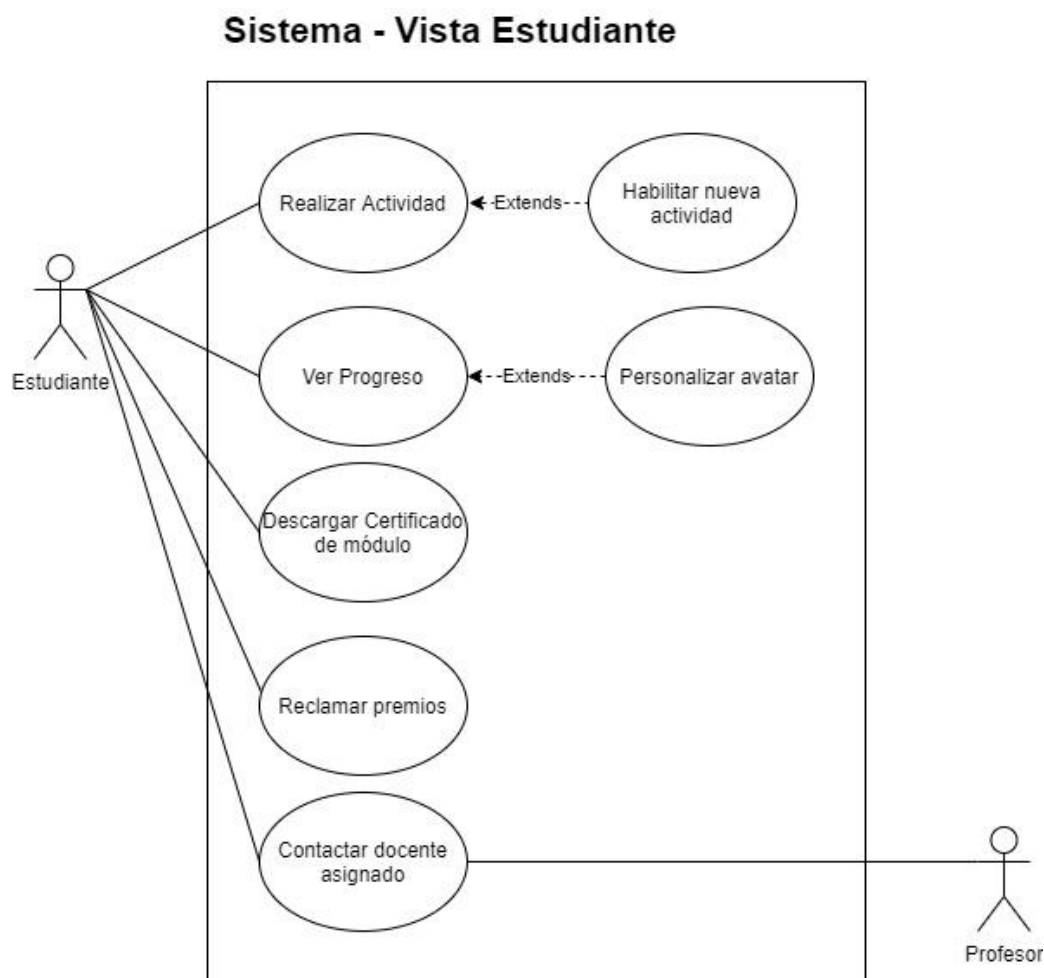
Para llevar a cabo la construcción y validación de los requisitos se hace uso de diagramas de caso de uso, con el fin de dar una visión general del comportamiento que tendrá el usuario al utilizar la herramienta. Los casos de uso utilizados se presentan en las Figuras 7 y 8.



**Figura 7 :** Diagrama de caso de uso desde el rol del profesor. Se muestran las funcionalidades con la que contará el usuario

El usuario con perfil de profesor al ingresar a la herramienta, puede ejecutar múltiples funcionalidades como lo son la gestión de un curso, agregar los temas de clase y su dificultad, agregar estudiantes, generar los certificados para un estudiante al momento en que este finaliza un curso y contactar a un externo encargado del soporte técnico de la herramienta.

Además, el usuario con perfil de profesor debe contar con conocimientos en programación incluyendo conceptos básicos como lo son algoritmia, partes de un computador, estructuras de datos y patrones, hasta programación avanzada donde conozca del lenguaje en el que se basa la herramienta Javascript y Python, patrones de diseño de software y programación orientada a objetos.

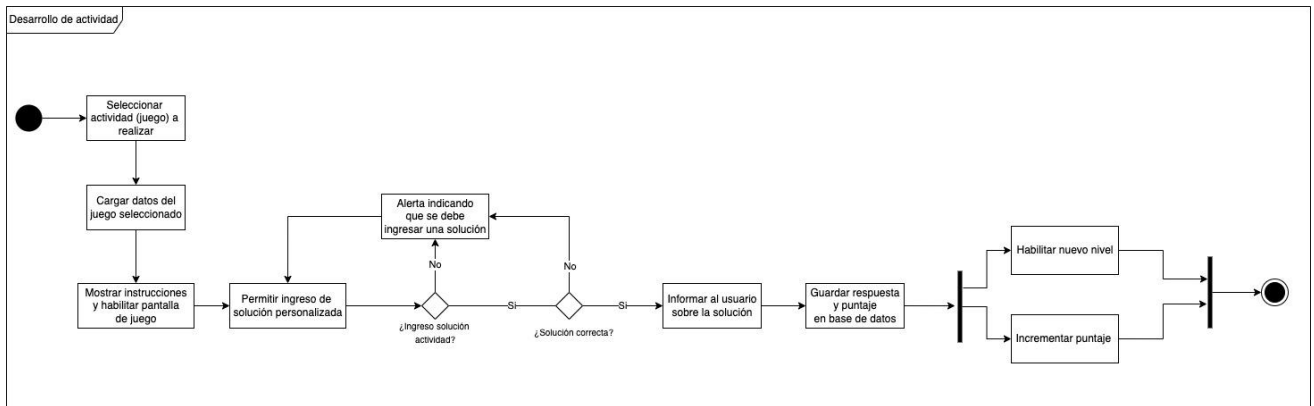


**Figura 8 :** Diagrama de caso de uso desde el rol del estudiante. Se muestran las funcionalidades con la que contará el usuario

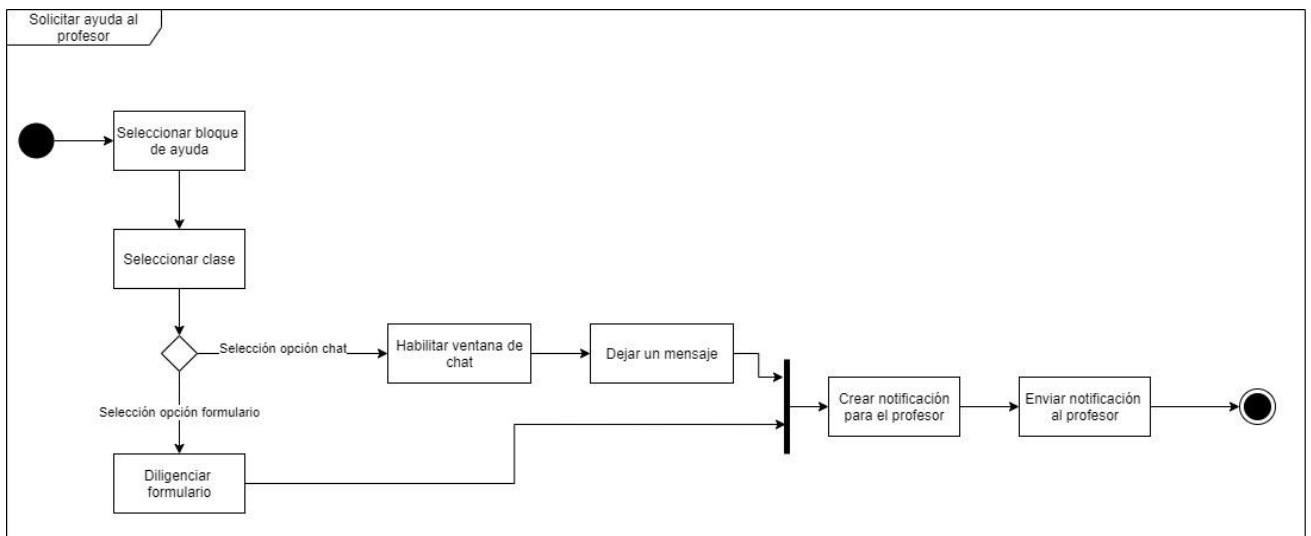
El usuario con rol de estudiante puede acceder a los cursos según el nivel asignado por el profesor, realizar sus actividades, ver su progreso y descargar el certificado al momento de finalizar todas las actividades de un curso. Podrá contactar vía chat o videollamada con el docente del curso y adquirir por medio de puntos otorgados al realizar una actividad ítems para la personalización de su avatar.

Los niños y jóvenes usuarios de la herramienta deberán tener competencias básicas lectoras y deben contar con experiencia previa en el manejo de algún dispositivo como celular o computador. Deben estar en edades entre los 6 a los 16 años, sin embargo la herramienta está abierta a la posibilidad de que el rango de edades de los usuarios pueda ser mayor. Adicionalmente, se establece el flujo principal de la herramienta haciendo uso de diagramas de actividades, los cuales a futuro no solo permitirán validar que los requisitos se cumplen según lo esperado, sino que también facilitará la creación de casos de prueba. Los diagramas

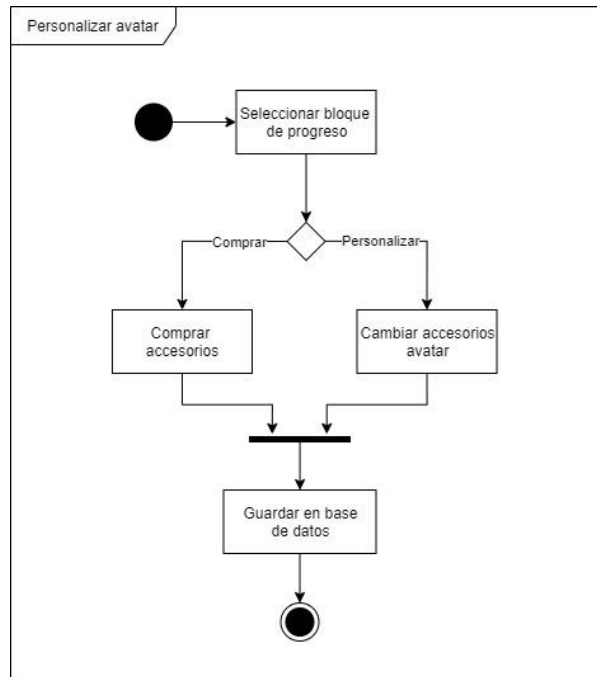
de actividades relacionados al funcionamiento de la plataforma se presentan en las figuras 9, 10 y 11.



**Figura 9:** Diagrama de actividad sobre el desarrollo de los juegos o actividades de la herramienta. Se muestra la interacción en determinados casos y respuesta a las acciones ejecutadas



**Figura 10:** Diagrama de actividad sobre el módulo de ayuda con el que contará la herramienta



**Figura 11:** Diagrama de actividad sobre el módulo de personalización del avatar. Esta sección se utilizará como incentivo para el usuario o estrategia de gamificación

### 5.2.1 Requisitos funcionales

Los requisitos funcionales, describen el paso a paso que contiene la herramienta web para la enseñanza de la programación, para así cubrir las necesidades que se presentan a nivel de ejercicios básicos de lógica computacional, fácil interacción, posibilidad de avanzar de nivel y actividades con código real avanzado.

Los requisitos funcionales se dividen en el portal del profesor y en el portal del estudiante, ya que las funcionalidades deben ser distintas, por lo tanto las pantallas deben verse acorde a lo que se describe.

Cada requisito, cuenta con un esfuerzo asignado teniendo en cuenta la metodología scrum. Este esfuerzo no corresponde a horas, sino a dificultad en su desarrollo. Para la selección del sprint el cual se propone al final de este documento, se priorizan las actividades por prioridad y esfuerzo. Los requisitos funcionales se observan por medio de la tabla 3.

**Tabla 3:** Definición de requisitos funcionales del rol del profesor y estudiante

| <b>Requisitos funcionales para el rol profesor</b> |   |
|--|---|
| <b>RQ-PP-001</b>                                   | La herramienta deberá contener un sistema de login por medio de correo electrónico / usuario y contraseña. La herramienta validará el correo o usuario ingresado. La contraseña deberá contener al menos 8 caracteres donde 1 elemento debe ser un carácter especial y otro una letra mayúscula.  |
| <b>RQ-PP-002</b>                                   | La herramienta deberá contar con la opción de crear una nueva clase, permitiendo el ingreso del nombre de la clase, número de estudiantes, nivel y selección del módulo. La creación de la clase se hará por medio de un formulario.  |
| <b>RQ-PP-003</b>                                   | La herramienta deberá permitir la visualización de las clases creadas por un profesor, mostrando el nombre de la clase, número de estudiantes asignados, nivel y url de acceso. Además cada registro de la tabla donde se mostrarán las clases deberá tener las opciones de añadir estudiantes, edición de una clase, eliminar una clase y visualización de preguntas. La clase solo podrá ser eliminada si no tiene estudiantes asignados o si los estudiantes de una clase no han realizado ningún ejercicio. |
| <b>RQ-PP-004</b>                                   | La herramienta deberá contar con una pantalla de creación de estudiantes, donde por medio de un formulario se solicitará nombre, apellido, nombre de usuario, correo electrónico y contraseña. El estudiante creado será asignado a una clase en particular y solo podrá pertenecer a una clase a la vez. Sin embargo, el estudiante podrá moverse a otra clase si lo autoriza desde el sistema el profesor.  |
| <b>RQ-PP-005</b>                                   | La herramienta deberá contar con una pantalla de creación de estudiantes, donde sea posible cargar un archivo de extensión .csv, el cual contenga la información de   |

|                  |  |
|------------------|--|
|                  | <p>nombre de usuario, nombre, apellido, contraseña, correo electrónico, niveles y módulo al que será asignado. Esto debería crear automáticamente los usuarios en la herramienta.</p>  |
| <b>RQ-PP-006</b> | <p>La herramienta deberá permitir la visualización de los estudiantes de una clase, mostrando el nombre completo, nombre de usuario y correo electrónico. Además deberá contar con las acciones de editar, eliminar y contactar por chat. Un estudiante sólo podrá ser eliminado si no ha realizado ninguna actividad o ejercicio.</p> |
| <b>RQ-PP-007</b> | <p>La herramienta deberá contar con una sección de recursos de apoyo al profesor, donde se explique el uso de la herramienta y se cuente con tutoriales y textos explicativos de los ejercicios por nivel, con el fin de que el profesor pueda estar capacitado.</p>   |
| <b>RQ-PP-008</b> | <p>La herramienta deberá contar con una sección de ayuda y soporte donde el profesor pueda dejar registro de los problemas encontrados, además pueda chatear con una persona de soporte técnico, para agilizar el proceso.</p>   |
| <b>RQ-PP-009</b> | <p>La herramienta deberá permitir al usuario contar con una sección de cursos, simulando un rol de estudiante. Podrá realizar todos los ejercicios, lo cual facilitará la comprensión de los mismos y a su vez de la herramienta.</p>  |
| <b>RQ-PP-010</b> | <p>La herramienta deberá contar con una pantalla de visualización de las preguntas dejadas sobre ejercicios por parte de los estudiantes.</p>  |
| <b>RQ-PP-011</b> | <p>La herramienta deberá contar con un formulario de respuesta, para dar solución a las inquietudes expresadas por parte de los estudiantes.</p>   |
| <b>RQ-PP-012</b> | <p>La herramienta deberá permitir al usuario</p>   |

|  |   |
|--|---|
|  | con rol de profesor contactar a un estudiante a través de chat.   |
| <b>Requisitos funcionales para el rol estudiante</b> |   |
| <b>RQ-PE-001</b>                                     | La herramienta deberá permitir al estudiante loguearse en la plataforma a través del nombre de usuario y contraseña. La contraseña debe ser mínimo de 6 caracteres, contener una mayúscula y al menos un número.  |
| <b>RQ-PE-002</b>                                     | La herramienta deberá permitir al estudiante visualizar a través de una interfaz los módulos asignados por el profesor.   |
| <b>RQ-PE-003</b>                                     | La herramienta deberá permitir desde la pantalla de clases, seleccionar la opción de ayuda que redireccionará a una nueva pantalla con un formulario y chat asociado al módulo seleccionado. Esto con el fin de que el usuario pueda realizar preguntas y resolver dudas de las actividades correspondientes al nivel y módulo. |
| <b>RQ-PE-004</b>                                     | La herramienta deberá mostrar en la pantalla de clases el progreso de cada uno y el puntaje adquirido.  |
| <b>RQ-PE-005</b>                                     | La herramienta deberá mostrar al usuario con rol de estudiante un mapa de selección de las actividades a realizar en modo de juego, habilitando solo un nivel si el realizado previamente fue superado con éxito.   |
| <b>RQ-PE-006</b>                                     | La herramienta deberá contar con un espacio de juego, para que el usuario pueda realizar el nivel habilitado y desbloqueado. La descripción de los juegos del módulo se describen en el documento de diseño de juego.   |
| <b>RQ-PE-007</b>                                     | La herramienta deberá mostrar una ventana flotante de confirmación, en caso de que el estudiante decida cambiar la vista de juego por error, o desee cambiar de   |

|                  |  |
|------------------|--|
|                  | pantalla. Esto se debe realizar con el fin de evitar perder el progreso del nivel. Aplica solo cuando se esté en la pantalla de juego.   |
| <b>RQ-PE-008</b> | La herramienta deberá guardar el avance y progreso del estudiante en cada nivel que realice con éxito. Debido a que las actividades son de lógica y programación, cada estudiante podrá realizar una solución a partir de lo que considere, sin embargo, la herramienta validará a partir de la solución más óptima. Por cada nivel se entregarán puntos. Si se finaliza cada nivel con la lógica y estructura esperada se entregarán todos los puntos del nivel, de lo contrario se disminuirán los puntos según el tipo de solución. |
| <b>RQ-PE-009</b> | La herramienta deberá permitir al estudiante por medio de una pantalla de ayuda contactar con el profesor de cada curso. Para esto contará con una ventana de chat y un formulario, donde el estudiante podrá ingresar una pregunta.   |

### 5.2.2 Requisitos no funcionales

Los requisitos no funcionales que se presentan a continuación por medio de la tabla 4, definen las condiciones de la herramienta, validando que no solo funcione correctamente, sino que además proporcione atributos de satisfacción y calidad al usuario.

**Tabla 4:** Definición de requisitos no funcionales de la herramienta web para la enseñanza de la programación

| Código del requisito | Descripción del requisito  |
|----------------------|--|
| RQ-NF-001            | <b>Usabilidad:</b> El tiempo de aprendizaje del prototipo de la herramienta por parte del usuario (estudiante - profesor) no deberá ser mayor a una hora. La herramienta debe ser lo suficientemente intuitiva con el fin de que el usuario pueda desenvolverse sin dificultades al ejecutar acciones. |
| RQ-NF-002            | <b>Usabilidad:</b> El prototipo de la herramienta deberá contar con mensajes de error claros, los cuales le indiquen al usuario porque   |

|           |  |
|-----------|--|
|           | se ha generado, además de que pasos debería seguir para que no siga apareciendo el mensaje en pantalla.  |
| RQ-NF-003 | <b>Usabilidad:</b> El prototipo de la herramienta deberá contar con adaptación de la web a dispositivos móviles, ya que gran cantidad de usuarios navegan principalmente en internet desde un celular.   |
| RQ-NF-004 | <b>Disponibilidad:</b> El prototipo de la herramienta deberá estar en línea 24/7, ya que el usuario podrá acceder a la plataforma en cualquier momento, desde cualquier lugar, siempre y cuando tenga internet.  |
| RQ-NF-005 | <b>Eficiencia:</b> El prototipo de la herramienta deberá estar en la capacidad de permitir inicialmente al menos 100 usuarios simultáneos, interactuando no sólo con las actividades de juego, sino también en la asignación y creación de clases, o revisión de video tutoriales. |
| RQ-NF-006 | <b>Eficiencia:</b> Los datos creados, actualizados o eliminados del prototipo de la herramienta, deben verse reflejados en menos de 1 segundo en el sistema.   |
| RQ-NF-007 | <b>Seguridad:</b> Los permisos al sistema (servidores), deberán ser otorgados y editados únicamente por el súper administrador.  |
| RQ-NF-008 | <b>Seguridad:</b> El sistema deberá contar con certificado SSL, para que el usuario pueda navegar seguramente en la plataforma web.  |
| RQ-NF-009 | <b>Seguridad:</b> La comunicación entre la aplicación y los servicios debe hacer uso Web Tokens o en su defecto RSA, para evitar la vulneración y consumo de los servicios por parte de externos.  |
| RQ-NF-010 | <b>Mantenibilidad:</b> El prototipo de la herramienta deberá soportar despliegues continuos sin afectar el servicio y siendo imperceptibles para el usuario.   |
| RQ-NF-011 | <b>Confiabilidad:</b> El prototipo de la herramienta debe garantizar al usuario que los datos ingresados son persistentes y el avance en los niveles por parte del estudiante se almacenará correctamente.   |

### 5.2.3 Restricciones de diseño

Teniendo en cuenta que la plataforma a construir será web se deben considerar las siguientes restricciones, antes de establecer los requisitos no funcionales. Las limitaciones de diseño de la plataforma web para la enseñanza de la programación se describen a continuación.

- La plataforma inicialmente se desarrollará para ser ejecutada en un ambiente de navegador web. Sin embargo, la tecnología a considerar debe permitir su visualización y uso en dispositivos diferentes a computadores principalmente celulares.
- Para la etapa inicial de lanzamiento de la plataforma, esta deberá funcionar en navegadores de uso común como lo es Google Chrome.
- Las actividades a desarrollar deben construirse bajo una tecnología soportada por navegadores sin necesidad de instalar plugins, módulos o aplicaciones externas para lograr un correcto funcionamiento.

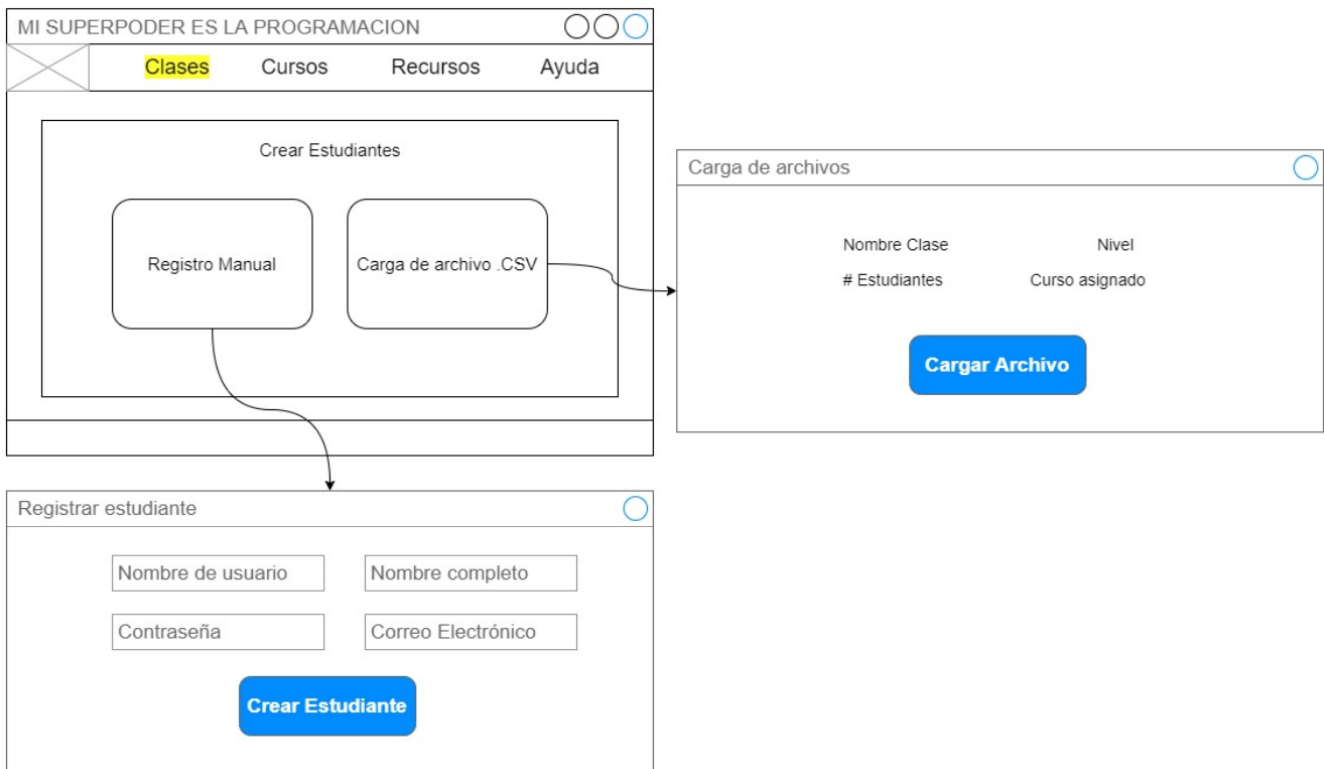
La especificación de los requisitos se encuentra en el [Anexo 1](#) de este proyecto, donde es posible encontrar mayor detalle de cada requisito como esfuerzo, prioridad y dependencia, lo que permitió definir con facilidad los sprints para llevar a cabo el desarrollo de la herramienta. Se debe tener en cuenta que este proyecto se desarrolló haciendo uso de la metodología de Scrum.

### 5.3 Validación de requisitos

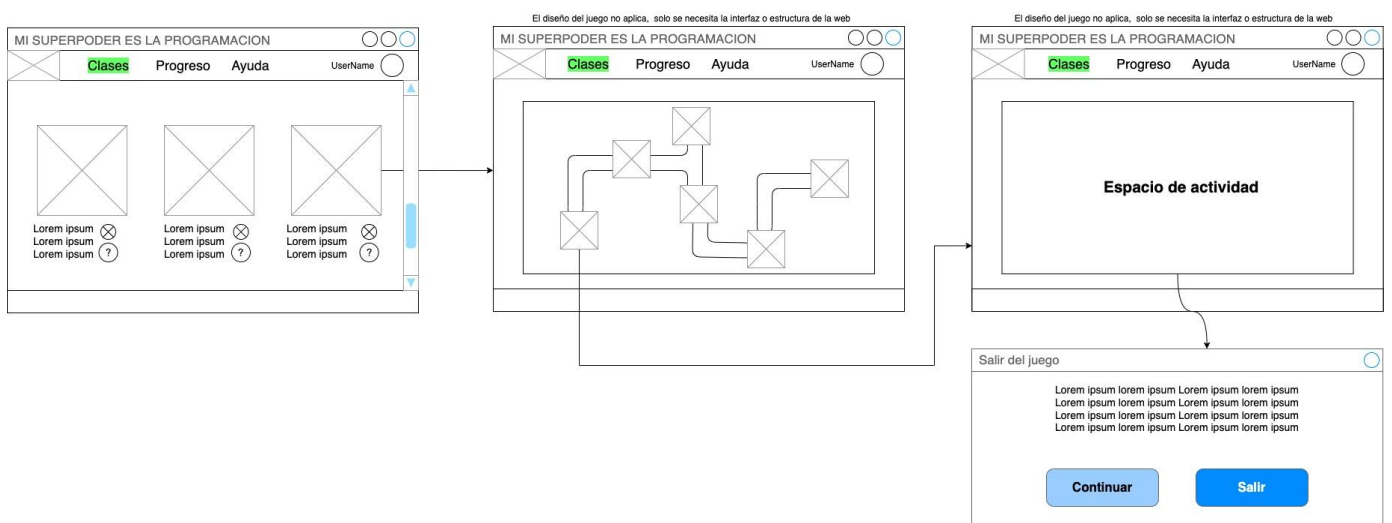
Los requisitos obtenidos y definidos fueron validados con el director del proyecto de grado, para identificar inconvenientes y plantear mejoras en el desarrollo, que permitieran lograr con éxito la construcción de la herramienta web para la enseñanza de la programación propuesta.

Se utilizaron los diagramas de casos de uso y diagramas de actividades presentados al inicio de este capítulo, además de los mockups de la herramienta, con el fin de validar si lo descrito en cada uno de los requisitos, se encuentra plasmado desde lo visual hasta los eventos a ejecutar. Se proponen algunos ajustes, por lo que los mockups son sometidos a tres revisiones adicionales, para garantizar que el flujo e interfaz contienen lo esperado en la definición de requisitos.

Las Interfaces que se muestran en las Figuras 12 y 13, son algunos de los ejemplos. La definición completa de las interfaces se encuentra en el [Anexo 5](#), en el capítulo de interfaces de usuario.



**Figura 12:** Interfaz de usuario tomada desde el rol del profesor, se visualiza la creación de estudiantes para una clase



**Figura 13:** Interfaz de usuario tomada desde el rol del estudiante, se visualizan las clases disponibles o asignadas a un estudiante, mapa de progreso de una clase en particular y espacio de juego

## 5.4 Construcción de sprints para el desarrollo del proyecto

Para llevar a cabo el desarrollo de la herramienta, se establecen la lista de tareas (Backlog) a partir de los requisitos previamente definidos. Se hace uso de la herramienta Trello (Ver Figura 14 y 15), para la gestión del desarrollo ya que este proyecto de grado se realizó bajo la metodología de Scrum.

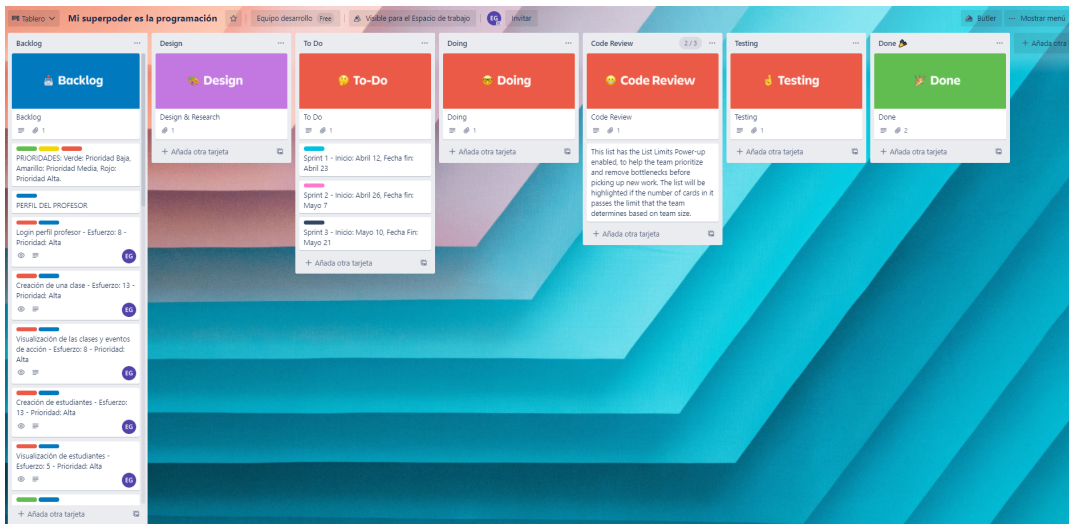


Figura 14: Tablero de trello creado para la gestión del desarrollo de los requisitos del proyecto<sup>11</sup>

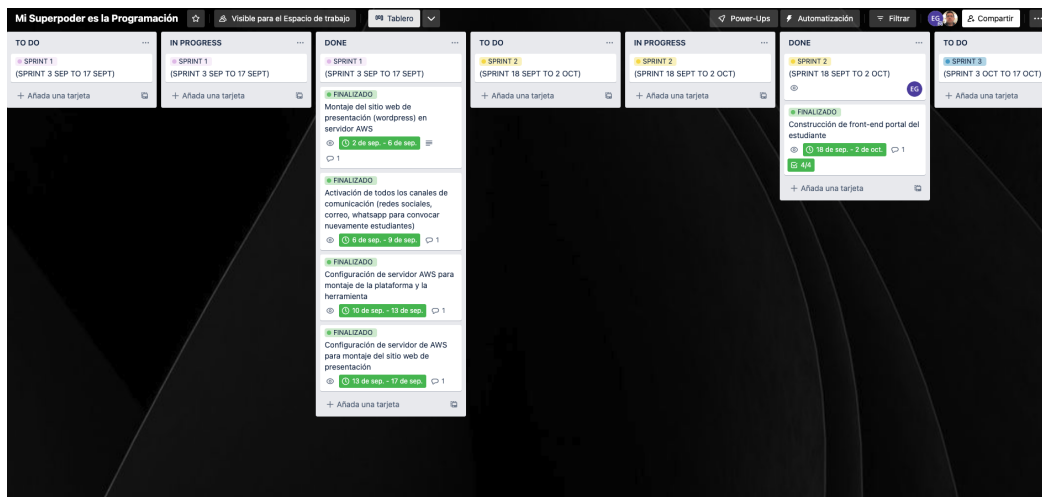


Figura 15: Tablero de trello creado para la gestión de la documentación y seguimiento general del proyecto<sup>12</sup>

<sup>11</sup> Tablero de trello, Mi Superpoder es la Programación:  
<https://trello.com/b/l8tAlsVr/mi-superpoder-es-la-programación>

<sup>12</sup> Tablero de trello, Mi Superpoder es la Programación:  
<https://trello.com/b/3KD5RZ4k/mi-superpoder-es-la-programaci%C3%B3n>

## 6. DISEÑO

### 6.1 Análisis de patrones de arquitectura de software

La arquitectura de un software define la estructura de un sistema ya sea desde etapas tempranas al desarrollo o posteriores según las iteraciones que se realicen, principalmente en proyectos ágiles. Permite comprender cómo funcionará el sistema, o cómo será la comunicación entre sus partes y cuantos elementos físicos (servidores, firewalls, proxys, etc) lo conforman [38].

Para llevar a cabo la construcción de la herramienta web para la enseñanza de la programación en niños y jóvenes, se realizó un análisis de los patrones de arquitectura y estilos de diseño y desarrollo de software más utilizados y conocidos. Esto permitió llevar a cabo la definición de la arquitectura del prototipo de la herramienta por medio de un diagrama, donde fue posible conocer la infraestructura a muy alto nivel.

Previo al desarrollo de la herramienta se llevó a cabo la definición de la arquitectura, la cual se deberá ajustar a diferentes dominios a través del tiempo. La arquitectura establecida puede ser reajustada a medida que en el desarrollo se identifiquen elementos que hagan falta o no sean necesarios y requieran ser eliminados. Los patrones de arquitectura a analizar son los más nombrados y utilizados tanto a nivel académico como de la industria. También se incluyen nuevos patrones de arquitectura enfocados en disminuir cargas y tiempos de comunicación entre el frontend y backend, además de mejorar la curva de aprendizaje en la implementación.

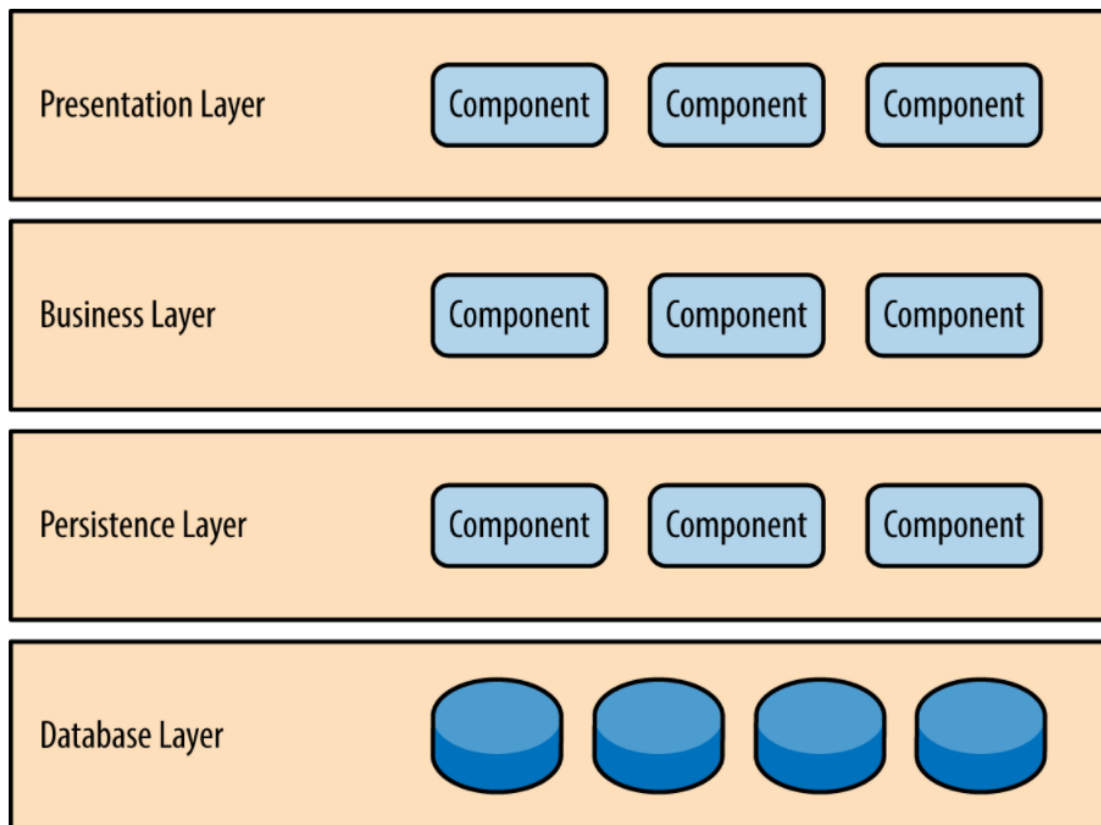
#### 6.1.1 Patrón de arquitectura en capas

Este patrón de arquitectura es uno de los más conocidos y comunes, además de ser un estándar para muchas de las aplicaciones en Java. Se puede utilizar para estructurar un software y dividirlo en grupos de subtareas, donde cada capa puede formar parte de un nivel diferente de abstracción y pueden variar según las necesidades de cada proyecto [39]. Las capas base de las cuales parte esta arquitectura son:

- **Capa de presentación:** Interfaz de usuario
- **Capa de negocio:** Lógica y reglas del negocio
- **Capa de persistencia:** Recuperación de los datos
- **Capa de base de datos:** Almacenamiento de información

Las arquitectura por capas se puede definir de manera distribuida en diferentes ordenadores físicos, donde según la distribución se determina el nivel que posee la arquitectura. Si todas las capas de la arquitectura se encuentran en un solo computador pero con separación por contenedores, significa que la arquitectura es un monolito distribuido en capas, pero si el código convive en el mismo lugar sin ninguna separación significativa, es de un único nivel.

Si la distribución se hace en múltiples ordenadores o contenedores, significa que cuenta con múltiples niveles distribuidos en diferentes máquinas, las capas no conviven necesariamente en el mismo lugar. El número de capas de este patrón de arquitectura lo determina la cantidad de capas creadas para organizar el proyecto.



**Figura 16:** Representación de las capas base de una arquitectura por capas [39]

### 6.1.2 Patrón de arquitectura microservicios

Este patrón de arquitectura se basa en la noción de unidades desplegadas por separado. Los componentes de un sistema al estar desacoplados y ser independientes permiten una implementación más sencilla, mayor escalabilidad y mantenibilidad. Para desarrollar componentes de una aplicación basados en la arquitectura por microservicios, se debe representar a cada componente como único en su funcionalidad [39].

Una arquitectura por microservicios es distribuida, lo que significa que los componentes se comunican entre sí por medio de protocolos de acceso remoto, uno de los más conocidos o utilizados es REST. Los desarrollos que se implementan usando este patrón, buscan la posibilidad de realizar despliegues continuos, disminuir curva de aprendizaje y mejorar la productividad [40] sin afectar todo el sistema. Esto permite que el sistema siga su funcionamiento aunque un componente esté sufriendo alguna falla. Esta arquitectura permite crear componentes enfocados en una única tarea y funcionalidad.

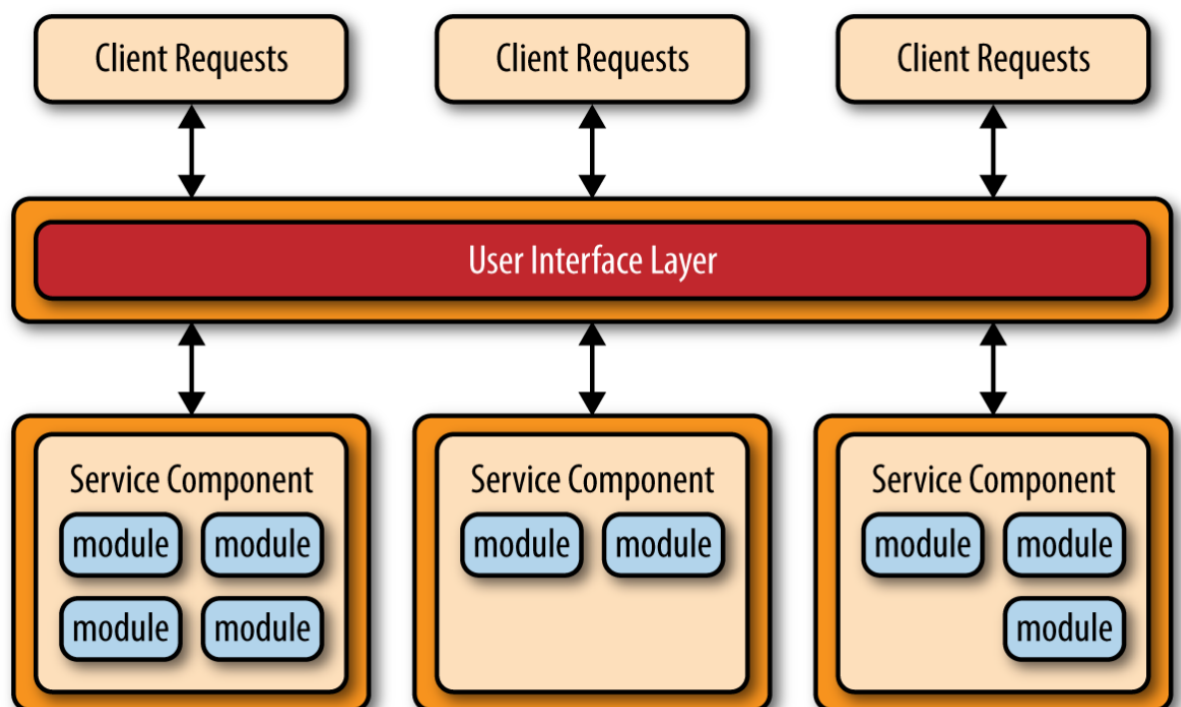


Figura 17: Representación de una arquitectura por microservicios [39]

### 6.1.3 Estilo de arquitectura modelo-vista-controlador

El modelo-vista-controlador es un estilo de arquitectura de software que define la construcción de un sistema haciendo uso de un modelo, una vista y un controlador [38], cada elemento tiene un rol definido y esperado a cumplir. Este estilo de arquitectura nace con el objetivo de reducir el esfuerzo de programación y proporcionar un estándar de desarrollo, donde el código sea más mantenible y legible [41].

Las partes del estilo de arquitectura modelo - vista - controlador, se definen como [41]:

- **Modelo:** Contiene los datos de la aplicación, maneja y controla todas sus transformaciones. El modelo es aislado, no tiene conocimiento de las otras partes que conforman el software.
- **Vista:** Representación visual de los datos de la aplicación. Permite al usuario realizar la interacción y capturar por medio del controlador los datos del modelo.
- **Controlador:** Administra los cambios de estados y es el componente que permite la comunicación entre el modelo y la vista. Captura las órdenes del usuario para modificar los datos del modelo.

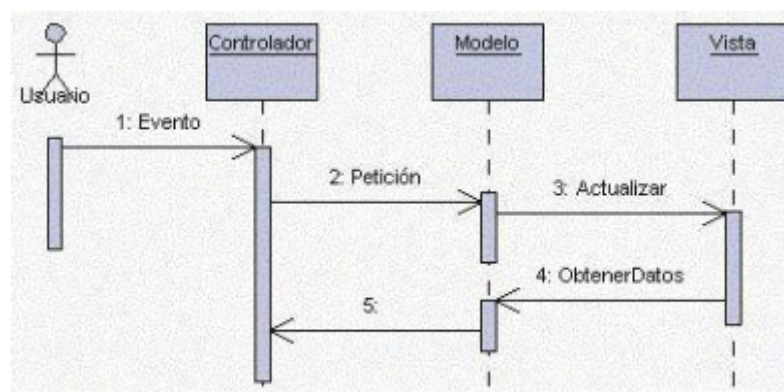


Figura 18: Diagrama de secuencia del modelo vista controlador [41]

#### **6.1.4 Estilo de arquitectura orientado a servicios (SOA)**

El estilo de arquitectura basado en servicios, permite reutilizar elementos por medio de las interfaces que se comunican entre sí haciendo uso de un protocolo basado en HTTP y formatos conocidos para el intercambio de datos como XML o JSON [42]. La comunicación de los elementos, se hace por medio de endpoints o URL donde es posible modificar, añadir, obtener y eliminar los datos, en un lenguaje conocido por la web.

Es posible encontrar REST o SOAP, definidos como estilos de arquitectura, donde las diferencias principales están en los tipos de formatos permitidos, rendimiento y escalabilidad. El estilo de arquitectura REST (Representational State Transfer) accede a los recursos por medio de identificadores uniformes (URI), intercambiando información entre un cliente y un servidor basándose en el protocolo HTTP [43]. Para considerar que una aplicación cumple con el estilo de arquitectura descrito previamente, debe cumplir los principios de cliente-servidor, almacenamiento en caché, sistema en capas y sin estado.

El protocolo SOAP (Simple Object Access Protocol), se enfoca en entornos descentralizados y distribuidos, al igual que REST se basa en el protocolo HTTP para ejecutarse y comunicar los componentes de una aplicación [44]. SOAP nació con el objetivo de comunicar aplicaciones creadas en diferentes lenguajes, sin embargo, al ser protocolo impone reglas, lo que ocasiona tiempos de procesamiento más largos [45].

#### **6.1.5 Patrones de arquitectura sin servidor (*Serverless*)**

Se consideran para la realización de este proyecto los patrones de arquitectura sin servidor ya que optimizan los tiempos de entrega del software, debido a que no se requiere asignar tareas en administración de la infraestructura, demanda, escala y seguridad [46].

Los patrones de arquitectura sin servidor están asociados a software diseñado y almacenado en la nube además del FaaS (Function as Services - Función como servicio)[47], siendo un modelo que se basa en eventos. Generalmente la consulta a datos o información de un sistema debe ser accionado desde un frontend el cual se comunica con nubes como Google Cloud, AWS, Azure, entre otros. El enfoque está en generar valor y escribir código realmente importante.

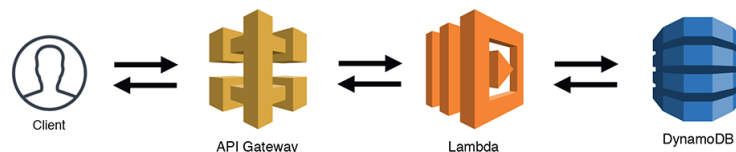
Aunque los patrones de arquitectura sin servidor no son exclusivos de AWS, se hace la descripción sobre este proveedor, ya que es el considerado para este proyecto por su capacidad de adaptabilidad en cualquier tipo de infraestructura (es este caso de sitios web y videojuegos), además de ser la alternativa con menos costos y recursos free tier que son de gran utilidad para este proyecto.

Se evalúan patrones de arquitectura sin servidor para determinar el que dará orden y lineamientos técnicos al proyecto, minimizando costos de los recursos a implementar, hacer reuso, minimizar la actualización de dispositivos y optimización de componentes.

Algunos de los patrones sin servidor considerados para el análisis son:

- **Patrón de microservicios sin servidor, servicio web sencillo**

Este patrón de arquitectura es posiblemente el más utilizado en arquitecturas sin servidor principalmente para AWS. A pesar de ser considerado como el más básico es potente dada la flexibilidad que ofrece. Este patrón cuenta con un evento que a través de una API REST dispara una función Lambda para consultar posteriormente a una base de datos o procesar un algoritmo [48].



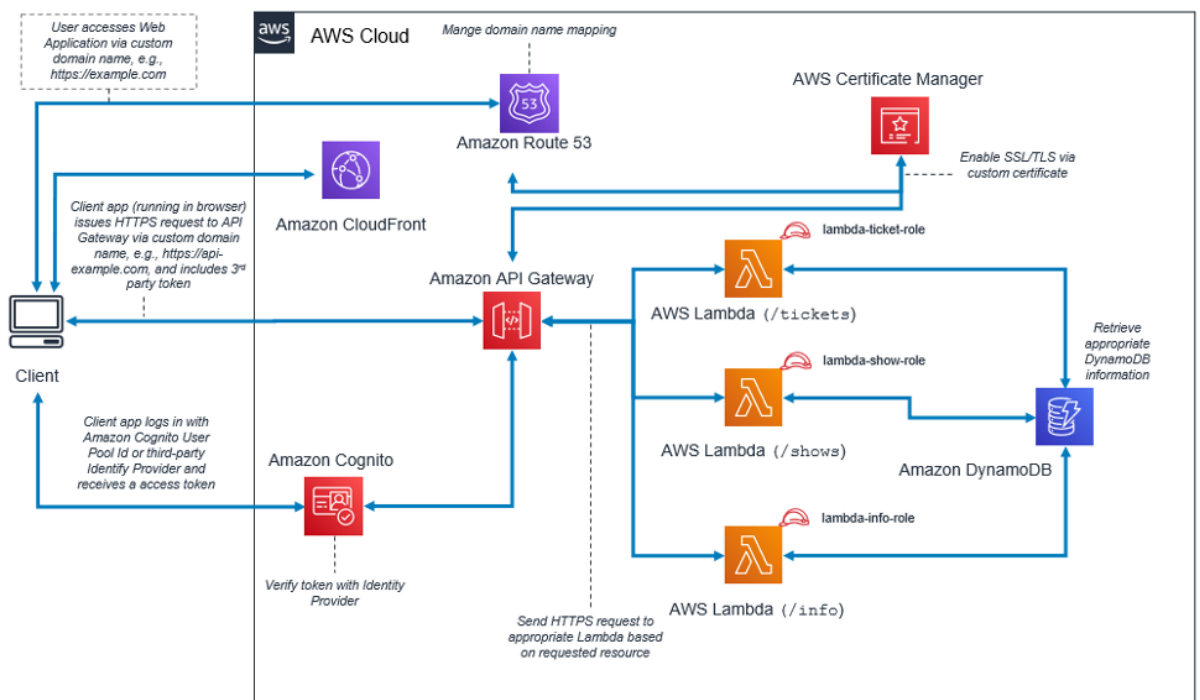
**Figura 19:** Diagrama de arquitectura para el patrón servicio web sencillo en AWS [49]

En AWS el API Gateway es el servicio que permite definir, crear e implementar la API. La interfaz de la aplicación o cliente se conecta a la API enviando solicitudes HTTP las cuales llegan a una lambda, donde se encuentra la lógica y manipulación de los datos de la aplicación. Amazon API Gateway incluye otro servicio de amazon conocido como CloudFront, encargándose de entregar contenido según la ubicación del cliente, también permite combatir escenarios de denegación de servicios (DDoS).

- **Arquitectura de tres niveles - modelo arquitectónico multinivel.**

Esta arquitectura es un modelo frecuente entre aplicaciones orientadas a usuarios, y posee los niveles de presentación, nivel lógico y nivel de datos. En AWS el enfoque no es solo microservicios sino que también es posible encontrar este tipo de arquitecturas. Para mayor detalle sobre la definición de este patrón ir al numeral [6.1.1 Patrón de arquitectura en capas](#), donde se describe previamente.

La diferencia de implementar este patrón sin servidor, es que a nivel lógico no se tendrá que elegir un sistema operativo, proteger la máquina, administrarlo o aplicarle parches. No es necesario preocuparse por la asignación de recursos y escalabilidad; el desempeño no se verá afectado por un aprovisionamiento mínimo [50].



**Figura 20:** Patrón de arquitectura para aplicación web en tres niveles sobre AWS [51]

En la arquitectura de referencia de la Figura 20, el nivel de presentación es todo el contenido estático (web en HTML, CSS, JS). El código de la aplicación web se almacena en el cloudfront y se empieza a ejecutar desde allí descargando todos los recursos al navegador, el cloudfront es un CDN

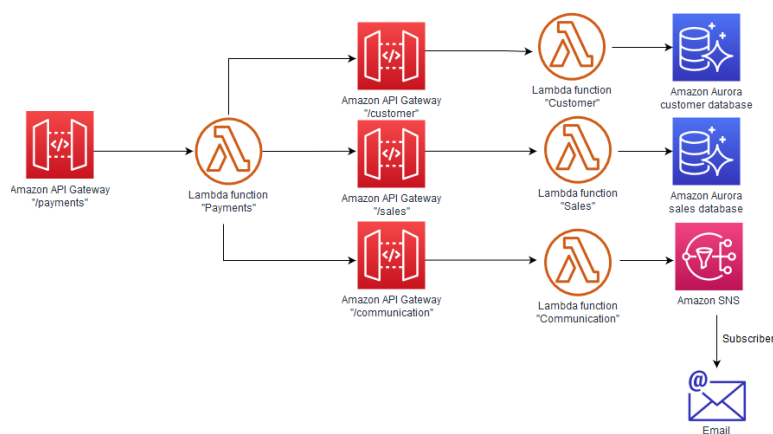
(content delivery network - red de distribución de contenidos), que permite reducir la latencia.

El nivel de lógica está compuesto de Lambdas y API REST usando el servicio de AWS API GATEWAY, cada Lambda está encargada de un aspecto en particular de la aplicación. El nivel de datos cuenta con una base de datos, en el caso de la arquitectura de referencia es DynamoDb pero esta puede ser reemplazada según el caso de uso [51].

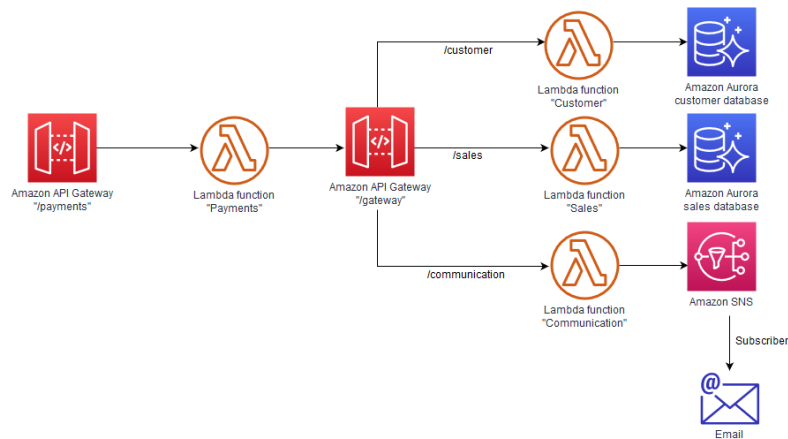
#### - Patrón de puerta de enlace API

Se recomienda el uso de este patrón para diseñar y crear aplicaciones basadas en microservicios. Este patrón enruta solicitudes del cliente a sus puntos finales que corresponden a los microservicios internos, las solicitudes a los microservicios se manejan internamente y nivel cliente se proporciona un único punto final o URL [52].

Se debe considerar este patrón si el número de dependencia de un microservicio no crece con el tiempo, se requiere una llamada síncrona del microservicio, se tiene necesidad de baja latencia y se debe exponer una API para la recopilación de datos de microservicios [52].



**Figura 21:** Patrón de arquitectura puerta de enlace API con múltiples puerta de enlace API, cada microservicio tiene su propia API [53]



**Figura 22:** Patrón de arquitectura puerta de enlace API con API única, microservicios conectados a la misma puerta de enlace [53]

En la figura 21 y 22 se muestran las posibilidades de implementar esta arquitectura usando múltiples API por cada Lambda que representa a un microservicio, o una única API que se conecta con cada una de las Lambdas.

## 6.2 Patrones de arquitectura para videojuegos

Debido a que el objetivo principal de este proyecto fue construir una herramienta para enseñar programación a través de juegos, se consideran patrones propiamente pensados en la construcción de videojuegos, para así tener más claridad en la selección de la arquitectura utilizada.

En el trabajo *“Arquitectura de software para juegos serios con aspectos culturales”* se define una arquitectura conformada por 6 capas como lo son interfaz de usuario, escenario, objetos juego, decoración, aplicación, contexto cultural [54]. Otra definición de arquitectura para videojuegos está dada por el trabajo *“Arquitectura de software para el desarrollo de videojuegos sobre el motor de juego Unity 3D”*, donde se combinan dos patrones de arquitectura como capas y basado en componentes [55].

Cada combinación de patrones o la definición de un nuevo patrón basado en uno ya existente, se construye pensada particularmente en las necesidades del videojuego a construir. Sin embargo, al igual que los patrones y estilos de arquitectura previamente analizados, cada uno se ajusta a las necesidades específicas del videojuego y el objetivo a alcanzar. Estos patrones al igual que los patrones previamente analizados, permiten estructurar cada aplicación, sin perder su propósito original.

### 6.3 Definición de la arquitectura a utilizar en el desarrollo de la herramienta

Para llevar a cabo la definición de la arquitectura a utilizar en la herramienta para la enseñanza de la programación, y del sitio web informativo que se propone en esta etapa del desarrollo como la primera pantalla con la cual el usuario tendrá contacto, se realiza una comparativa entre los patrones y estilos de arquitectura descritos previamente.

**Tabla 5:** Comparación de las arquitecturas, análisis de ventajas y desventajas

| Nombre del patrón                     | Ventajas  | Desventajas  |
|---------------------------------------|---|--|
| Patrón de arquitectura en capas       | <ul style="list-style-type: none"> <li>- Reuso de capas y estandarización de las mismas.</li> <li>- Agrupar los elementos de la aplicación según su responsabilidad.</li> <li>- Cambios enfocados en una capa particular, ya que los demás componentes o elementos están prácticamente desacoplados y se encuentran en las demás capas de la arquitectura, para no generar afectación (Esto no aplica en todos los casos).</li> </ul> | <ul style="list-style-type: none"> <li>- Dificultad para granularizar las capas, lo que puede generar redundancia.</li> <li>- Afectación en varias capas en forma de cascada al realizar cambios.</li> <li>- Una capa depende de la anterior por lo que un problema puede afectar a múltiples capas (Depende del diseño de la arquitectura).</li> <li>- Componentes complejos ya que se deben agrupar según su tipo por lo que podrían ser muy grandes.</li> </ul>                           |
| Patrón de arquitectura microservicios | <ul style="list-style-type: none"> <li>- Despliegues independientes ya que la aplicación está modularizada, lo que evitaría la afectación de otros servicios.</li> <li>- Posibilidades de escalar y optimización de tiempo al enfocarse en desarrollo de componentes pequeños.</li> <li>- Mayor reuso e integración con otras aplicaciones y fácil mantenimiento.</li> </ul>  | <ul style="list-style-type: none"> <li>- Los costos pueden ser más elevados ya que se pueden consumir más recursos debido a que cada micro servicio tiene sus propia base de datos, dependencias y configuraciones, es como si fuera una aplicación independiente.</li> <li>- La arquitectura inicial puede ser más compleja de diseñar, ya que son más elementos a comunicar. Esto puede tomar mucho más tiempo.</li> <li>- Las pruebas pueden ser complejas de realizar a nivel</li> </ul> |

|   |  |   |
|---|--|---|
|   |  | <p>global y se debe tener precauciones con la infraestructura, un mal diseño puede generar problemas de latencia y comunicación entre componentes.</p>  |
| <p>Estilo de arquitectura modelo-vista-controlador</p>    | <ul style="list-style-type: none"> <li>- Claridad en la separación del tipo de lógica de cada parte del código.</li> <li>- Se puede adaptar fácilmente a diferentes frameworks sin importar el lenguaje.</li> <li>- Fácil de escalar y de mantener en equipo sin son proyectos de pequeño o mediano tamaño.</li> <li>- Facilita el manejo de errores.</li> </ul> | <ul style="list-style-type: none"> <li>- Es un estilo de arquitectura que a pesar de ser muy nombrado y reconocido es más difícil de aprender, en comparación con otros modelos, requiere implementar un mayor número de clases e interfaces que en otro patrón o estilo no son necesarios para lograr su aplicación.</li> <li>- Si los programas son grandes, la cantidad de archivos se incrementa significativamente, lo que dificulta el mantenimiento si desde un principio no es clara la división por tipo.</li> <li>- Se enfoca en el paradigma de la programación orientada a objetos, por lo que es una implementación más costosa y puede tomar más tiempo si no se maneja o se tiene experiencia en POO.</li> </ul> |
| <p>Estilo de arquitectura orientado a servicios (SOA)</p> | <ul style="list-style-type: none"> <li>- Favorece la reutilización de servicios y a la escalabilidad de los mismos.</li> <li>- Capacidad de comunicarse con otros sistemas o módulos más fácilmente a través de puntos de acceso (interoperabilidad)</li> <li>- Facilidad en el testeado y la definición de la seguridad de los servicios.</li> </ul>            | <ul style="list-style-type: none"> <li>- Para la definición de este estilo de arquitectura es necesario conocer del negocio para así estandarizar y definir la cantidad de servicios que serán necesarios.</li> <li>- El intercambio de información entre servicios puede ser lenta, todo depende de la correcta</li> </ul>   |

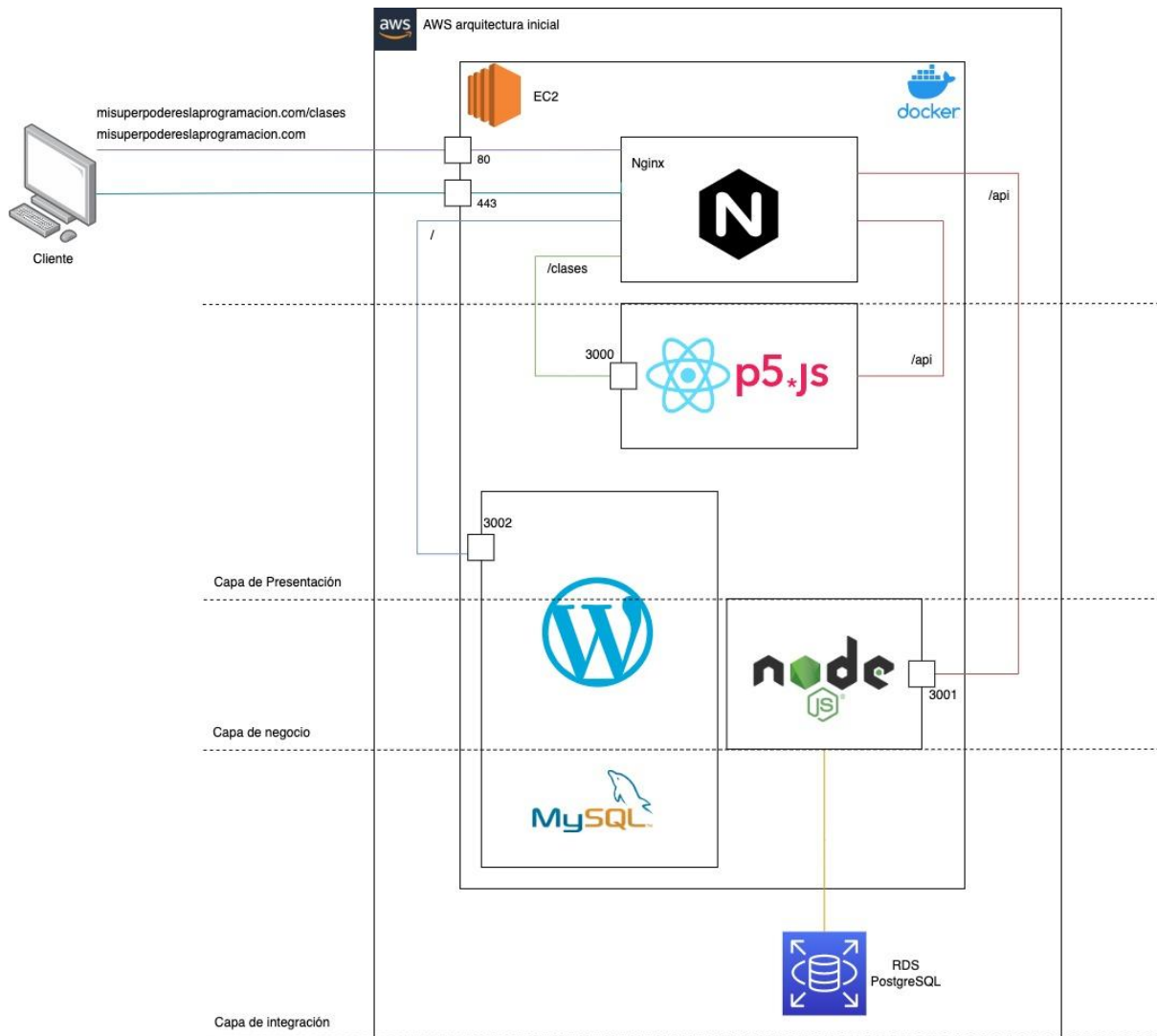
|   |   |   |
|---|---|---|
|   | <ul style="list-style-type: none"> <li>- Menor acoplamiento en comparación con otros estilos o arquitecturas.</li> <li>- Disponibilidad y mayor confiabilidad, permite depurar en pequeñas partes.</li> </ul>   | <p>definición de los elementos y su comunicación.</p> <ul style="list-style-type: none"> <li>- No es recomendada para usarse en aplicaciones con alto nivel de transferencias de datos.</li> </ul>  |
| Patrones de arquitectura <i>Serverless</i> (sin servidor) | <ul style="list-style-type: none"> <li>- Ahorro en el mantenimiento y administración de los servidores, esto en comparación con otras arquitecturas, donde el servidor debe ser configurado y gestionado por el desarrollador encargado.</li> <li>- Facilidad de escalamiento y de integración con otras aplicaciones que utilicen funcionalidades iguales o similares.</li> <li>- Se paga únicamente por lo que se utiliza en el proveedor del servidor para el backend de la aplicación.</li> <li>- Ofrece alta disponibilidad, consistencia y tolerancia de errores.</li> <li>- No se requieren conocimientos especializados en la operación e infraestructura.</li> <li>- Posibilidad de ajustes y adaptación de la aplicación, se enfoca en la mejora continua.</li> </ul> | <ul style="list-style-type: none"> <li>- Complicaciones con la latencia debido a la distribución que manejan este tipo de aplicaciones.</li> <li>- Dependencia total del proveedor de servicios frente a configuraciones de software y hardware de la máquina asignada.</li> <li>- A pesar de que el desarrollo y construcción de una aplicación de este tipo es más fácil, la curva de aprendizaje acerca de la consola del proveedor, requiere tiempo de análisis para lograr las configuraciones correctas de la máquina y la seguridad de la misma. Se debe asignar un tiempo extra para tener comprender lo que el proveedor ofrece y sacar provecho.</li> </ul> |

Se resalta que aunque los patrones de arquitectura SOA y de microservicios se basan en un concepto similar, se diferencian en que SOA es un enfoque para todo un modelo de negocio o empresa, mientras que microservicios es llevado hasta el punto en que solo se usa como estrategia para la implementación entre equipos de desarrollo [56]. En microservicios los servicios son implementados independientemente, mientras que en SOA los servicios aunque están desacoplados conviven en el mismo ecosistema y comparten integraciones, si el ecosistema falla, fallará todo el sistema que cuente con SOA .

Para la construcción de la herramienta y el sitio web de presentación del proyecto, se opta por el patrón de arquitectura sin servidor (*Serverless*): servicio web sencillo [48] con hospedaje de sitio web estático usando buckets de S3. Al inicio del proyecto, la herramienta se construyó sobre un monolito utilizando el patrón de arquitectura por capas. Para ello se usó una instancia de EC2 con cuatro contenedores en Docker distribuidos de la siguiente manera:

- 1) **Capa de presentación:** a esta capa le fue asignado 1 contenedor encargado de mostrar la vista de la herramienta web. Sin embargo en esta capa es posible encontrar parte del contenedor para el sitio estático informativo diseñado en Wordpress. Para este caso Wordpress utiliza su propia base de datos MYSQL, esta se dejó sobre el mismo contenedor. Además, Wordpress al ser un software que requiere una aplicación de servidor y al mandar las peticiones a la base de datos a través de tcp/ip, hace que este contenedor pueda ser encontrado en todas las demás capas.
- 2) **Capa de negocio:** para esta capa fue asignado solo un contenedor, el cual contiene Node JS y se utilizó para definir las interfaces de los servicios, flujos de negocio y conexiones con la capa de datos, de los que hará uso la herramienta web.
- 3) **Capa de datos:** para el prototipo de la herramienta, no se utilizó en esta capa un contenedor, fue utilizado el servicio RDS propio de AWS y se configuró con una base de datos de PostgreSQL. Esta base de datos contiene toda la información de usuarios, módulos y progresos de la herramienta web. En esta capa también se encuentra la base de datos MYSQL que utiliza Wordpress.

El contenedor con Nginx no se incluyó en las capas y se dejó a nivel de cliente ya que solo cumplía a nivel funcional como proxy reverso, encargado de redireccionar las peticiones del usuario al contenedor según la url que este consultara. El diagrama de arquitectura que se construyó y utilizó al inicio del proyecto, se presenta a través de la figura 23.



**Figura 23:** Diseño de arquitectura para la herramienta web y el sitio informativo utilizando docker en una instancia de EC2

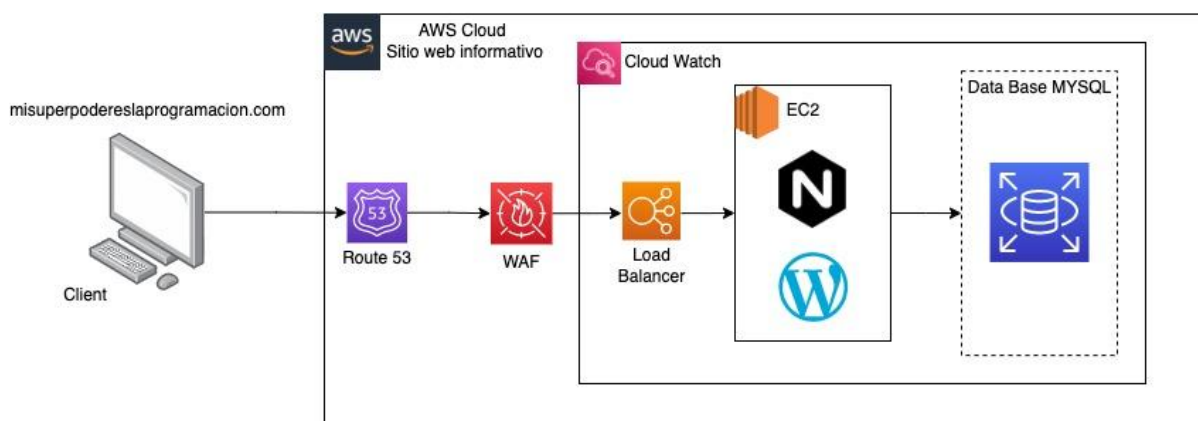
Con la implementación de esta arquitectura se obtuvieron varios problemas, uno de los principales fue el tema de acoplamiento ya que el cobro en AWS se genera por espacio de aprovisionamiento total de la máquina y horas de instancia encendida, a pesar de que se apagaran los contenedores que no estuvieran en uso, el costo seguía siendo el mismo. Al apagar la instancia de EC2 para disminuir costos, no era posible acceder el sitio web informativo. Además, la carga operativa quedaba sobre el administrador del servidor, el cual debía encargarse de actualizar cada una de las aplicaciones instaladas, sistema operativo y base de datos.

La configuración de seguridad correspondía en su totalidad al administrador del servidor, ya que era una máquina personalizada. En este caso se tuvieron problemas con configuraciones de Nginx y de Wordpress lo que dejó una brecha de seguridad y permitió el acceso a

intrusos, los cuales vulneraron el sitio y afectaron su funcionamiento modificando y alterando archivos tanto del sitio informativo como de la herramienta web.

Debido a las problemáticas con esta definición, se decide separar la herramienta y el sitio web informativo por instancias, asignando un EC2 por separado para cada uno (Figura 24), con menos capacidad de almacenamiento ya que no era necesario tener uno de gran capacidad. En esta ocasión se creó la base de datos de MYSQL para Wordpress en un RDS y se colocó en una única instancia Wordpress con Nginx como aplicación de servidor en vez de Apache como se hace normalmente, debido a que Nginx posee un mejor soporte, tiene mayor *performance* gracias a su capacidad de manejar múltiples hilos por su enfoque de eventos asíncronos, mejorando la velocidad y el rendimiento.

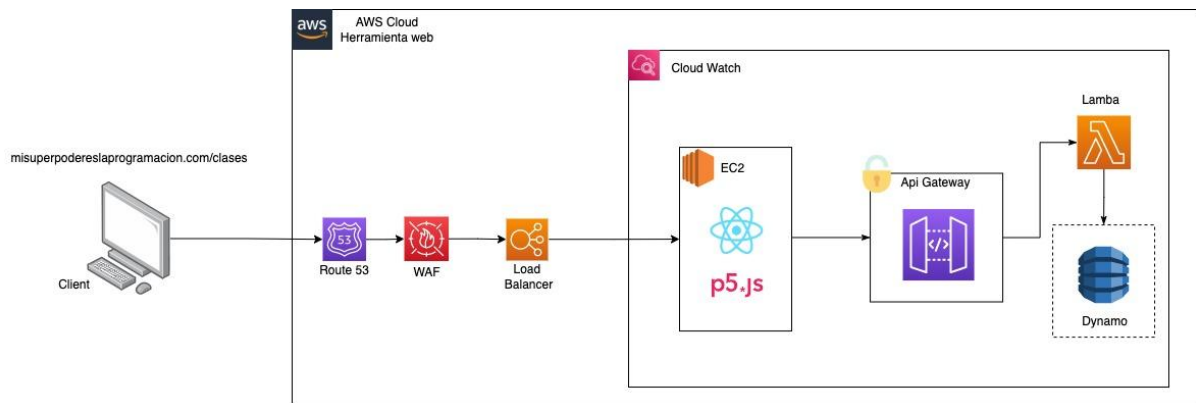
En esta arquitectura aparece por primera vez el balanceador de carga, este servía para redireccionar al usuario a una instancia según al path de la url consultada a través de las reglas configuradas en el balanceador. Se cambia la resolución del DNS usando Route 53, configurando los nombres de servidores de dominio de AWS en Godaddy para que, el encargado de resolver el nombre de dominio sea AWS. El WAF se agregó con la finalidad de configurar reglas de seguridad asociadas al bloqueo de ip con mala reputación o consideradas bots, bloqueo de ips que han bloqueado u ocultado su identidad, bloqueo de patrones asociados a la explotación de vulnerabilidades de PHP, bloqueo del acceso a páginas de administración expuestas, bloqueo de solicitudes asociadas a las vulnerabilidades de sitios construidos en Wordpress y bloqueo a patrones de vulnerabilidad de SQL como los ejecutados en SQL injection.



**Figura 24:** Diseño de arquitectura para sitio web informativo diseñado en WORDPRESS

La herramienta web también se migró a su propia instancia de EC2 con la diferencia de que el backend fue trasladado a una API Gateway con Lambdas, la base de datos se migró a DynamoDB ya que el uso de una base de datos NoSQL facilita modificaciones y cambios significativos en el modelo de datos de la herramienta que está en proceso de desarrollo y

mejoras. Contar con una base de datos de este tipo permite mayor adaptabilidad sin necesidad de hacer grandes cambios, el tráfico es sobre demanda en cada una de las tablas, siendo fácilmente escalable ya que lo hace de manera automática. Además cuenta con una fácil integración con las Lambdas de AWS, realizando cambios en las tablas a partir de un desencadenador de eventos, el cual reacciona a cualquier acción especificada (Figura 25). Route 53 es la misma configuración en todos los casos.



**Figura 25:** Diseño de arquitectura para la herramienta web utilizando API Gateway y Lambdas

Para el caso de la nueva arquitectura del sitio web creado en Wordpress se contaba con problemas de seguridad a nivel de sitio a pesar de contar con el WAF, ya que algunos plugins estaban en su versión gratuita y no podían ser adquiridos por sus elevados precios en las licencias. Esto dejaría puertas de entradas a los intrusos nuevamente. En ambas arquitecturas (Figuras 24,25) los costos de facturación aumentaron de manera significativa, debido a las bases de datos y en mayor medida al balanceador de carga que presentaba problemas en su configuración afectando levantar el sitio en caso de que la instancia con Wordpress fuera apagada.

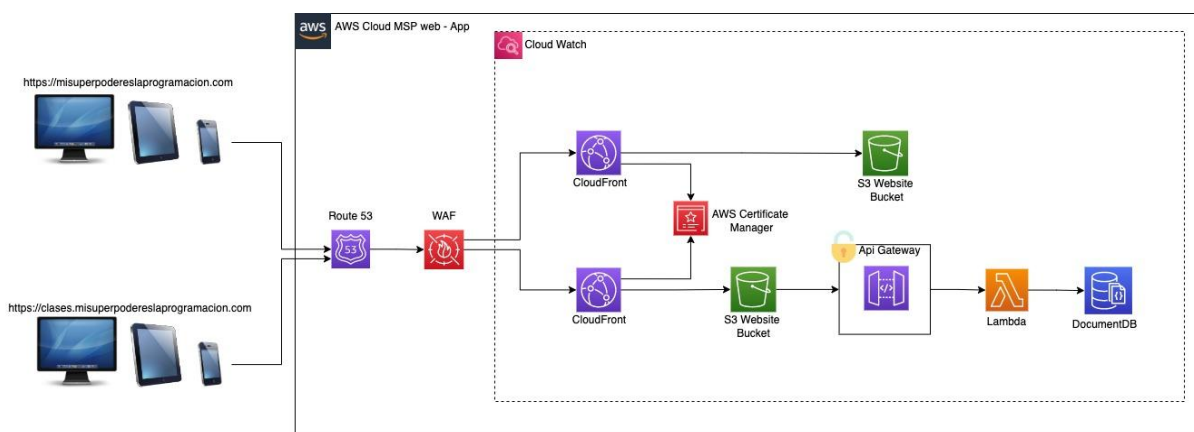
La configuración de Wordpress ocasionaba una redirección 301 , el cual estaba afectando las validaciones necesarias del balanceador de carga para comprobar el estado saludable determinado en los targets group (conjunto de servidores registrados que responde a reglas configuradas, para validar el estado de la respuesta de una aplicación).

El problema con el balanceador de carga se presentó cuando los target group identificaban estado no saludable a pesar de que la instancia reiniciada estaba arriba correctamente. El balanceador de carga al ir al servidor de Wordpress arrojaba 301, lo cual era una redirección propia de Wordpress. Para solucionar este problema temporalmente, se puso el validador (health check) del balanceador apuntando a otro elemento o index dentro del servidor que no fuera de Wordpress, esto se configuró desde el Nginx utilizando un index con un html

básico, para así recibir una respuesta correcta o 200. Al tener el health check y el dominio con estado y respuesta correcta, el balanceador se apuntó nuevamente a Wordpress. Como este proceso era largo y demorado y podría afectar la calidad del sitio, además que se afectaría la correctitud del flujo cada vez que se reiniciara la instancia, se decidió pasar el sitio informativo de Wordpress a una landing page construida en React. Esto evita la dependencia a módulos de Wordpress, mejorando el *performance* del sitio en la evaluación de Google Speed Test.

Se consideró antes de la migración de tecnología utilizar el servicio de AWS LightSails para configurar Wordpress, pero se tuvo problemas. Se tomó la decisión de descartar esta opción, ya que la versión de PHP en la que estaba desarrollado el sitio no era compatible y al momento de realizar la actualización de versión del PHP del sitio web informativo, las plantillas y algunos plugins dejaron de funcionar correctamente generando errores en el sitio.

Finalmente, la arquitectura de la herramienta y el sitio informativo se determinó sobre *Serverless* y se presenta en la Figura 26. Aquí es posible observar que se utilizan Buckets de s3 para almacenar el build (compilado de la aplicación) de la landing page y del prototipo de la herramienta. Se agrega un cloudfront (CDN) para almacenar caché de la página, con el fin de no tener que ir hasta el origen del sitio almacenado sino a puntos más cercanos disminuyendo la latencia del sitio, esto evitaría que la petición para ser completada deba ir hasta la región AWS. Se agrega a la arquitectura AWS certificate manager para administrar los certificados SSL del DNS, esto permite eliminar el proceso manual de renovación y carga de los certificados.



**Figura 26:** Diseño de arquitectura final para el sitio web estático e informativo y la herramienta para la enseñanza de la programación

La base de datos se reemplazó por DocumentDB ya que las consultas sobre DynamoDB podían volverse complejas a nivel de operaciones de escaneo, siendo difíciles de escalar,

mientras que DocumentDB se ajustaba más a las necesidades del proyecto, debido a que las posibilidades de creación de consultas pueden ser más simples, su lenguaje de consulta es más enriquecido y la creación de documentos no requiere la creación de una estructura inicial. DynamoDB no posee un optimizador de consulta por lo que se debe utilizar un índice secundario para hacer consultas o escanear. Adicionalmente, DynamoDB requiere más claridad en la información que se desea almacenar, podría agregar complejidad y ralentizar las consultas si no se tiene un buen diseño.

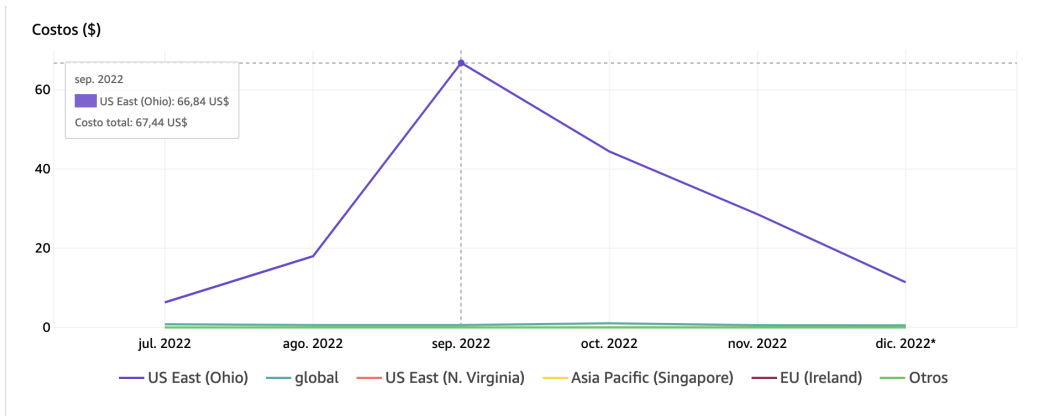
Se posee una experiencia previa en MongoDB por lo que la curva de aprendizaje es menor y la base de datos podría quedar con mejores características que al construirla con DynamoDB donde los conceptos de partición y rango son complejos de comprender y afectan significativamente a la escalabilidad y rendimiento si no son implementados correctamente.

Las instancias de DocumentDB pueden ser más costosas que las de DynamoDB donde se factura por aprovisionamiento y uso por mes, ya que en DocumentDB se factura por hora permitiendo ajustar los costos según las necesidades. Para este proyecto se seleccionó una instancia pequeña de DocumentDB, para no generar gran impacto en los costos, siendo esta considerada como la mejor opción ya que garantiza la disponibilidad de los datos y se presenta una reducción significativa en la curva de aprendizaje al realizar la integración con servicios de AWS.

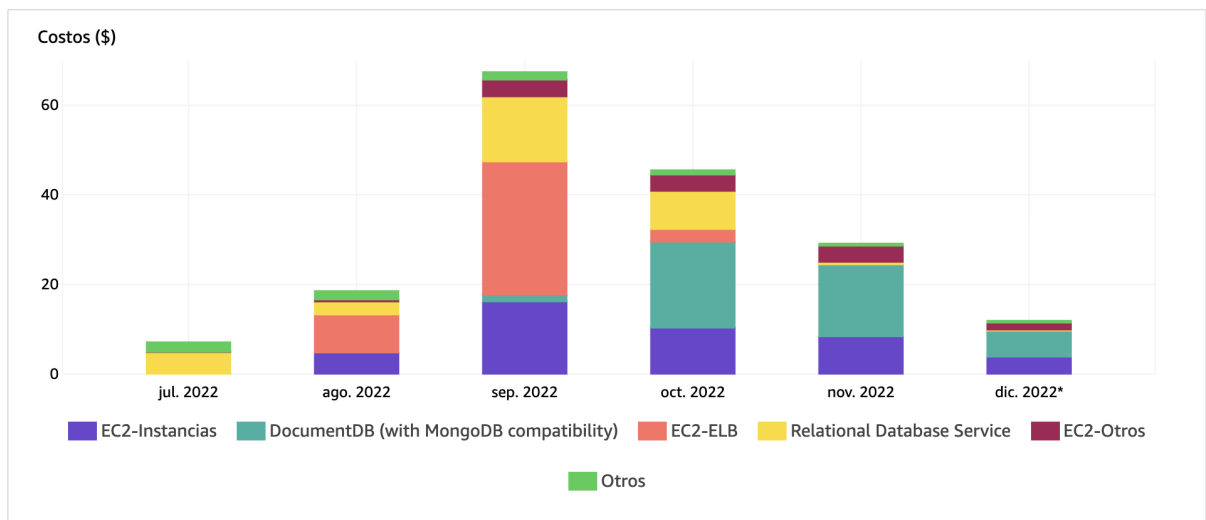
La arquitectura fue validada rigurosamente a través de la asesoría recibida de arquitectos altamente capacitados en la industria del software y que además cuentan con experiencia en la construcción de sistemas basados en *Serverless*. También se consultó a expertos en base de datos sobre la correctitud del modelo de datos, lo cual fue fundamental para asegurar la escalabilidad, rendimiento, confiabilidad y mantenibilidad de la herramienta.

Los ajustes y selección de la arquitectura para este proyecto está basada en las necesidades, requisitos y caso de uso. Las decisiones de arquitectura frente a los costos pueden observarse en las Figuras 27 y 28, donde el mes con mayor incremento en gastos es septiembre, para esta fecha se contaba con la implementación del balanceador de carga (EC2 - ELB), el costo pasó de 18.57 dólares en agosto a 67.44 dólares en septiembre. Se resalta que las pruebas funcionales de la herramienta se realizaron finalizando octubre, para la identificación de errores y aplicar los cambios. Las pruebas con los niños se llevaron a cabo en el mes de noviembre.

Para noviembre y diciembre se ve un ajuste en los gastos, para esta fecha ya se contaba con la implementación de la nueva arquitectura y la que sería declarada a utilizar finalmente para el desarrollo de este proyecto.



**Figura 27:** Costo mensual de los servicios utilizados por región (Ohio) en AWS



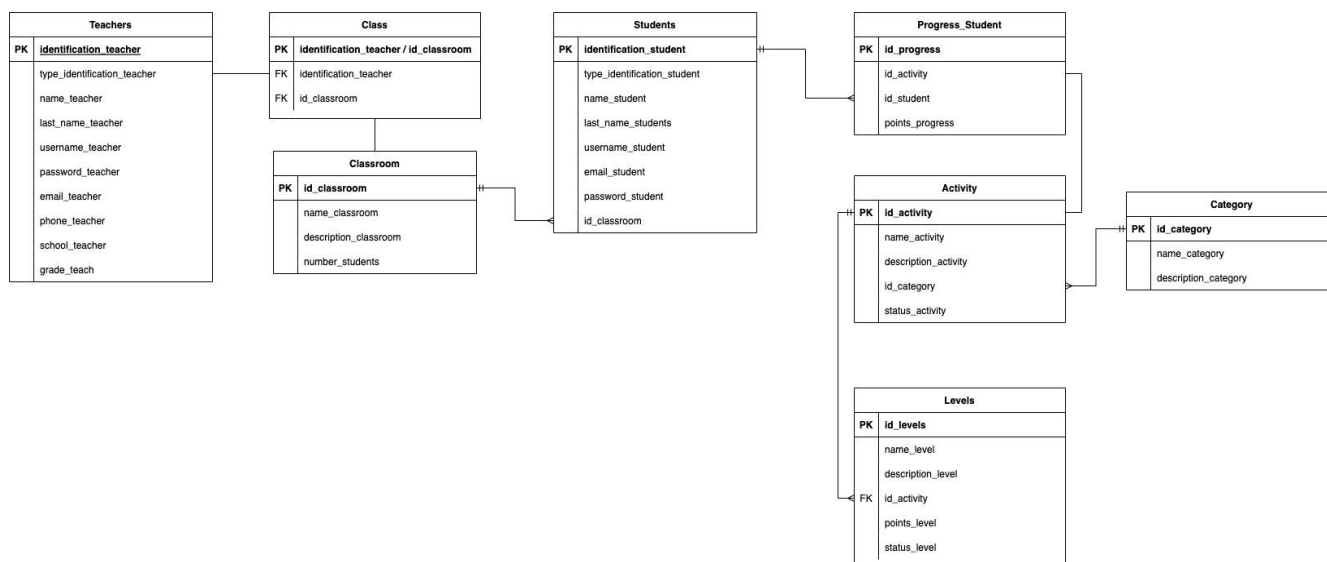
**Figura 28:** Gráfica de costos mensuales generados por los servicios utilizados mensualmente en AWS

#### 6.4 Diseño de base de datos

Al igual que la arquitectura, la base de datos de este proyecto pasó por diferentes esquemas y modelos, a partir de las necesidades presentadas, se realizaron adaptaciones que permitieron mejorar la base de datos para ajustarse a los casos de uso y necesidades técnicas.

Inicialmente el modelo de datos se construyó sobre una definición de modelo relacional, donde las entidades cuentan con llave primaria y foránea para ser relacionadas entre sí, además de la cardinalidad de 1 a muchos. En la Figura 29 se muestra la propuesta inicial, la cual contaba con 7 entidades como:

- **Teachers:** Entidad encargada de almacenar la información de los usuarios tipo profesor.
- **Students:** Entidad encargada de almacenar la información de los usuarios de tipo estudiante. Relacionada con **Classroom**.
- **Class:** Entidad encargada de almacenar la información de las clases creadas por un profesor, relacionada a la entidad **Teachers**.
- **Classroom:** Entidad encargada de almacenar la información de nombre, descripción y número de estudiantes de una clase.
- **Progress\_Student:** Entidad encargada de almacenar la información del estudiante relacionada a su progreso como puntos adquiridos por nivel. Se relaciona con las entidades Students y Activity, la información se almacena al momento de realizar y desbloquear un ejercicio.
- **Activity:** Entidad encargada de almacenar los módulos que serían asignados al estudiante.
- **Levels:** Entidad encargada de almacenar la información de los niveles que conforman un módulo o actividad. Se relaciona con la entidad **Activity**.
- **Category:** Entidad encargada de guardar información que será utilizada para asignar el tipo de categoría a un módulo o actividad como por ejemplo, principiante, intermedio, avanzado. Se relaciona con la entidad **Activity**.



**Figura 29:** Primera propuesta para la base de datos de la herramienta

Debido a que la primera arquitectura fue migrada y se reemplazó el backend construido previamente en NodeJS a un servicio de Lambdas en AWS, también se migró el esquema inicial de base de datos SQL a NoSQL sobre DynamoDB (llave - valor), ya que este tipo de

base de datos funciona muy bien con implementaciones que cuenten con Lambdas, proporcionando ventajas como: mejor aprovechamiento en almacenamiento, costos y tiempo, ya que no se requiere configuración de servidor.

Algunas de las tablas creadas para el esquema relacional, fueron reestructuradas para dar mayor claridad y entendimiento. Se renombran algunos atributos y se ajustan las claves primarias con el fin de que las búsquedas estuvieran mas optimizadas.

| Tabla: Teachers           |  |
|---------------------------|--|
| Clave tabla (Primary Key) | Atributos  |
| id_teacher (string)       | name (string), lastname (string), username (string), password (string), email (string), phone (number) |

| Tabla: Activities         |   |
|---------------------------|---|
| Clave tabla (Primary Key) | Atributos                                     |
| id_activity (string)      | name (string), level (string), levels (array) |

| Tabla: Students           |   |
|---------------------------|---|
| Clave tabla (Primary Key) | Atributos   |
| id_student (string)       | name (string), lastname (string), username (string), password (string), email (string), phone (number), points (number), level (number) |

| Tabla: Progress           |   |
|---------------------------|---|
| Clave tabla (Primary Key) | Atributos   |
| id_progress (string)      | student (string), activity (string), points(number), percent (number) |

| Tabla: Classes            |   |
|---------------------------|---|
| Clave tabla (Primary Key) | Atributos   |
| id_class (string)         | name (string), level (string), teacher (string), activities (array), students (array) |

| Tabla: Questions          |  |
|---------------------------|--|
| Clave tabla (Primary Key) | Atributos  |
| id_question (string)      | title (string), description (string), teacher (string), student (string), class (string) |

| Tabla: Resources          |   |
|---------------------------|---|
| Clave tabla (Primary Key) | Atributos   |
| ir_resource (string)      | title (string), description (string), linkvideo (string), type (string) |

**Figura 30:** Esquema de base de datos NoSQL a ser implementado en DynamoDB.

En la Figura 30 se muestra la definición del esquema que está conformado de las siguientes tablas:

- **Teacher:** Tabla encargada de almacenar la información de los usuario tipo profesor.
- **Students:** Tabla encargada de almacenar la información de los usuarios de tipo estudiante
- **Activities:** Tabla encargada de almacenar la información de nombre y descripción de un módulo o actividad, también cuenta con el campo para almacenar en forma de arreglo los niveles que la conforman.

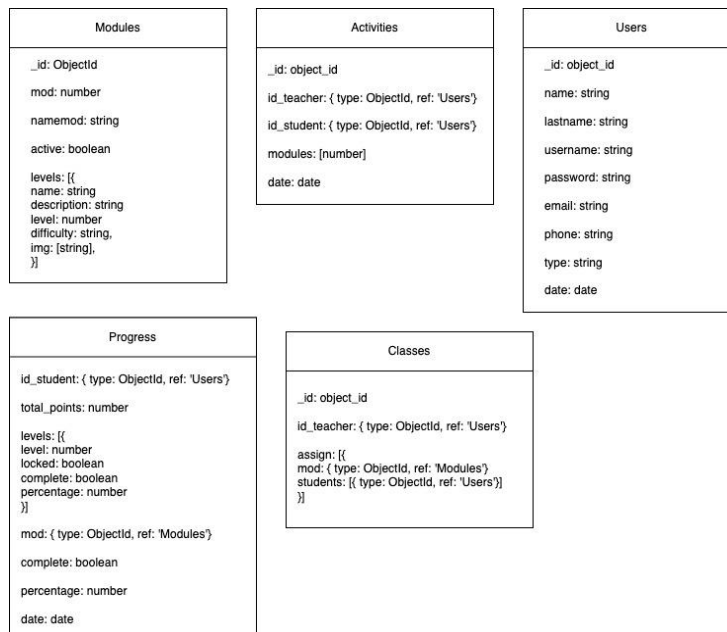
- **Classes:** Tabla encargada de almacenar información relacionada a las clases creadas por un profesor, actividades que han sido asignadas esa clase y los estudiantes que pertenecen a la clase.
- **Resources:** Tabla encargada de almacenar los links de los recursos como imágenes y videos que se mostrarían en la herramienta.
- **Questions:** Tabla encargada de almacenar las preguntas de los estudiantes realizadas a los profesores.
- **Progress:** Tabla que permite almacenar información sobre el progreso de un estudiante al realizar una actividad.

Este modelo a pesar de contar con una llave única e información estructurada, el uso de tablas separadas dificultaba las consultas, además se requería construir más cantidad de servicios para lograr obtener la información esperada, lo que impactaba significativamente en los costos del sistema.

Haciendo uso de llave-valor se incrementaría la dificultad en la escalabilidad al momento en fueran agregados nuevos elementos como tablas o relaciones. Además, se resalta que el prototipo podría tener ajustes futuros, a lo que la base de datos en Dynamo podría ser limitada ya que es menos flexible en cuanto la reestructuración de datos y consultas si se compara con una base de datos de MongoDB.

Con esta base de datos se tenía problemas para conocer el porcentaje de avance del nivel perteneciente a una actividad, se debían realizar operaciones a nivel lógico en las Lambdas para obtener esta información. Al presentar una mayor cantidad de joins entre tablas las consultas no estaban correctamente optimizadas, adicionalmente el manejo de índices y regiones de DynamoDB no se estaba ejecutando de la manera correcta logrando afectar significativamente el rendimiento de la base de datos.

El esquema fue migrado a una base documental, DocumentDB compatible con MongoDB, donde se mejoró significativamente la estructura para optimizar las consultas, se agregaron referencias y a su vez índices de búsqueda. Se decide hacer esta migración ya que una base de datos basada en MongoDB es más flexible y escalable, permite un mayor rendimiento en las consultas, es compatible con lenguajes de programación ampliamente utilizados en la industria, principalmente Javascript, lenguaje seleccionado para la construcción de la herramienta, y la información almacenada en las colecciones (tablas) son de tipo JSON.



**Figura 31:** Esquema de base de datos NoSQL a ser implementado en MongoDB o DocumentDB.

En este nuevo esquema de base de datos está distribuido de la siguiente manera:

- **Modules:** Almacena la información de los módulos que serán asignados a los estudiantes y a su vez se asigna en su interior la información de los niveles para el módulo creado.
- **Activities:** Esta colección se encarga de almacenar la información de los estudiantes, con sus módulos asignados y profesor.
- **Users:** Esta colecciones la unión de las tablas **Teachers** y **Students** de los esquemas anteriores. Los usuarios se identifican a través del campo tipo.
- **Progress:** Esta colección almacena el progreso general de los estudiantes y es discriminado por niveles del módulo. Permite también conocer el estado y la completitud, si los niveles han sido desbloqueados, finalizados o están progreso.

Este esquema diseñado para MongoDB es compatible con DocumentDB de AWS donde finalmente se creó la base de datos de la herramienta. Este diseño tiene datos redundantes con el objetivo de optimizar las consultas y que las Lambdas obtengan los datos rápidamente sin necesidad de consultar en múltiples documentos. Las Lambdas se crearon por tipo y funcionalidad, por lo que cada colección de la base de datos está asociado a una Lambda, mejorando la mantenibilidad del código. Se decide utilizar índices para organizar los datos y agilizar las consultas, esto evitará que sea escaneada toda la colección.

## 6.5 Definición de la tecnología a utilizar

Las tecnologías para el desarrollo web están en constante evolución y crecimiento. Sin embargo, hay algunas que han demostrado ser persistentes y las más utilizadas a través del tiempo. Cada tecnología se describe a continuación, servirá para comprender mejor la selección final de las tecnologías del proyecto.

Para el desarrollo de este proyecto, las tecnologías consideradas y seleccionadas actualmente cuentan con soporte y gran comunidad que hacen aportes significativos en foros que facilitan el diseño, construcción y solución de errores.

- **REACT:** Es una biblioteca de Javascript enfocada en la creación de interfaces de usuario. Es fácil de depurar y renderizar de manera eficiente el componente cuando los datos cambian, es reactiva (reacción a cambios). El frontend puede ser renderizado desde el servidor para potencializar el rendimiento y las aplicaciones.<sup>13</sup>
- **P5:** Es una biblioteca de Javascript enfocada en la programación creativa, busca que programar sea fácil aunque no se tengan muchos conocimientos. Su conjunto de componentes permiten dibujar y crear elementos interactivos.<sup>14</sup>
- **Typescript:** Es un lenguaje de programación basado en Javascript fuertemente tipado lo que permite crear programas más ordenados y limpios. Agrega sintaxis adicional que puede ser integrada fácilmente con proyectos que han sido desarrollados previamente con Javascript. Al ejecutarse se convierte en código Javascript para ser comprendido correctamente por el navegador.<sup>15</sup>
- **Javascript:** Es un lenguaje de programación interpretado, el cual es conocido como el lenguaje Scripting (secuencia de comandos) para la web. Este lenguaje en la actualidad puede ser utilizado del lado del backend por medio de NodeJs. También cuenta con soporte a programación orientada a objetos.<sup>16</sup>
- **DocumentDB - Mongo:** Es una base de datos gestionada completamente por la nube de AWS, la cual es compatible con los comandos y sintaxis de MongoDB. Es escalable y de larga duración, realizando constantes copias de seguridad y replicación automática. Maneja formato JSON, procesa múltiples solicitudes por segundo con lecturas globales a baja latencia.

---

<sup>13</sup> React, biblioteca de Javascript: <https://es.reactjs.org/>

<sup>14</sup> P5, biblioteca de Javascript: <https://p5js.org/es/>

<sup>15</sup> Typescript, lenguaje de programación: <https://www.typescriptlang.org/>

<sup>16</sup> Javascript, lenguaje de programación: <https://developer.mozilla.org/es/docs/Web/JavaScript>

## 6.6 Bocetos, escenarios e interfaces

Para llevar a cabo la construcción de la herramienta se establece el flujo entre pantallas y funcionalidades utilizando mockups e interfaces de alta fidelidad que permiten comprender la manera en que se verá el sitio web y las opciones con las cuales contará el usuario. Para la validación de las pantallas se contó con el apoyo de un diseñador en UX.

También, se diseñan los niveles o juegos que estarán contenidos en la herramienta para enseñar sobre conceptos básicos de la programación en el prototipo a entregar. Se define la historia, fondos, elementos, personajes y gama de colores. Todo lo anterior se detalla en el documento de juego o GDD (Game Design Document), presente en el [Anexo 2](#).

En el GDD se puede encontrar detalle sobre los niveles básicos de la herramienta, los cuales se enfocan en el desarrollo de la lógica computacional por medio de patrones, secuencias y algoritmos, además los niños y jóvenes aprenderán también sobre el funcionamiento de los dispositivos informáticos, como se comunican entre ellos y como procesan y almacenan los datos.

Posterior al aprendizaje de los conceptos básicos, los niños y jóvenes podrán pasar a conocimientos avanzados enfocados en codificación, donde se enseña propiamente un lenguaje y se aprende sobre programación orientada a objetos, programación funcional, sintaxis del código y patrones de diseño.

A continuación se presenta el diseño UX final de las pantallas del portal del estudiante y profesor, que facilitan la implementación y validación de requisitos. Se presentan en este trabajo las pantallas que permiten la definición del producto mínimo viable correspondiente a la interacción con el flujo más importante.

- Pantallas mínimo viable portal del profesor:

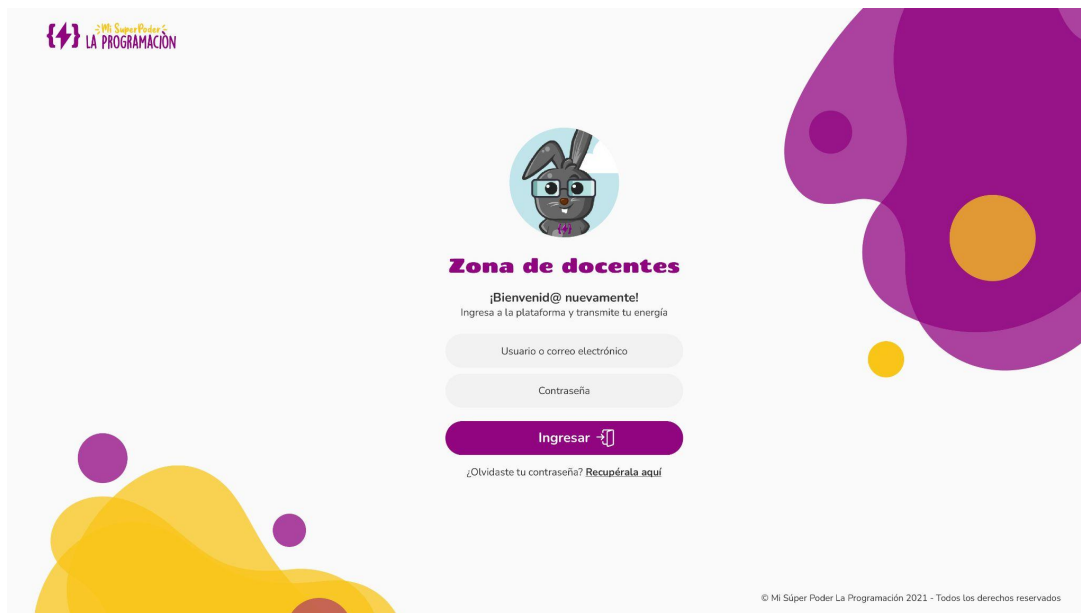


Figura 32: Diseño UX para pantalla de login cuando se ha validado que el usuario es de tipo Docente.

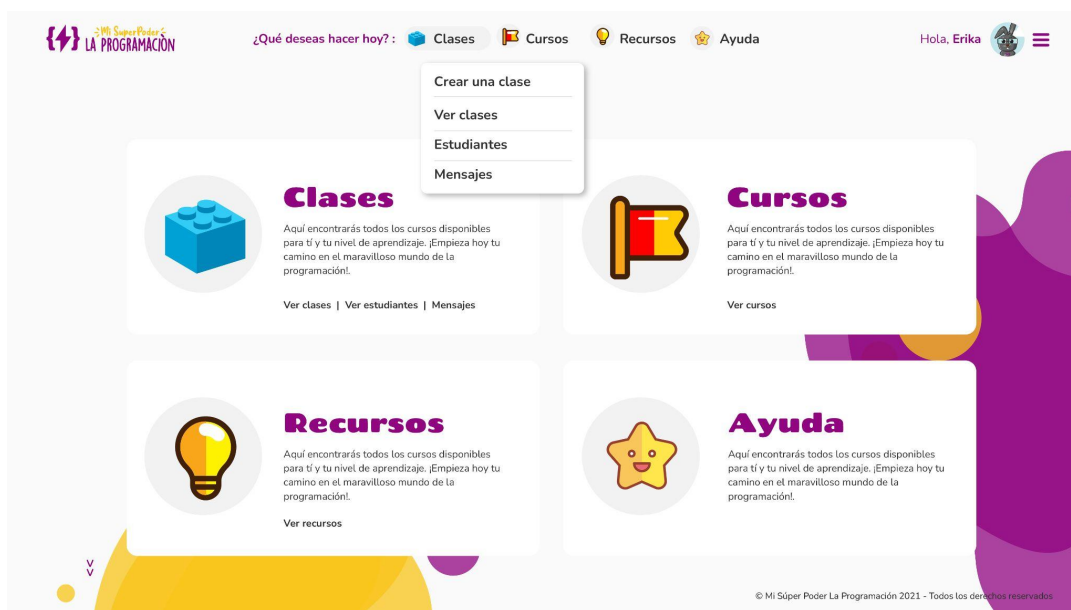
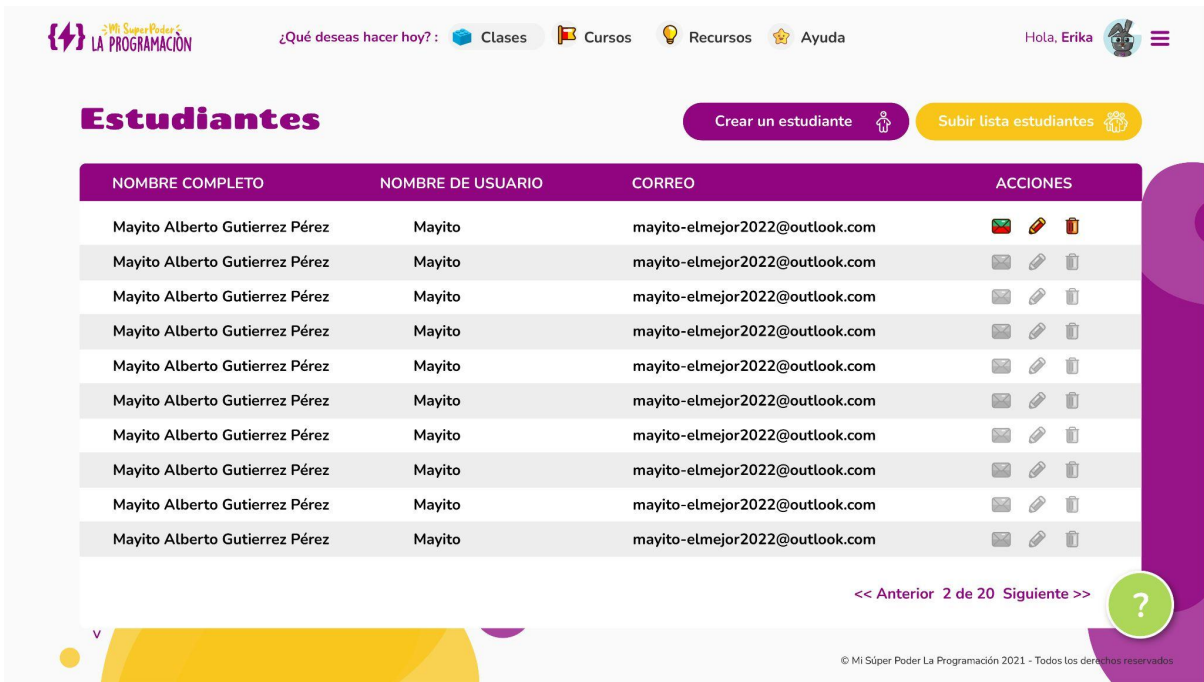
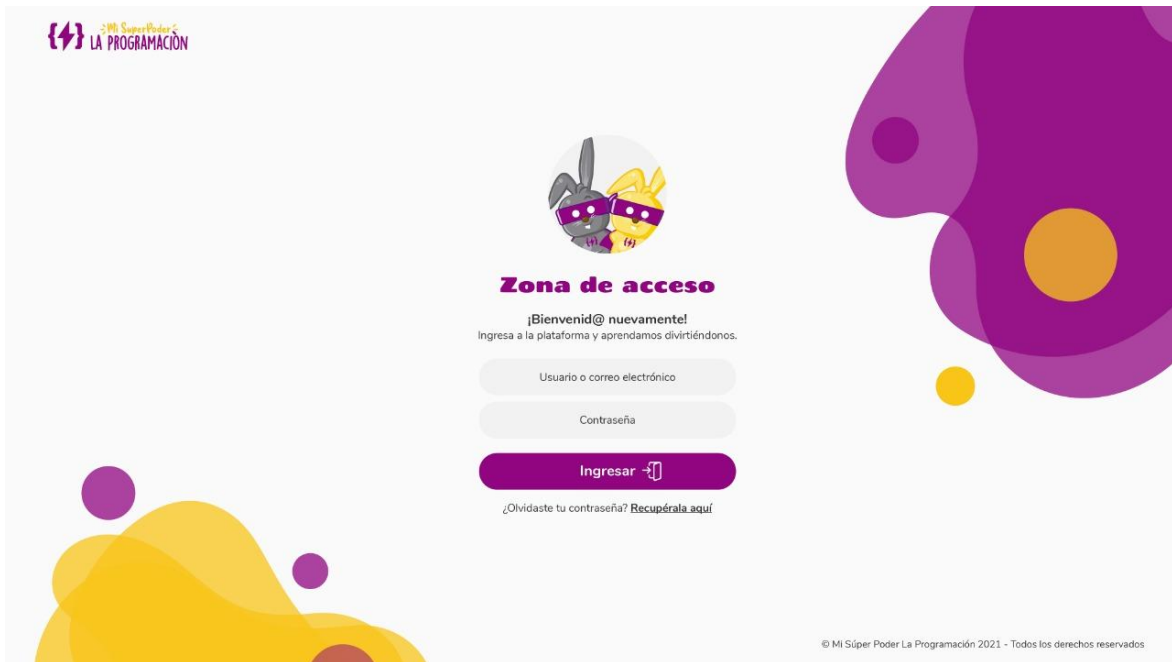


Figura 33: Diseño UX para pantalla de inicio, después del login. Aquí el profesor contará con las acciones que se muestran en pantalla.



**Figura 34:** Diseño UX para pantalla Ver Estudiantes, para visualizar los estudiantes asociados a una clase

- **Pantallas mínimo viable portal del estudiante:**



**Figura 35:** Diseño UX para pantalla de login cuando se ha validado que el usuario es de tipo Estudiante.

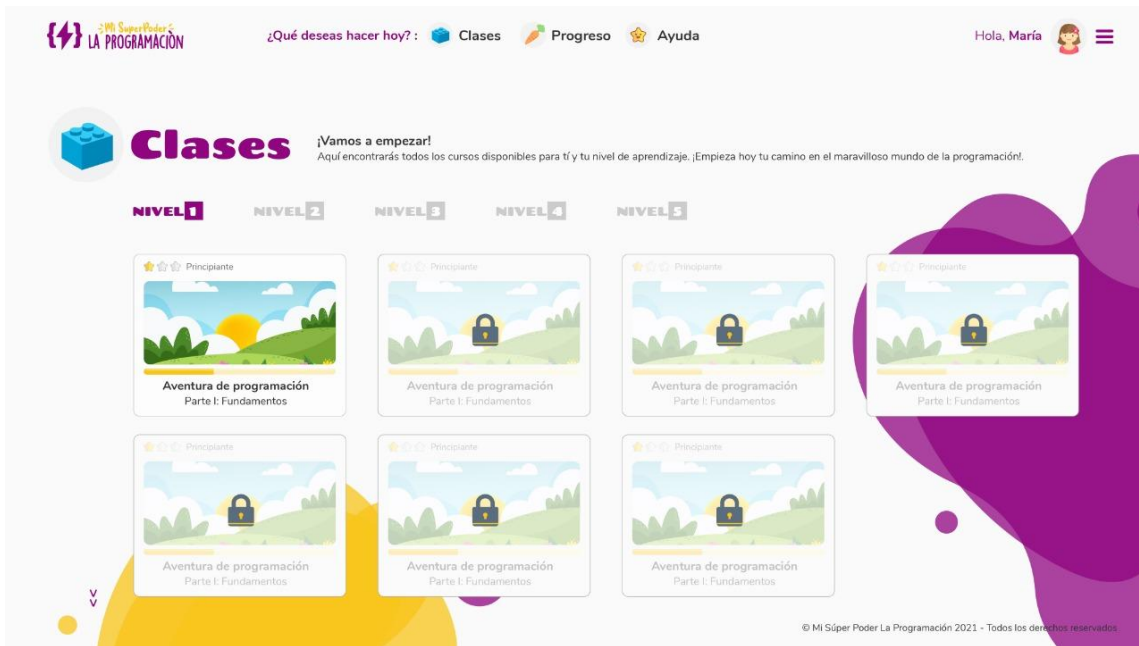


Figura 36: Diseño UX para pantalla con módulos asignados a un estudiante.

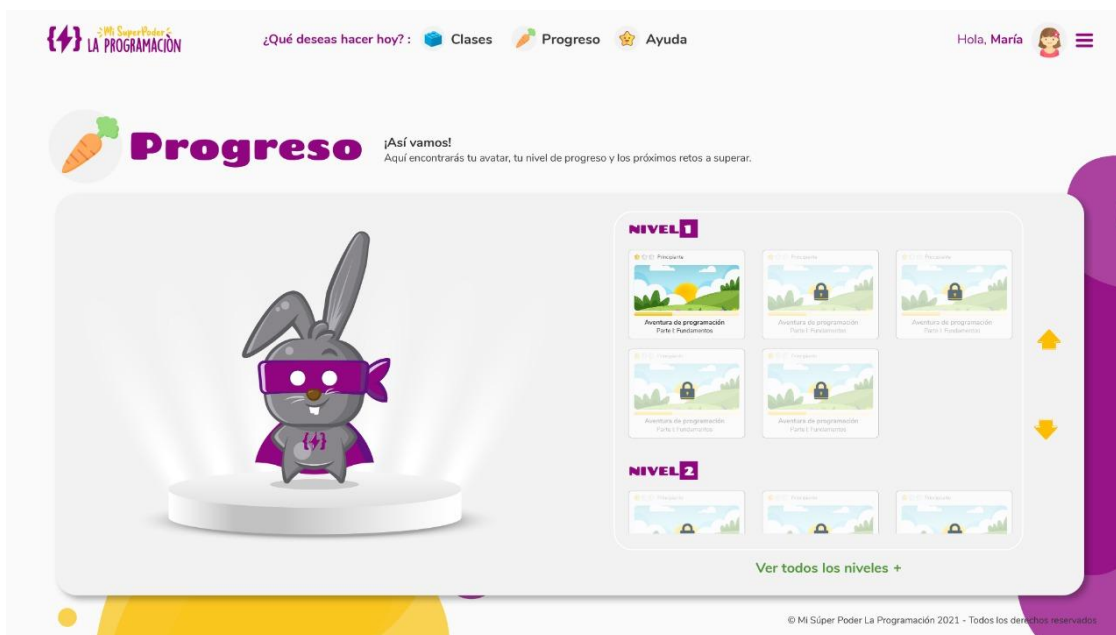


Figura 37: Diseño UX para pantalla de progreso e interacción con el avatar.

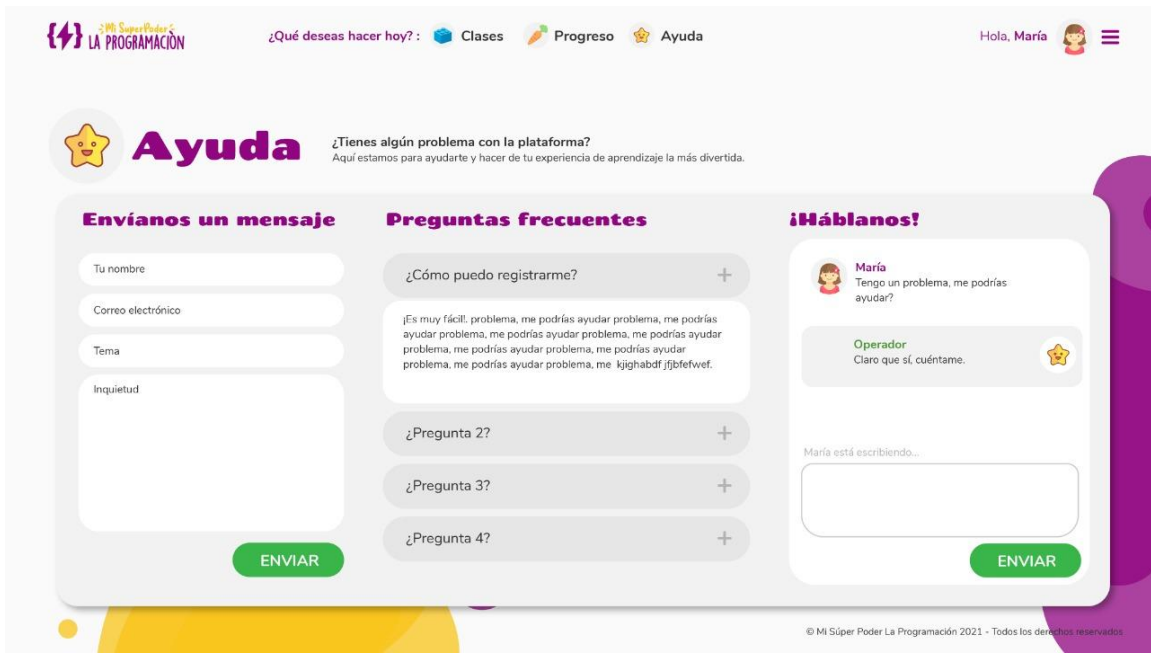


Figura 38: Diseño UX para determinar sección de ayuda que facilitará el contacto directo con el profesor.

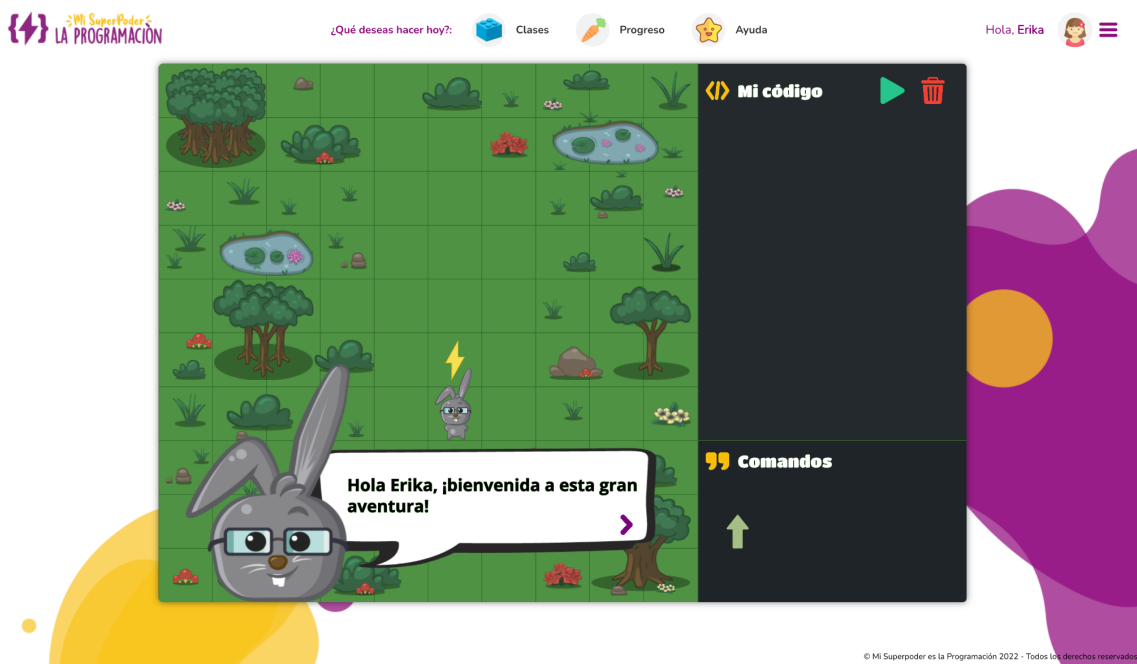


Figura 39: Diseño UX para pantalla de juego, creación de código y ejecución del mismo.

## 7. DESARROLLO

### 7.1 Construcción del prototipo de una herramienta web para la enseñanza de la programación

La construcción de la herramienta atravesó múltiples cambios, la arquitectura inició como un monolito por capas, siendo migrada después de diversos retos técnicos y de seguridad a una tipo de arquitectura *serverless*, permitiendo obtener mejoras en temas de gastos, optimización y rendimiento.

La base de datos inicial se diseñó relacional para ser creada sobre PostgreSQL, se definió de esta manera por la gran comunidad y documentación con la que se cuenta, no se requería curva de aprendizaje por la experiencia previa en el manejo de estas base de datos y por la atomicidad se garantizarían transacciones y datos en correcto estado y no duplicados. Sin embargo y debido a que en la construcción del prototipo se identificó la volatilidad con la que se esperaba construir módulos y nuevas funcionalidades para la herramienta, la escalabilidad que se quiere lograr en el futuro y su descentralización se decidió migrar a una base de datos NoSQL como Document DB de AWS la cual es compatible con MongoDB.

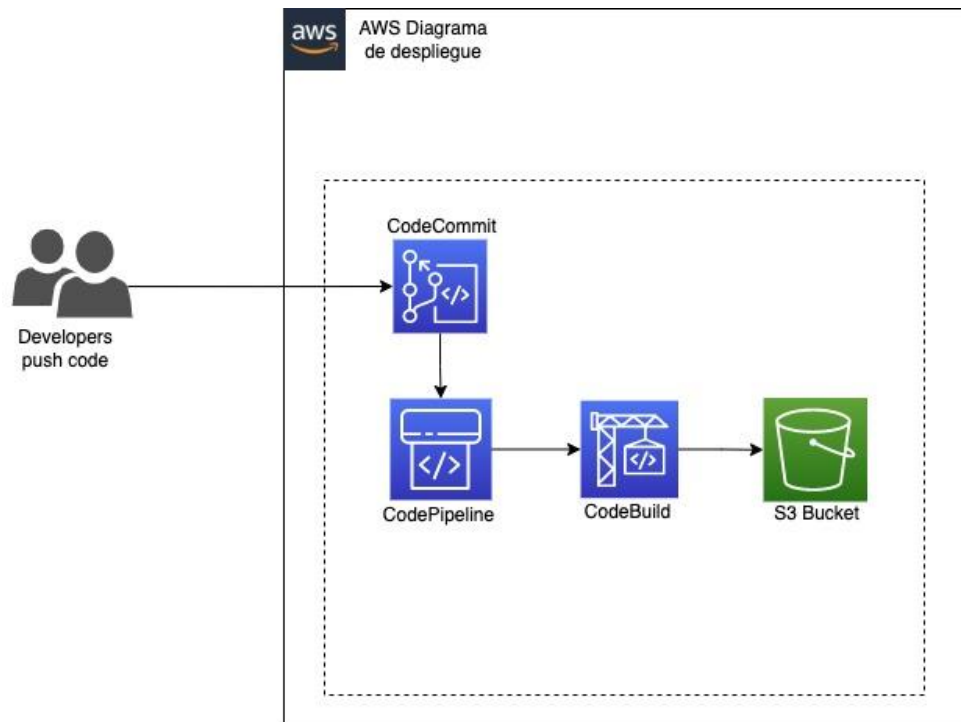
El backend construido en NodeJS se convirtió en una API Gateway de AWS utilizando Lambdas (basado en NodeJS) , ya que se encontró como ventaja la posibilidad de tener una infraestructura escalable, tiempo de desarrollo más corto, ejecuciones simultáneas, optimización en tiempo y costo de ejecución. Las Lambdas se construyeron utilizando Javascript. Se agregó una Lambda de autorización y un token en la cabecera, además de Cryptojs para enviar las peticiones encriptadas e incrementar la seguridad en los servicios.

En el frontend el cambio significativo fue presentado en el tipo de lenguaje ya que se decidió utilizar Typescript en vez de Javascript como se tenían algunos de los módulos inicialmente. Typescript permitió dar mayor orden a los componentes, tener una sintaxis y estructura más limpia, además de mayor reuso de componentes.

La estructura del proyecto es por función o modular, permitiendo organizar por carpetas el código relacionado e importar en forma de módulos los elementos reutilizables. El código se organiza a nivel jerárquico y según su tipo dentro del proyecto.

Para el sitio web informativo el cual se usa como pantalla de bienvenida antes de ingresar a la herramienta y para proveer mayor detalle sobre el proyecto, se migró de Wordpress a ser una landing page construida utilizando React y montada en un bucket S3 de AWS como un sitio web estático.

El código fuente se encuentra almacenado en codecommit con la intención de generar despliegues automáticos, es decir, al momento de realizar push o merge sobre la rama master, se genera un nuevo compilado según el proyecto modificado, ya sea del frontend de la herramienta o el sitio web, sin tener qué interferir manualmente en el proceso. En la Figura 40 se muestra el diagrama de despliegue de la herramienta y el sitio web informativo.



**Figura 40:** Diagrama de despliegue de herramienta web y sitio web informativo

La herramienta construida puede ser accedida a través del siguiente enlace:

<https://misuperpodereslaprogramacion/>

## 8. PRUEBAS

En este capítulo del proyecto se pone en evidencia los resultados obtenidos por medio de las pruebas de la herramienta web Mi Superpoder es la Programación, en donde se buscaba obtener información sobre la funcionalidad, experiencia de usuario y motivación en el uso del prototipo por parte de los niños. En las pruebas también se buscaba evaluar el correcto funcionamiento de los dos módulos con los que actualmente cuenta la herramienta. El módulo 1 se encuentra enfocado en secuencias y desarrollo de estructuras lógicas y el módulo 2 en diagramas de flujo.

Así como se definió en los objetivos y alcance de este proyecto, las pruebas se desarrollaron con un total de 13 niños, en un ambiente completamente virtual utilizando Google Meets como herramienta de comunicación, donde participaron niños de edades entre los 6 y 11 años de diferentes departamentos y países como el Valle del Cauca, Antioquia, Guajira, Cundinamarca y Canadá, los cuales se conectaron utilizando celular o computador.

La convocatoria para las pruebas fue abierta y se utilizaron las redes sociales como medio de difusión, el proyecto social cuenta con página de Instagram y Facebook, las cuales han sido de ayuda para dar a conocer los talleres en el pasado. También se difundió la invitación a los niños para participar de las pruebas a través de LinkedIn, donde además se sumaron niños asistentes a la biblioteca Rafael Pombo De todos los niños inscritos 20 en total, se le dió prioridad a todos aquellos que no sabían programar y que participarían todos los días.

Para realizar las pruebas de la herramienta se construyó un taller con duración de 10 horas distribuidas en 2 horas semanales; los niños fueron separados en 3 grupos para facilitar la comunicación durante la jornada. Se diseñó un cronograma de trabajo donde se describe lo que se haría por cada día. Este cronograma de pruebas se presenta a continuación.

- **Día 1:** Llevar a cabo una entrevista inicial que permita obtener información de los niños participantes de las pruebas de la herramienta. Las preguntas a formular son las siguientes:
  - ¿Cuál es tu dispositivo electrónico favorito para navegar por internet?
  - ¿Cuál es tu aplicación favorita?
  - ¿Tienes algún videojuego favorito?, ¿cuál?
  - Qué te gusta más, ¿hacer tareas desde el computador o en el cuaderno?
  - ¿Te aburres al momento de usar un computador, tablet o celular?
  - ¿Qué es para tí la programación?
  - ¿Para qué crees o te imaginas que sirve la programación?
  - ¿Sabes programar?, si la respuesta es sí, nombra en que herramientas estás

aprendiendo o aprendiste a hacerlo.

- ¿Qué es lo que más te gusta y menos te gusta de usar internet?

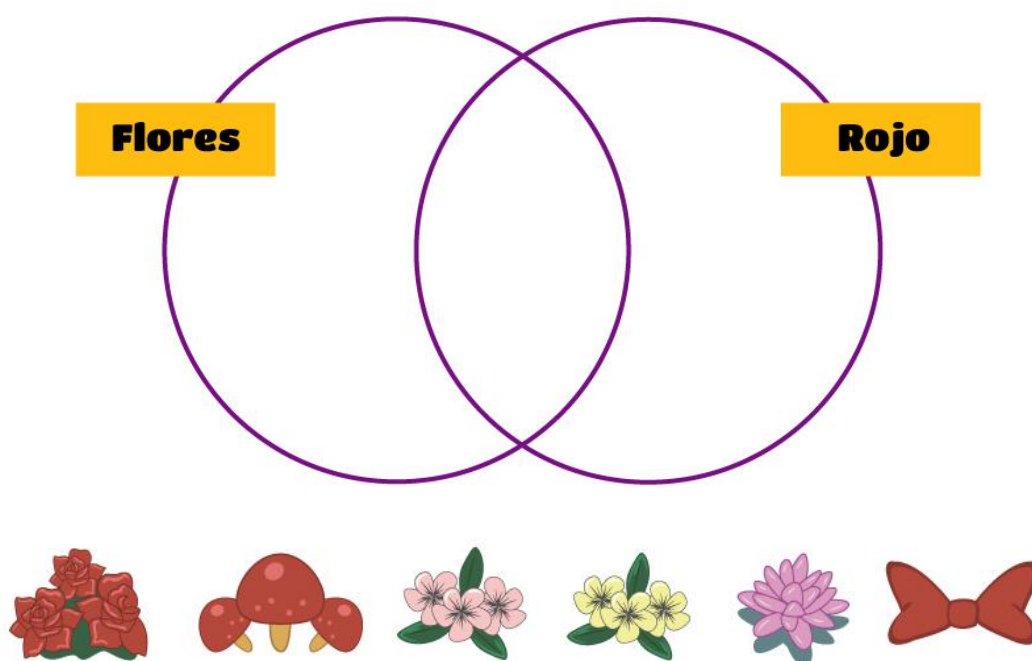
Adicionalmente, en el proceso de captura de información se deben obtener los conocimientos previos que permitirán posterior al uso de la herramienta, validar si los ejercicios planteados otorgan conocimientos sobre los temas esperados. Para lograr esto se diseñó un cuestionario que contiene preguntas sobre secuencias lógicas, conjuntos y construcción de algoritmos en papel (Ver [anexo 3](#))

- **Día 2:** Explicación de la actividad a realizar con la herramienta web, entrega de credenciales de acceso y desarrollo por parte de los niños de todos los niveles o módulos de la herramienta. Observar detalladamente la interacción por parte de los niños y capturar información durante el transcurso de las pruebas asociada a satisfacción y entendimiento de los ejercicios.
- **Día 3:** Desarrollo de los puntos del cuestionario de ejercicios previos realizados en el día 1. En esta ocasión los niños deberán resolver los puntos utilizando secuencias y diagramas de flujo tal y como lo aprendieron de la herramienta web.
- **Día 4:** Desarrollo de actividades utilizando Code.org con el fin de comparar ambas herramientas y determinar cual logró captar de la mejor manera la atención de los niños. Además de identificar cuál se ajusta mejor a sus necesidades y es más fácil de manejar. Se decide utilizar Code.org, ya que 2 niños asistieron al taller utilizando dispositivos móviles y como se validó previamente con las herramientas analizadas, esta es de las que mejor se ajusta a dispositivos celulares.
- **Día 5:** Realización del formulario de satisfacción y evaluación de la herramienta, entrevistas para conocer sugerencias de los niños frente a las actividades realizadas en la herramienta y dinámica del taller. Revisión de los resultados, captura de errores y definición de plan de mejora. Fin y cierre de las pruebas.

Posterior a la captura de la información fue posible obtener resultados que permitieron conocer si realmente la herramienta web tiene el impacto esperado. Los resultados fueron separados en 1) análisis cualitativo, en donde se describe cada uno de los elementos identificados desde la observación y desde lo que se obtuvo a través del formulario de conocimientos previos y 2) análisis cuantitativo, para comparar resultados a través de números y gráficas.

## 1) Análisis cualitativo

Para este análisis se llevó a cabo la captura de la información por medio del cuestionario inicial de conocimientos previos el cual fue clave, ya que permitió identificar los conocimientos que poseían los niños antes del uso de la herramienta. Se obtuvo que al menos 5 niños tuvieron problemas con la asignación de elementos en conjuntos, ya que se tornaba confuso colocar los elementos que cumplieran la características del conjunto A y del conjunto B (Ejercicio de ejemplo Figura 41).



**Figura 41:** Ejercicio para la identificación de elementos y separación utilizando conjuntos

Las secuencias y la construcción de algoritmos paso a paso utilizando lenguaje natural la realizaron con normalidad. Sin embargo, se destaca que en la construcción de algoritmos los niños a pesar de escribir un paso a paso detallado, tienen tendencia a descartar elementos que consideran obvios o demasiado repetitivos, donde estos eran asumidos y daban por hecho que en este caso el profesor entendería completamente el proceso del algoritmo descrito. La Figura 42, fue utilizada como referencia para solicitar a los niños diseñar un algoritmo sobre el lavado de los dientes.



**Figura 42:** Imagen de referencia para la construcción del algoritmo para describir el paso a paso de lavarse los dientes

En secuencias algunos niños no lograron identificar correctamente el elemento que sigue, debido a que se enfocaban en el elemento anterior y no en el siguiente. Muchos niños tenían mayor capacidad de observación lo que les permitía lograr con mayor facilidad y más rápidamente la solución de los ejercicios. Sin embargo, si un elemento de la secuencia se encontraba en la mitad de la cadena a evaluar generaba mayor confusión en el análisis, lo que proporcionaba resultados mayoritariamente incorrectos. Se identificó que para algunos niños es más fácil encontrar el primer y último elemento de la secuencia (Ejercicio de ejemplo Figura 43).

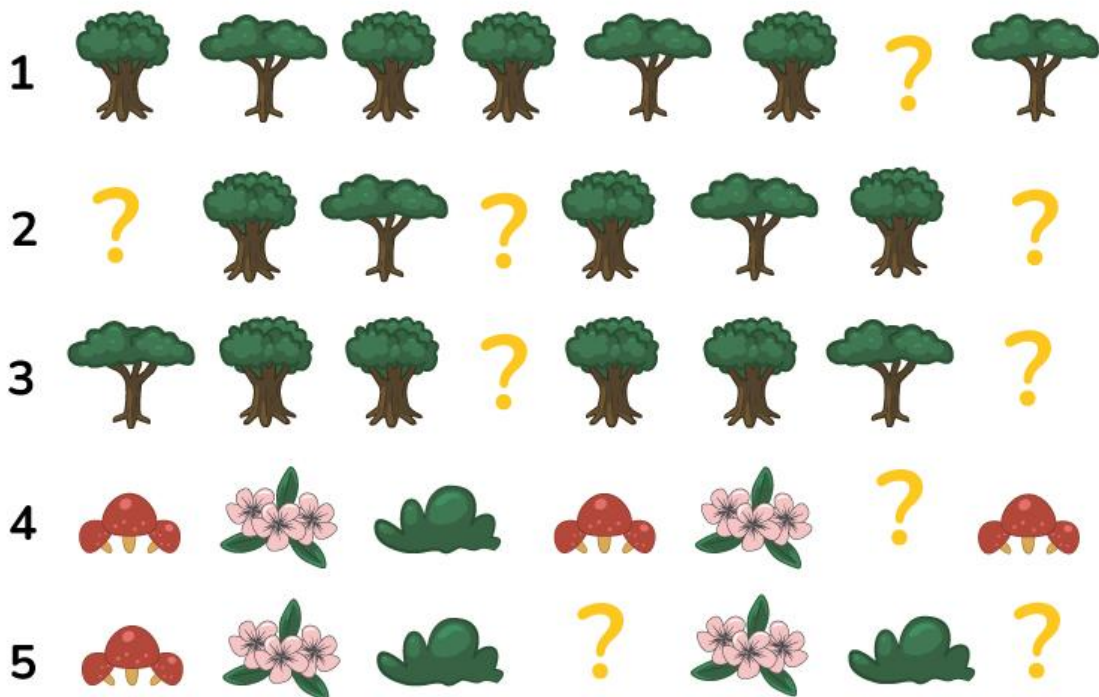


Figura 43: Ejercicio de identificación del elemento faltante de la secuencia

En uno de los ejercicios diseñado para convertir figuras en letras del abecedario, no se observaron dificultades y todos los niños lograron realizar ese punto del cuestionario sin problemas, comprendiendo que existen diferencias entre el lenguaje que hablamos las personas para comunicarnos y el lenguaje de máquina para dar indicaciones a los computadores, que era lo que se intentaba simular con este ejercicio como se puede observar en la Figura 44.

|          |  |          |  |          |  |          |  |          |  |          |  |
|----------|--|----------|--|----------|--|----------|--|----------|--|----------|--|
| <b>A</b> |  | <b>F</b> |  | <b>K</b> |  | <b>O</b> |  | <b>T</b> |  | <b>Y</b> |  |
| <b>B</b> |  | <b>G</b> |  | <b>L</b> |  | <b>P</b> |  | <b>U</b> |  | <b>Z</b> |  |
| <b>C</b> |  | <b>H</b> |  | <b>M</b> |  | <b>Q</b> |  | <b>V</b> |  |          |  |
| <b>D</b> |  | <b>I</b> |  | <b>N</b> |  | <b>R</b> |  | <b>W</b> |  |          |  |
| <b>E</b> |  | <b>J</b> |  | <b>Ñ</b> |  | <b>S</b> |  | <b>X</b> |  |          |  |

Figura 44: Ejercicio de traducción de figuras a lenguaje natural presente en el taller de conocimientos previos

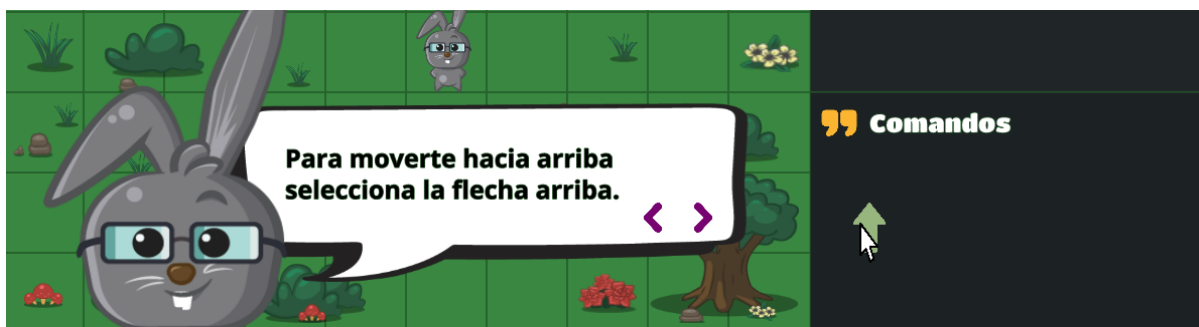
El ejercicio diseñado para trazar la cantidad de pasos a dar desde una posición inicial (figura rayo) hasta el personaje del mapa indicado, muchos niños colocaron solo el valor numérico de la cantidad de pasos a realizar o colocaron el número al lado de la palabra que indica la acción como por ejemplo 5 arriba, 3 abajo. Los niños proporcionaron diferentes soluciones, variaron principalmente en la cantidad de pasos descrita ya que prefieren escribir la menor cantidad de veces posible.



**Figura 45:** Ejercicio para la creación del camino desde el inicio hasta el personaje

Al pasar a utilizar la herramienta web Mi Superpoder es la Programación, no se presentaron complicaciones ya que todos los niños participantes habían tenido contacto previo con dispositivos electrónicos como celular, tablet o computador. Se destaca también, que los niños han interactuado con internet y la gran mayoría ya sabía cómo hacer búsquedas utilizando un buscador. Para el caso de los niños más pequeños con edades entre los 6 y 7, estos llegaron a considerar que una forma correcta de buscar un sitio web era a través de Youtube.

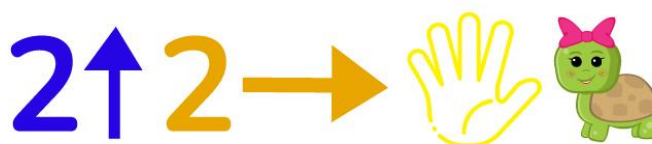
Durante el uso de la herramienta se observó que los niños entendieron completamente las instrucciones indicadas por el avatar de apoyo, pero se notó que al querer repetir nuevamente el ejercicio y no contar con la opción de saltar la introducción del nivel, manifestaron inconformidad. Con respecto a los primeros niveles, los niños desarrollaron los ejercicios de secuencia sin ningún problema, comprendieron la dinámica de juego fácilmente y no fue necesario recibir apoyo para la solución de estos ejercicios, se valieron únicamente de las instrucciones y el avatar de apoyo como se muestra en la Figura 45.



**Figura 46:** Avatar de apoyo el que se muestra en la herramienta web

Se destaca que para los ejercicios del módulo 2, los niños más pequeños tuvieron dificultades para comprender las decisiones de los diagramas de flujo por lo que fue necesario repetir los niveles enfocados en este tema al menos 2 veces.

Al finalizar los ejercicios disponibles en la herramienta web, se desarrolló la segunda parte del cuestionario y fue posible observar que con respecto a la creación de caminos utilizando flechas, los niños no tuvieron problemas y pudieron resolver el ejercicio relacionado sin ningún inconveniente, ya que comprendieron correctamente las direcciones (arriba, abajo, izquierda, derecha). Además, lograron identificar la cantidad de pasos en una determinada dirección y convertir en diagramas de flujo las instrucciones como se muestran en la Figura 26.

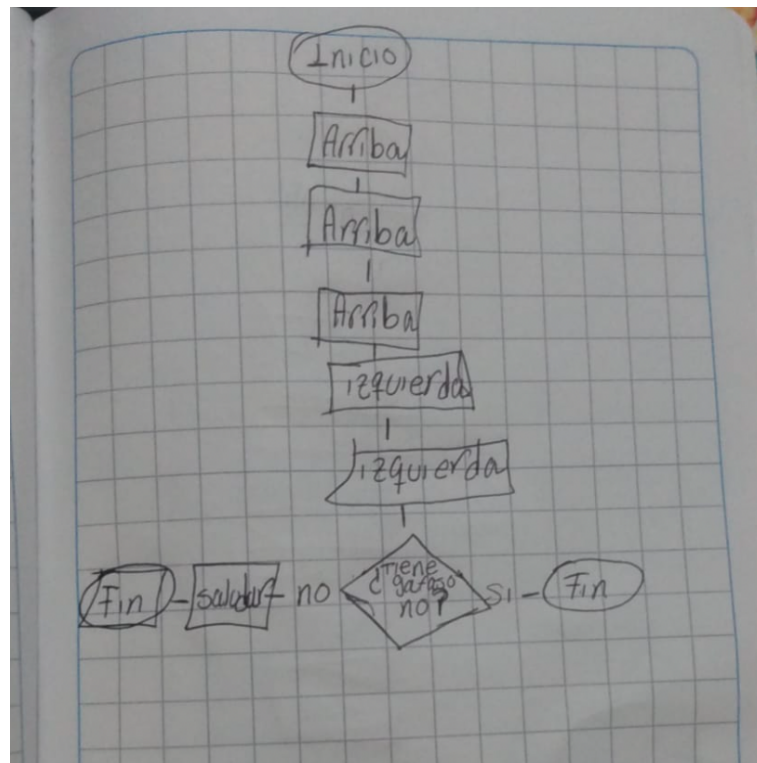


**Figura 47:** Ejercicio de secuencia de pasos a convertir en diagrama de flujo

En los diagramas de flujo, los niños no tuvieron problemas para comprender los elementos de inicio, proceso y fin, todos comprendieron la finalidad de cada uno. Para el caso de la decisión muchos de los niños asumieron la cuestión o pregunta de manera implícita, es decir, comprendieron la funcionalidad del elemento decisión pero no consideraron necesario representarla en el flujo.

Uno de los casos más representativos fue el de construir un diagrama utilizando una decisión con dos caminos distintos para el caso en que el conejo de una imagen tuviera gafas

o no; en el caso de que el conejo no tuviera gafas, se debía saludar en el caso de que sí, se debía finalizar el diagrama. Como muchos niños veían en la imagen al conejo con gafas, no hicieron el camino para el caso en que la respuesta a la pregunta fuera negativa (NO). Se esperaba que asumieran ambos casos y los representaran. Un ejemplo de solución correcta implementando los caminos para los dos posibles casos, se puede observar en la figura 47.

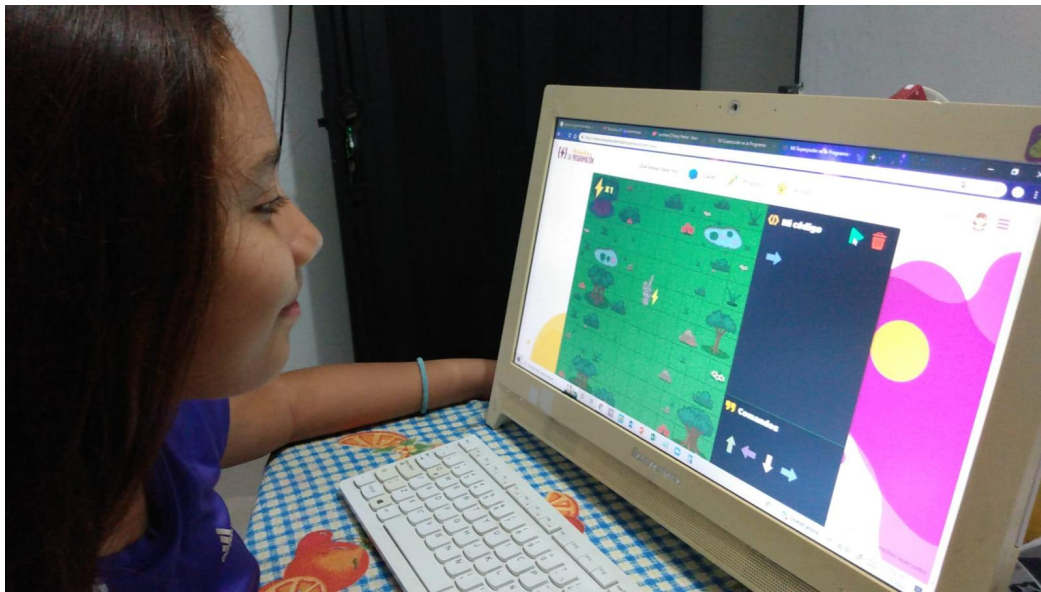


**Figura 48:** Diagrama de flujo creado por uno de los niños participantes de las pruebas

A nivel general es posible determinar que la herramienta cumplió la expectativa, ya que los niños comprendieron su funcionalidad, los ejercicios fueron considerados como entretenidos y divertidos. Además, posterior al uso de la herramienta, los niños pudieron enfrentarse con ejercicios sobre temas que aprendieron utilizando la misma.

Adicionalmente, en el proceso de pruebas se hizo uso de la herramienta code.org para indicar por medio de bloques instrucciones a un personaje. En el caso de los niños menores de 9 años se identificó que las instrucciones del juego eran más difíciles de entender y los bloques no eran muy claros al momento de intentar indicar instrucciones. Los niños mayores de 9 años, simpatizaron con esta herramienta, incluso manifestaron que les llamaba la atención el uso de sonidos y mayor complejidad en los ejercicios, recomendaron tener un módulo en bloques o similar, el cual les permita incrementar el nivel.

En la Figura 48 se observa a una de las estudiantes participantes del taller haciendo uso de la herramienta. Para la captura de la información a través de entrevistas, videos y fotografías, se entregó a los acudientes, padres de familia y representante legal las políticas de privacidad de la información, donde se describe el uso a otorgar de los datos capturados de los niños y la manera en que estos serán procesados. Las imágenes donde aparecen los niños utilizando la aplicación serán utilizadas únicamente con fines académicos. Para mayor detalle ir al [Anexo 4](#).



**Figura 49:** Estudiante haciendo uso de la herramienta web Mi Superpoder es la Programación

## 2) Análisis cuantitativo

Los niños que hicieron parte de las pruebas, tienen como aplicación favorita Youtube, Tik Tok, Roblox y Minecraft. A pesar de que Roblox y Minecraft se utilizan para enseñar programación, al momento en que se les preguntó si conocían algo sobre programación, la respuesta fue negativa.

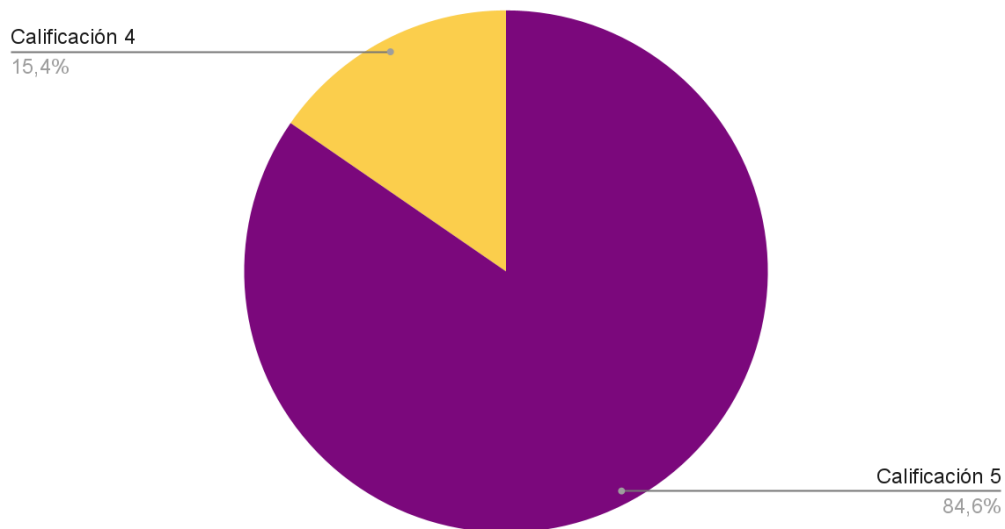
3 de los niños participantes estuvieron en los talleres de años anteriores de Mi Superpoder es la Programación en el año 2019 y 2020, por lo que al preguntárseles sobre programación dieron respuestas como: “la programación permite dar instrucciones” y “la programación es un conjunto de comandos”. Ellos también indicaron que sabían programar, ya que interactuaron en el pasado con la herramienta CodeMonkey.

La mayoría de los niños prefiere utilizar el celular, ya que para muchos de ellos es el dispositivo con el que principalmente tienen contacto. Muchos niños utilizan el celular para investigar sus tareas y hacerlas. Sin embargo, para las pruebas recurrieron a los computadores de algún familiar. Para la solución de dudas de las tareas e investigaciones suelen recurrir a videos de Youtube.

Los niños suelen perder interés al momento de utilizar dispositivos electrónicos cuando han finalizado sus tareas o las aplicaciones y juegos con lo que interactúan no les ofrece retroalimentación. Para finalizar las pruebas, posterior a la entrevista y al uso de la herramienta, a los niños se les entregó un formulario el cual permitió obtener información sobre nivel de satisfacción, experiencia en el uso, con espacio para realizar propuestas de mejora y recomendaciones por parte de los niños, también se capturó información sobre lo que más o menos les gustó.

En la Figura 49 se observa que al menos el 15.4%, lo que corresponde a 2 niños de 13 evaluados, dieron una calificación de 4 en facilidad de uso de la herramienta. A estos niños se les preguntó el motivo de la calificación, a lo que indicaron que el manejo del avatar a veces era complejo ya que no permitía saltar las instrucciones a pesar de que el nivel ya había sido superado, o no se indicaba en qué momento se podía pasar al siguiente mensaje, cuando terminara este de hablar.

#### Facilidad de uso de la herramienta

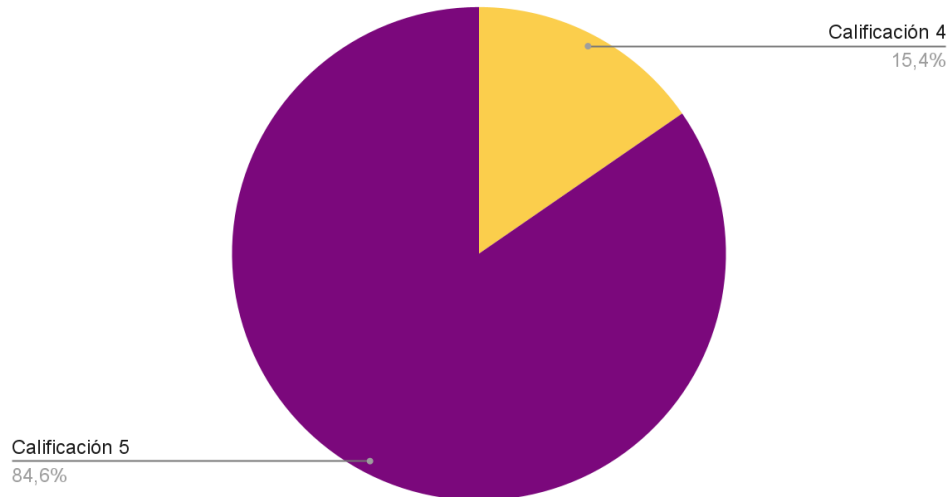


**Figura 50:** Gráfica resultado de las respuestas para la pregunta asociada a facilidad de uso de la herramienta

En apariencia visual de la herramienta, 2 niños correspondiente al 15.4% asignaron una

calificación de 4, ya que consideraron que hace falta sonidos, imágenes y videos que mejoren la interacción y el entendimiento de algunas opciones de la aplicación. Sin embargo, todos estaban a gusto con los colores utilizados, tipografía y estructura de los elementos.

#### Apariencia visual de la herramienta



**Figura 51:** Gráfica resultado de las respuestas para la pregunta asociada a la apariencia visual de la herramienta

Para el caso de las actividades jugadas, todos los niños estuvieron satisfechos con las actividades, las encontraron divertidas, fáciles de entender y de ejecutar. Consideraron que el Avatar es de gran ayuda para el entendimiento del ejercicio. De los 13 niños evaluados, 3 prefieren el módulo 2 por su complejidad, ya que eran más grandes y estaban en búsqueda de actividades retadoras.

A los niños evaluados se les preguntó si recomendarían la herramienta a lo que todos respondieron positivamente que sí lo harían. Posterior a esta pregunta se les consultó sobre elementos que le agregarían a la herramienta a lo que los niños respondieron.

- Tamaño elementos, 1 niño (7.7%)
- Personajes, 2 niños (15.4%)
- Sonidos, 1 niño (7.7%)
- Más niveles, 4 niños (30.8%)
- Nada nuevo, 5 niños (38.5%)

Los resultados obtenidos de esta pregunta se muestran a continuación en la figura 51.



**Figura 52:** Gráfica resultado de las respuestas para la pregunta asociada a qué le agregarías a la herramienta

Se les consultó si le cambiarían algo a la herramienta a lo que el 100% de los niños indicó que no cambiarían nada. En el caso de los dispositivos utilizados 2 niños de 13 años hicieron las pruebas de la herramienta por medio de sus dispositivos celulares sin ningún problema, esto se puede ver en la figura 52.



**Figura 53:** Gráfica resultado de las respuestas para la pregunta asociada a dispositivos utilizados

En las pruebas los niños no identificaron errores, sin embargo proponen mejorar el tiempo en que el Avatar dice el mensaje y la posibilidad de contar con un botón para repetir la instrucción, sobre todo en el módulo 2 que es donde se tiene mayor complejidad en los ejercicios y se requiere mayor detalle en la instrucción.

Previo a las pruebas con los niños, se hicieron pruebas de carga, estrés, usabilidad y funcionalidad con 35 estudiantes de la Universidad de San Buenaventura del programa de ingeniería de sistemas, pertenecientes a la clase Introducción a la ingeniería de software. Se observó por medio de la consola de AWS el correcto funcionamiento de la aplicación y las múltiples conexiones sin problemas.

El cuestionario de evaluación fue realizado solo por 23 estudiantes, donde 6 de ellos utilizaron su celular como dispositivo de pruebas. El elemento que destacaron de la aplicación fue el diseño y la fácil interacción con los elementos y acceso a los módulos de juego. Los errores que más destacaron fueron asociados a los componentes que se arrastran al espacio de juego, ampliar el área de selección para que sea más fácil llevar el comando al espacio asignado.

Algunos estudiantes presenciaron errores con el comportamiento del personaje ya que era posible activar movimientos sin terminar de realizar las instrucciones o sin finalizar de organizar los comandos. Los mensajes podían pasarse con rapidez aunque aún el avatar no hubiera terminado de dar la indicación y se reportó problemas de responsive en dispositivos diferentes a computador o celular. Los problemas reportados por los estudiantes fueron solucionados antes de las pruebas con los niños y se garantizó que los comportamientos del personaje y las instrucciones ocurrieran en el momento indicado o esperado.

Durante las pruebas fue posible identificar que los niños tienen diferentes formas de aprender según su edad y capacidad de aprendizaje, por lo que será necesario realizar ajustes en algunas de las actividades de acuerdo a sus necesidades. Esto con el fin de mejorar la experiencia de usuario y contenido de aprendizaje, recibiendo apoyo de profesionales en el área de la educación.

## 9. CONCLUSIONES Y TRABAJOS FUTUROS

### 9.1 Conclusiones

A partir de las pruebas realizadas a la herramienta, es posible determinar que se cumplieron los objetivos planteados como lo son la creación de un prototipo web para llevar a cabo la enseñanza de la programación en niños y jóvenes, siendo posible acceder a esta herramienta desde dispositivos celulares.

El prototipo construido cuenta con dos módulos iniciales donde es posible aprender sobre secuencia y diagramas de flujo. Se definieron un conjunto de requisitos funcionales y no funcionales que fueron la base para la construcción de los mockups, que se transformarían finalmente en las pantallas finales.

Para lograr el desarrollo de la herramienta se analizaron y estudiaron las herramientas más utilizadas actualmente gratuitas y de pago. Esto fue útil en la identificación de las debilidades y creación de los requisitos a partir de las necesidades que se han presentado al momento de dictar los talleres de programación por medio de herramientas de terceros.

Se estudiaron diversas arquitecturas, incluso el diagrama de arquitectura pasó por múltiples etapas para finalmente elegir la más conveniente con mayores posibilidades de escalabilidad, optimización y rendimiento.

Las pruebas con los niños permitieron determinar puntos claves de la herramienta y posibilidades de mejora. De este proceso de pruebas es posible concluir lo siguiente:

- A los niños les gustó hacer uso de la herramienta, les pareció entretenida, divertida y fácil de utilizar. Los niveles ofrecidos les parecieron llamativos y les permitió aprender sobre secuencias y diagramas de flujo. Los conocimientos adquiridos se vieron reflejados en las respuestas al cuestionario realizado al finalizar los ejercicios.
- A pesar de las múltiples conexiones en simultáneo, el diseño de la arquitectura es escalable y no afectó a ninguno de los usuarios. Todos los usuarios lograron conectarse sin problemas y estar en videollamada con un profesor a través de Google Meets.

- Las posibilidades de acceso a la herramienta desde celulares le permite a los niños que no cuentan con computador hacer actividades diferentes a estar en Youtube o redes sociales como Tiktok. Además pueden potencializar sus conocimientos sin necesidad de poseer dispositivos con muchas capacidades de hardware.
- El avatar influye en las decisiones del usuario al momento de realizar la actividad, por esta razón las explicaciones deben ser claras y detalladas. El tutorial de introducción debe complementarse con más recursos interactivos como audios.
- Los niños siempre buscan tener el control completo de lo que realizan, por lo que es necesario que el futuro las soluciones puedan ser libres y que no siempre sean limitadas o el sistema no deje continuar si no tiene las características o comandos esperados.
- La pandemia fue clave para la decisión de la virtualización de las clases dictadas en Mi Superpoder es la Programación, ya que permite manejar un modelo híbrido permitiendo llegar a muchos más niños sin importar la ciudad o país donde se encuentren. Esta herramienta surgió gran parte de su propuesta y desarrollo durante la pandemia. Durante esta etapa fue posible comprender de mejor manera las necesidades de los niños y jóvenes al ver una clase virtual.
- Con la implementación del despliegue continuo a través de *pipelines*, se garantiza que es posible subir cambios de la herramienta o sitio web mientras está en uso sin generar indisponibilidad y afectar la interacción del usuario con la herramienta.
- El prototipo cumple con lo esperado y se logra obtener resultados relevantes y positivos en la experiencia de usuario, satisfacción, motivación y facilidad en los ejercicios.

A través de la comparación realizada en el capítulo 4.1, precisamente en la [Tabla 1](#), se puede concluir que el prototipo de la herramienta construida cumple con el 87.5% de los elementos necesarios para satisfacer las necesidades identificadas. Esto se refleja en la Tabla 6, donde se puede observar que el prototipo cumple con 7 de los requisitos que permiten cubrir las necesidades identificadas.

**Tabla 6:** Extracción de características del prototipo de la herramienta Mi Superpoder es la Programación

| Herramienta                      | Características |                          |            |                |  |                                     |                            |          |
|----------------------------------|-----------------|--------------------------|------------|----------------|--|-------------------------------------|----------------------------|----------|
|                                  | Ambiente web    | Portabilidad (app móvil) | Responsive | Idioma Español | Ejercicios de algoritmos (previos a la programación) | Posibilidad de incrementar el nivel | Código diferente a bloques | Gratuita |
| Mi Superpoder es la Programación | ✓               | X                        | ✓          | ✓              | ✓  | ✓                                   | ✓                          | ✓        |

El prototipo de la herramienta se ejecuta sobre un ambiente web, es responsive y se adapta a dispositivos móviles, está en el idioma español, cuenta con ejercicios previos a la codificación, los cuales fueron utilizados para hacer la validación del prototipo de la herramienta, permite avanzar desde algo básico como el desarrollo de una secuencia hasta el desarrollo de un diagrama de flujo y comprender los elementos que lo conforman. Se aprenden las bases sin requerir de bloques y es gratuita. Esta herramienta no es portable ya que no cuenta con una aplicación móvil que le permita ser descargada desde una *app store*.

Sin embargo y debido a la tecnología con la que el prototipo fue desarrollado, es posible migrar de web a una aplicación móvil, pero se debe tener en cuenta que la migración puede requerir ciertos ajustes y adaptaciones para una mejor experiencia de usuario en dispositivos móviles.

Se identifica que es importante culminar todos los requisitos de la herramienta para hacerla más robusta y pueda contar con funcionalidades adicionales para los niños y jóvenes, donde sea posible conocer el progreso (RQ-PE-004) y se muestre un mapa para la selección de los niveles de manera interactiva (RQ-PE-005), complementando la manera en que se elige el nivel actualmente.

El perfil del profesor es de gran valor, ya que permite realizar un seguimiento detallado de los niños y jóvenes usuarios de la herramienta. Además, los profesores podrán fortalecer sus conocimientos y metodologías de la enseñanza en el área de la programación, mediante la realización de las actividades disponibles. Por esta razón es importante culminar los requisitos asociados a este perfil de la herramienta.

## 9.2 Trabajos Futuros

La herramienta actualmente sigue en proceso de desarrollo, ya que hace parte del proyecto social Mi Superpoder es la Programación, además se está en búsqueda de recursos para lograr culminar la creación de todos los módulos de la herramienta de los cuales hace parte:

- Nuevos niveles para los módulos de secuencia y diagramas de flujo
- Partes del computador
- Construcción de algoritmos usando pseudocódigo
- Desarrollo de lógica usando bloques
- Desarrollo de niveles utilizando código de Javascript y Python

Se espera finalmente cubrir desde la herramienta módulos y niveles que se ofrecen regularmente cuando se desarrollan los talleres, con el objetivo de que los niños y jóvenes puedan aprender sobre:

- Qué es un autómata
- Secuencias y patrones
- Diagramas de flujo
- Pseudocódigo
- Variables
- Condiciones y Ciclos
- Funciones
- Estructuras de datos
- Programación orientada a objetos (POO)

Durante las pruebas del prototipo de la herramienta se identificó que es necesario mejorar el responsive para cumplir con la totalidad de la adaptabilidad de la herramienta. Aunque como objetivo de este proyecto estaba ejecutar la herramienta desde la web y desde dispositivos móviles, lo cual a día de hoy es posible, es necesario ajustar algunos elementos que pueden afectar la apariencia de la herramienta a nivel de interfaz si se usa otro dispositivo como tablet.

Como trabajo futuro se espera obtener fondos a través de convocatorias del gobierno, para que sea posible incrementar los recursos de la herramienta, ya que desde AWS se cobra mes a mes por los recursos utilizados. En promedio y con la nueva arquitectura se está pagando de 10 a 30 USD, depende de la cantidad de usuarios que hagan consumo y uso de la herramienta ese mes.

Se espera agregar a la herramienta las funcionalidades extras como lo son interfaz para el portal del profesor, ya que todo se encuentra nivel de servicios desde el API Gateway construido con Lambdas y maquetación frontend. Los niños en las pruebas pidieron mayor

interacción y nuevos personajes por lo que se tiene como objetivo del trabajo futuro lograr esto y permitir que los sonidos, audios e interacciones enriquezcan la experiencia de usuario.

Se debe agregar a la herramienta los requisitos faltantes como lo son: 1) requisitos funcionales para el rol profesor enfocándose principalmente en la interfaz, ya que se cuenta con los servicios que se consumen a través de Postman para la creación de clases, estudiantes y asignación de módulos; y 2) culminar los requisitos del rol estudiante RQ-PE-004 donde se debe mostrar el puntaje adquirido y no solo el progreso o porcentaje de avance, RQ-PE-005 donde se muestre un mapa de selección de actividades para enriquecer la experiencia de usuario, actualmente se muestra una modal informativa y RQ-PE-008 con el fin de permitir múltiples respuestas a partir de la lógica de los usuarios, en algunos ejercicios actuales solo se permite una solución concreta.

Se proyectan modificaciones sobre el esquema de base de datos donde aparecerán nuevos documentos para enriquecer la plataforma a nivel de datos y manejo de componentes de interacción como por ejemplo el chat o intercambio de mensajes entre profesores y estudiantes.

Se plantea a futuro pruebas a los requisitos no funcionales donde sea posible identificar el rendimiento, la seguridad, escalabilidad, disponibilidad y usabilidad de la herramienta, para asegurar el cumplimiento en todas las expectativas esperadas y donde además sea posible determinar la calidad del software. Aunque previo a las pruebas con los niños, se llevaron a cabo pruebas de rendimiento con alrededor de 35 usuarios contactados al tiempo, se deben evaluar otros componentes como la seguridad para identificar vulnerabilidades, fiabilidad para conocer la capacidad de su funcionamiento en diferentes situaciones o selecciones y escalabilidad para conocer el manejo de los recursos de AWS al presentar un aumento en la cantidad de usuario conectados, datos o transacciones cuando el software está sometido a estrés.

Aunque en las pruebas de la herramienta estuvieron dirigidas a niños de edades entre los 6 y 12 años, en el futuro se espera validar la herramienta con jóvenes de edades entre los 13 y los 18 años, para así establecer si hay diferencias que deben ser plasmadas en la herramienta. Se resalta que las pruebas iniciales para identificar errores en el funcionamiento de la herramienta, fueron realizadas con personas de edades desde los 17 hasta los 30 años, donde indicaron que los niveles son adecuados para personas sin experiencia en la programación.

Se considera como trabajo futuro la internacionalización de la herramienta, donde se cuente con la selección de idiomas como inglés y portugués, con el propósito de que esta pueda ser utilizada por otros niños y jóvenes que se encuentren por fuera de Colombia y no pertenezcan a un país hispano hablante.

Como trabajo futuro se espera presentar el artículo sobre este proyecto en al menos uno de los congresos 1) CLEI 2023<sup>17</sup>, más específicamente en el Simposio Latinoamericano de Ingeniería de Software ( SLISW)<sup>18</sup>, en el tópicos aspectos sociales de la ingeniería de software, con indexación de los trabajos a presentar en IEEE Xplore Digital Library. 2) Simposio latinoamericano de informática y sociedad (SLIS)<sup>19</sup>, en el tópicos educación y tecnologías digitales, perteneciente también al CLEI; y 3) CibSE 2023<sup>20</sup> XXVI Ibero-American Conference on Software Engineering, en el tópicos ingeniería de software en la industria ya que se utilizaron tecnologías que son el core de muchas de las aplicaciones en las industrias de TI actualmente. Los artículos de este congreso serán indexados en DBLP y Scopus.

Se buscan congresos de preferencia latinoamericanos ya que el objetivo es impactar principalmente en estos países. En talleres anteriores niños de Ecuador y Perú han hecho parte y se ha logrado identificar que tienen modelos educativos similares al colombiano.

## REFERENCIAS

- [1] Dane.gov.co. Boletín Técnico Indicadores básicos de tenencia y uso de tecnologías de la información y comunicación en empresas (TIC empresas) 2018. [online] Disponible en: [https://www.dane.gov.co/files/investigaciones/boletines/tic/bol\\_empresas\\_2018.pdf](https://www.dane.gov.co/files/investigaciones/boletines/tic/bol_empresas_2018.pdf) (último acceso 1 Octubre 2020)
- [2] Dane.gov.co. Boletín Técnico Educación Formal (EDUC) 2019. [online] Disponible en: [https://www.dane.gov.co/files/investigaciones/boletines/educacion/bol\\_EDUC\\_19.pdf](https://www.dane.gov.co/files/investigaciones/boletines/educacion/bol_EDUC_19.pdf) (último acceso 7 Junio 2020)
- [3] González-Quintero M, 2016. El problema educativo colombiano. Aibi revista de investigación, administración e ingeniería, pp.63-67
- [4] Dane.gov.co. Boletín Técnico Indicadores Básicos De TIC En Hogares - 2017. [online] Disponible en: [https://www.dane.gov.co/files/investigaciones/boletines/tic/bol\\_tic\\_hogares\\_2017.pdf](https://www.dane.gov.co/files/investigaciones/boletines/tic/bol_tic_hogares_2017.pdf) (último acceso 7 Junio 2020)
- [5] Jara I, Hepp P, 2016. Enseñar Ciencias de la Computación: creando oportunidades para los jóvenes de América Latina. Repositorio institucional - MINEDU Perú.
- [6] Resnick M, Maloney J, Monroy-Hernandez A, et al, 2017. Programación para Todos. Communications of the ACM, vol. 52, pp.60-67
- [7] Seymour, A. Papert. 1980. Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

---

<sup>17</sup> XLIX Conferencia Latinoamericana de Informática <https://conferencia2023.clei.org/call-for-papers/>

<sup>18</sup> Simposio Latinoamericano de Ingeniería de Software <https://conferencia2023.clei.org/simposios/slisw/>

<sup>19</sup> Simposio Latinoamericano de Informática y Sociedad <https://conferencia2023.clei.org/simposios/slis/>

<sup>20</sup> XXVI Ibero-American Conference on Software Engineering <https://conf.researchr.org/home/cibse-2023>

- [8] Torres F, Vivas G, 2009. Nativos digitales: ¿ocultamiento de factores generadores de fracaso escolar?. Revista iberoamericana de educación, no 50, p. 113-130.
- [9] Kalelioglu Filiz, 2015. A new way of teaching programming skills to K-12 students: Code.org. Computers in Human Behavior, vol. 52, p. 200-210.
- [10] Armoni M, Meerbaum-Salant O, Bem-Ari M, 2015. From scratch to “real” programming. ACM Transactions on Computing Education. Vol 14, p. 1-15.
- [11] Meerbaum-Salant O, Armoni M, Ben-Ari M, 2011. Habits of programming in scratch. Proceedings of the 16th annual joint conference on Innovation and technology in computer science education, p. 168-172.
- [12] Palma Suárez C, Sarmiento Porras R, 2015. Estado del arte sobre experiencia de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. Revista mexicana de investigación, P. 607-641.
- [13] Iskrenovic-Momcilovic O, 2020. Improving geometry teaching with Scratch. International Electronic Journal of Mathematics Education, vol. 15, p. em0582.
- [14] F. Biocca y M. R. Levy, 2013. “Communication in the age of virtual reality”, Routledge.
- [15] Rodríguez Santiago, López Pablo, Rivera Valery, Neira Juliana, 2017. THINK ING: UNA CHISPA DE INGENIO EN LA SOCIEDAD. Encuentro internacional de Educación en ingeniería.
- [16] Lamb Annette, Johnson Larry, 2011. Scratch: Computer programming for 21st century learners.
- [17] Ahumada H. C, et al, 2018. Pensamiento computacional mediante programación por bloques: intervención didáctica usando pilas bloques. XII Congreso de tecnología en educación.
- [18] Ching Yu-Hui, Yu-Chang Baldwin, 2018. Developing computational thinking with educational technologies for young learners. TechTrends, vol. 62, p. 563-573
- [19] Sapura Danny, Oswari Liniyanti, 2020. Usability User Experience, Learning Motivation and Learning Rate Characteristic of Learning Programming Using Game Making onUsers with Different Computer Experience. International Conference on Information Technology and Its Application, p. 649-654.
- [20] PALMA SUAREZ, Carlos Andrés y SARMIENTO PORRAS, Román Eduardo, 2015. Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria, vol.20, pp.607-641
- [21] García de los Salmones Gómez, P. 2017. Desarrollo de un videojuego para enseñar programación a niños.
- [22] Patiño, Alex Julián Montoya, Jorge Mario Hernández Álvarez, and Hermis Ramos Espinal, 2017. Analizar por qué se debe enseñar la lógica de programación a niños y jóvenes en los hogares de la ciudad de Medellín. Revista CIES Escolme 7.01, pp. 2-14.
- [23] González, Carina Soledad, 2018 "La enseñanza-aprendizaje del Pensamiento Computacional en edades tempranas: una revisión del estado del arte." Pensamiento computacional.
- [24] Ching Yu-Hui, Yu-Chang Baldwin, 2018. Developing computational thinking with educational technologies for young learners. TechTrends, vol. 62, p. 563-573

- [25] Basogain Olabe, X., Olabe Basogain, M. Ángel, & Olabe Basogain, J. C, 2015. Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. Revista de Educación a Distancia (RED)
- [26] Zapata-Ros, Miguel, 2018. Pensamiento computacional: Una tercera competencia clave. El pensamiento computacional como una nueva alfabetización en las culturas digitales, p. 4-87
- [27] Selwyn Neil, 2009. The digital native-myth and reality. En Aslib proceedings. Emerald Group publishing Limited.
- [28] Vidal Cristian et al, 2015. Experiencias prácticas con el uso del lenguaje de programación Scratch para desarrollar el pensamiento algorítmico de estudiantes en Chile. Formación universitaria, vol. 8, p. 23-32.
- [29] LEAD, 2012. The Lead Project, Super Scratch Programming Adventure!: Learn to Program by Making Cool Games.
- [30] Martín Pedro Pablo Gómez, Calero Pedro Antonio, Gómez-Martín Marco, 2004. Aprendizaje basado en juegos. Icono14, vol. 2, p. 1.
- [31] Ossa Guillermo, 2002. Tendencias Educativas para el siglo XXI. Educación virtual, online y @learning. Elementos para la discusión. Edutec, Revista electrónica de tecnología educativa. p. a025-a025
- [32] Requena Stefany, 2008. El modelo constructivista con las nuevas tecnologías: aplicado en el proceso de aprendizaje RUSC. Universities and Knowledge Society Journal, vol. 5, p. 26-35.
- [33] [mineducacion.gov.co](https://www.mineduacion.gov.co/1759/w3-article-196492.html?_noredirect=1), Educación virtual o educación en línea 2020. [online] Disponible en: [https://www.mineduacion.gov.co/1759/w3-article-196492.html?\\_noredirect=1](https://www.mineduacion.gov.co/1759/w3-article-196492.html?_noredirect=1) (último acceso 6 noviembre 2020)
- [34] García Lucy, Pernet Angélica, Cano John, 2017. Estudio exploratorio de usabilidad para niños de Colombia. Zona próxima, p. 12-30
- [35] Sherwin K, Nielsen J, 2019. Children's UX: Usability issues in designing for young people. Nielsen Norman Group, vol. 13.
- [36] Nielsen Jakob, 1995. Usability heuristics for user interface design. Nielsen Norman Group, vol. 1.
- [37] Claros I, Collazos C, 2006. Propuesta metodológica para la evaluación de la usabilidad en sitios web: Experiencia Colombiana. 7mo congreso internacional de interacción Persona-Ordenador, asociación para la interacción Persona-Ordenador, p. 165-174.
- [38] Bass, L., Clements, P., & Kazman, R, 2003. Software architecture in practice. Addison-Wesley Professional.
- [39] Richards, Mark, 2015. Software Architecture Patterns. O'Reilly, media.
- [40] Richardson, Chris, 2018. Pattern: Microservice Architecture [online] Disponible en <https://microservices.io/patterns/microservices.html> (último acceso 10 de julio 2021)
- [41] González, Y. D., & Romero, Y. F, 2012. Patrón Modelo-Vista-Controlador. Telemática, 11(1), 47-57

- [42] Enciso-Quispe, Liliana, et al, 2017. REST architecture in the implementation of a web and mobile application for vehicular tariff rotating parking. 12th Iberian Conference on Information Systems and Technologies (CISTI). IEEE. p. 1-6.
- [43] What is REST, 2021. [online] Disponible en: <https://restfulapi.net/> (último acceso 27 de julio 2021)
- [44] IBM, 2021. Rational Software Architect Standard Edition, SOAP. [online] Disponible en: <https://www.ibm.com/docs/es/rsas/7.5.0?topic=standards-soap> (último acceso 27 de julio de 2021)
- [45] Red Hat, 2021. Integration REST vs SOAP. [online] Disponible en: <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest> (último acceso 27 de julio 2021)
- [46] AWS, 2022. Creación de aplicaciones con arquitecturas sin servidor. [online] Disponible en: <https://aws.amazon.com/es/lambda/serverless-architectures-learn-more/#:~:text=What%20is%20a%20serverless%20architecture,management%20is%20done%20by%20AWS.>
- [47] Redhat, 2020. ¿Qué es la función como servicio (Faas)?. ¿Qué son las aplicaciones desarrolladas en la nube?. [online] Disponible en: <https://www.redhat.com/es/topics/cloud-native-apps/what-is-faas>
- [48] Daly Jeremy, 2020. The Simple Web Services. Serverless Reference Architectures. [online] Disponible en: <https://www.jeremydaly.com/simple-web-service/>
- [49] Daly Jeremy, 2020. Serverless Microservices Patterns for AWS. [online] Disponible en: <https://www.jeremydaly.com/serverless-microservice-patterns-for-aws/>
- [50] AWS, 2015. Arquitecturas multinivel sin servidor de AWS con API GATEWAY y AWS Lambda. [online] Disponible en: [http://d0.awsstatic.com/whitepapers/International/es/AWS\\_Serverless\\_Multi-Tier\\_Architectures\\_ES](http://d0.awsstatic.com/whitepapers/International/es/AWS_Serverless_Multi-Tier_Architectures_ES)
- [51] AWS, 2022. AWS Serverless multi-tier architectures with amazon API Gateway and AWS Lambda. Sample Architecture Patterns, web application. [online] Disponible en: <https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/web-application.html>
- [52] AWS, 2022. AWS Prescriptive Guidance. Patterns for integrating microservices, API Gateway patterns. [online] Disponible en: <https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-integrating-microservices/api-gateway-pattern.html>
- [53] AWS, 2022. API gateway pattern. [online] Disponible en: [https://docs.aws.amazon.com/es\\_es/prescriptive-guidance/latest/modernization-integrating-microservices/api-gateway-pattern.html](https://docs.aws.amazon.com/es_es/prescriptive-guidance/latest/modernization-integrating-microservices/api-gateway-pattern.html)
- [54] Gutiérrez-Hernández, R. E., Álvarez, F. J., & Muñoz-Arteaga, J, 2013. Arquitectura de Software para Juegos Serios con Aspectos Culturales: Caso de Estudio en un Videojuego para Fórmulas Temperatura.
- [55] Paez, Andy Hernández, Domínguez Falcón, Javier Alejandro, and Alejandro Andrés Pi Cruz, 2018. "Arquitectura de software para el desarrollo de videojuegos sobre el motor de juego Unity 3D." I+ D Tecnológico 14, no. 1, p. 54-65.

**[56]** Redhat, 2020. ¿Qué es la arquitectura orientada a los servicios SOA?. Diferencias entre la arquitectura orientada a servicios y a los microservicios. [online] Disponible en: <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture#diferencias-entre-la-soa-y-los-microservicios>

## ANEXOS

### 1. Documento de especificación de requisitos de la herramienta



Versión: 1.2

Erika Jissel Gutiérrez Beltrán

**Documento de Requisitos**

## Historia de revisiones

| Fecha      | Versión | Descripción   | Autor  |
|------------|---------|---|--|
| 17/11/2020 | 1.0     | Revisión de requisitos funcionales  | Erika Jissel Gutiérrez Beltrán                       |
| 11/12/2020 | 1.1     | Se añade listado de definición de requisitos  | Erika Jissel Gutiérrez Beltrán                       |
| 04/04/2021 | 1.2     | Validación de requisitos haciendo uso de prototipos, casos de uso y diagrama de actividades | Erika Jissel Gutiérrez Beltrán, Juan Carlos Martínez |

### 1. Introducción

Para llevar a cabo el desarrollo de una herramienta web que permita enseñar programación a niños y jóvenes, inicialmente se deben identificar las necesidades con las cuentan los usuarios. A partir de los elementos encontrados, se debe definir cada uno de los requisitos ya sean funcionales donde se describe lo que hará el sistema o su comportamiento, y requisitos no funcionales, donde se hace énfasis en el diseño y lineamientos para la implementación.

En el presente documento se plantean los requisitos funcionales y no funcionales con los que contará la herramienta web que finalmente indicarán el proceso de construcción, además se definirá el propósito, alcance y diseño de la herramienta a construir.

#### 1.1 Propósito

A través del siguiente documento, se definirán los puntos para lograr el desarrollo de una herramienta web que permita la enseñanza de la programación en niños y jóvenes. El público al que se dirige la herramienta es infantil, por esta razón es necesario tener en cuenta algunos criterios en el diseño de la interfaz para que esta sea llamativa como lo son el color, la forma y flujo de interacción. En este documento se describirán elementos de la interfaz como estructura y flujo, los cuales permitirán enriquecer la experiencia de usuario para construir una herramienta acorde al público objetivo y necesidades identificadas.

#### 1.2 Alcance

La herramienta web a construir, está pensada para ser utilizada por niños y jóvenes, por esta razón debe ser llamativa, rápida en tiempos de carga y fácil de usar. Además es necesario que se ajuste a los dispositivos tecnológicos con los que tienen mayor relación los usuarios. Deberá contar con una versión en móvil inicialmente desde un sitio web responsive y la versión tradicional de escritorio.

A nivel funcional la herramienta contará con un bloque de programación fundamental, donde se aprenderá sobre secuencias y diagramas de flujo. La aplicación web será accesible desde cualquier dispositivo, sin embargo, para la etapa inicial del desarrollo, prototipado y pruebas, las actividades a realizar estarán disponibles únicamente para computador.

La aplicación web se ejecutará bajo el navegador más utilizado, Google Chrome, sin descartar la posibilidad de que en un futuro pueda ser ejecutada en otros navegadores de uso no tan común. La plataforma contará con 2 perfiles, profesor y estudiantes, donde las vistas desde cada uno de los perfiles deben ser claramente distinguibles y las funcionalidades deben ser acordes a cada rol.

### **1.3 Definiciones, siglas y abreviaturas.**

**RQ** - Requisito funcional

**PP** - Página profesor

**PE** - Página estudiante

**RQN** - Requisito no funcional

**RSA** - Algoritmo de cifrado, el cual se basa en una llave pública y privada

**SSL** - Seguridad de la capa de transporte, protocolo para conexiones seguras

### **1.4 Visión general**

La herramienta web para la enseñanza de la programación en niños y jóvenes contará con diversos módulos de ejercicios o juegos enfocados en conceptos básicos, intermedios y avanzados de programación, que van desde secuencias, diagramas de flujo, algoritmos, pseudocódigo hasta código real.

El funcionamiento general de la herramienta es que niños de edades entre los 5 y 14 años puedan aprender a programar en una plataforma completa que fortalezca un conjunto de conocimientos asociados a la informática y computación. Además, se espera que los profesores puedan crear sus clases, asignar actividades a los estudiantes y hacer el seguimiento del avance de los niños y jóvenes, todo desde la herramienta.

## **2. Descripción general**

La herramienta web a construir está enfocada en enseñar a los niños y jóvenes programación, desde las etapas iniciales hasta código real, donde a diferencia de otras herramientas, se podrá aprender desde las bases hasta lo avanzado desde una misma, sin necesidad de cambiar constantemente de herramientas para aprender un concepto u otro.

Inicialmente se han obtenido una serie de necesidades, por medio de encuestas y entrevistas a los niños que han interactuado o no tienen experiencia con el uso de herramientas para aprender programación. Las características nombradas que más destacan son: 1) la necesidad de tener una

plataforma de libre acceso y accesible desde cualquier dispositivo; 2) posibilidad de interactuar sin conexión a internet; 3) fácil de usar; 4) variedad en los ejercicios y 5) selección de dificultad.

En este documento se describen uno a uno los elementos funcionales con los que contará la herramienta, sirviendo de apoyo los casos de uso, interfaces, definición de las características de los usuarios, restricciones del diseño y supuestos. Además de otorgar las indicaciones para cubrir las necesidades previamente expuestas.

Para lograr con éxito el desarrollo de la herramienta web para la enseñanza de la programación en niños y jóvenes, se debe identificar el alcance del proyecto y a su vez las limitaciones con las que este cuenta, para que al momento de la definición de los requisitos, estos estén asociados a objetivos alcanzables y posibles de realizar.

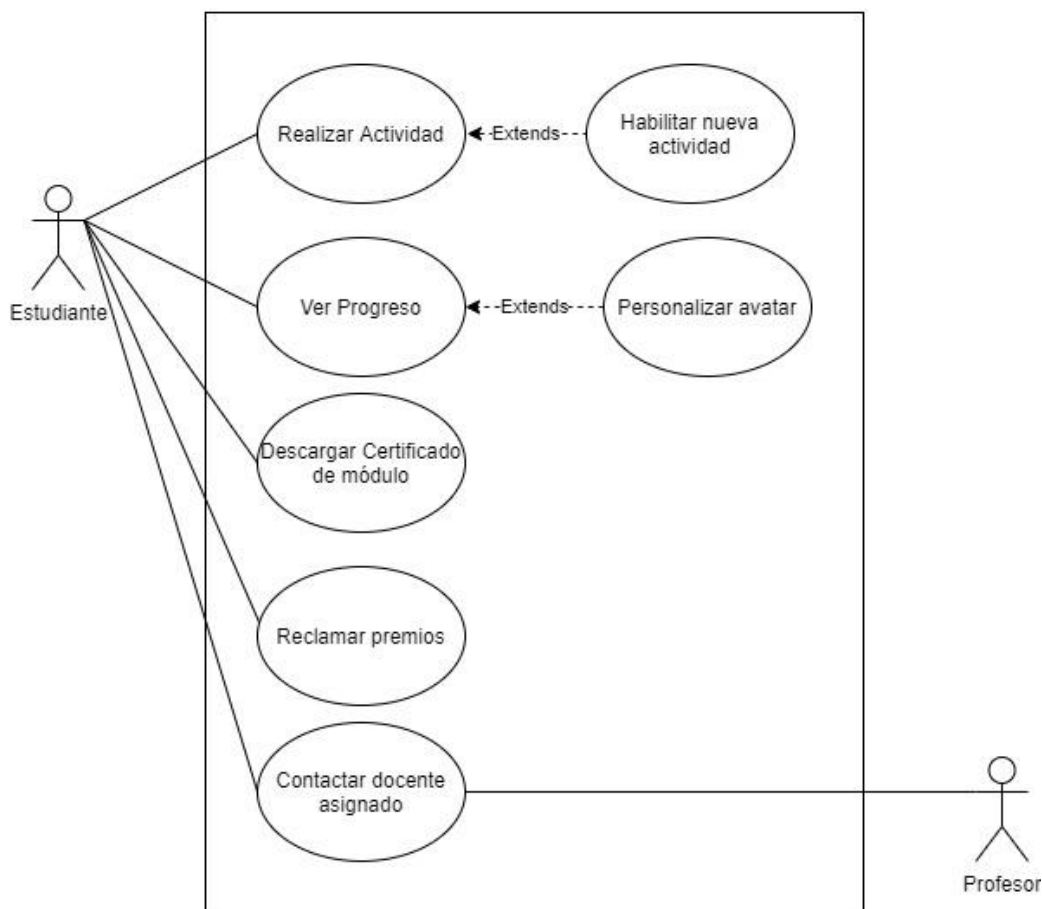
## **2.1 Perspectiva del producto**

Para representar los módulos con los que contará la herramienta y roles que deberán ser diferenciados no solo por la interfaz sino también por funcionalidades específicas, se proponen los siguientes casos de uso:

## Sistema - Vista Profesor



## Sistema - Vista Estudiante



### 2.2 Características de los usuarios

La herramienta web para la enseñanza de la programación, está dirigida a niños y jóvenes de edades entre los 5 y los 14 años, con capacidades básicas en lectura, escritura y matemáticas.

A continuación se describen tres perfiles de usuarios, dos de ellos de estudiantes de edad mínima y máxima, además del perfil del profesor, el cual debe poseer conocimientos en programación.

Se debe tener en cuenta, que la herramienta tiene un público objetivo, sin embargo, cualquier persona de una edad diferente a la esperada podrá hacer uso de la herramienta si así lo desea, solo que para lograr una construcción acorde a las necesidades obtenidas a partir del análisis previo, es necesario modelar haciendo uso de unos perfiles de usuario en particular.

Característica de los usuarios de edad mínima:

|                          |  |
|--------------------------|--|
| Edad mínima:             | 5 años   |
| Nivel educativo:         | Preescolar - Primer grado  |
| Experiencia:             | Uso o manejo de dispositivos móviles, principalmente la interacción se da con plataformas de streaming |
| Especialización técnica: | Fácil aprendizaje y dominio de las nuevas tecnologías  |

Característica de los usuarios de edad máxima:

|                          |   |
|--------------------------|---|
| Edad máxima esperada:    | 16 años   |
| Nivel educativo:         | Secundaria - Séptimo, octavo, noveno grado  |
| Experiencia:             | Uso o manejo de dispositivos móviles, principalmente la interacción se da con plataformas de streaming y redes sociales |
| Especialización técnica: | Fácil aprendizaje y dominio de las nuevas tecnologías   |

Característica de los usuarios con perfil profesor:

|                          |   |
|--------------------------|---|
| Tipo de perfil:          | Profesor con conocimientos en programación desde conceptos básicos hasta los más avanzados, para que sea posible llevar a cabo la transición de niveles fundamentales y básicos a programación avanzada |
| Nivel educativo:         | Persona de áreas relacionadas a la programación y tecnología, e ingeniero   |
| Experiencia:             | Manejo en herramientas para la enseñanza de la programación y manejo de lenguajes de programación actuales, principalmente en python y javascript   |
| Especialización técnica: | Manejo de lenguajes de programación python y javascript, además de frameworks y librerías para la web   |

### 2.3 Supuestos y dependencias

El desarrollo de la herramienta puede verse afectado por los cambios que puedan aparecer a nivel de construcción de las actividades a modo de juego que esta va a contener, las cuales hacen parte de los módulos de enseñanza. En relación a las tecnologías a seleccionar deben permitir la construcción para web y móvil al tiempo, lo cual optimizará tiempos en el desarrollo. Sin embargo, no tener en cuenta algunos criterios destacados como velocidad de internet con la que mayormente cuentan los niños y jóvenes en el país, podría afectar en el futuro el interés por parte de los usuarios si esta es lenta al cargar o depende 100% de internet para ser accedida.

A continuación se listan los posibles factores que pueden afectar el tiempo de creación de la herramienta y su estructura inicial, estos ítems están mayormente asociados a limitaciones técnicas.

- Dificultad de las actividades (juegos) para adaptarse a dispositivos móviles
- Las tecnologías seleccionadas no son aptas para cubrir las necesidades de la herramienta. Esta debería funcionar en un entorno tanto web como móvil.
- Incompatibilidad de librerías para integrar las actividades (juegos) en un entorno híbrido.
- Arquitectura de referencia seleccionada no óptima para lograr la solución esperada

### 3. Definición de requisitos

Previo a la construcción del sistema, se obtienen los requisitos que definen el funcionamiento de la plataforma. Estos se logran identificar y obtener por medio del análisis de plataformas ya existentes, entrevistas y encuestas realizadas a usuarios y consideraciones del autor. La definición de los requisitos obtenidos a través de los procesos anteriormente nombrados se describen a continuación.

| <b>Requisitos funcionales para el rol profesor</b> |   |
|--|---|
| <b>RQ-PP-001</b>                                   | La herramienta deberá contener un sistema de login por medio de correo electrónico / usuario y contraseña. La herramienta validará el correo o usuario ingresado. La contraseña deberá contener al menos 6 caracteres donde 1 elemento debe ser un carácter especial, una letra mayúscula y un número |
| <b>RQ-PP-002</b>                                   | La herramienta deberá contar con la opción de crear una nueva clase, permitiendo el ingreso del nombre de la clase, número de estudiantes, nivel y selección del módulo. La creación de la clase se hará por medio de un formulario.  |
| <b>RQ-PP-003</b>                                   | La herramienta deberá permitir la visualización de las clases creadas por un profesor, mostrando el nombre de la clase, número de   |

|                  |   |
|------------------|---|
|                  | <p>estudiantes asignados, nivel y url de acceso. Además cada registro de la tabla donde se mostrarán las clases deberá tener las opciones de añadir estudiantes, edición de una clase, eliminar una clase y visualización de preguntas. La clase solo podrá ser eliminada si no tiene estudiantes asignados o si los estudiantes de una clase no han realizado ningún ejercicio.</p>                                |
| <b>RQ-PP-004</b> | <p>La herramienta deberá contar con una pantalla de creación de estudiantes, donde por medio de un formulario se solicitará nombre, apellido, nombre de usuario, correo electrónico y contraseña. El estudiante creado será asignado a una clase en particular y solo podrá pertenecer a una clase a la vez. Sin embargo, el estudiante podrá moverse a otra clase si lo autoriza desde el sistema el profesor.</p> |
| <b>RQ-PP-005</b> | <p>La herramienta deberá contar con una pantalla de creación de estudiantes, donde sea posible cargar un archivo de extensión .csv, el cual contenga la información de nombre de usuario, nombre, apellido, contraseña, correo electrónico, niveles y módulo al que será asignado. Esto debería crear automáticamente los usuarios en la herramienta.</p>   |
| <b>RQ-PP-006</b> | <p>La herramienta deberá permitir la visualización de los estudiantes de una clase, mostrando el nombre completo, nombre de usuario y correo electrónico. Además deberá contar con las acciones de editar, eliminar y contactar por chat. Un estudiante sólo podrá ser eliminado si no ha realizado ninguna actividad o ejercicio.</p>  |
| <b>RQ-PP-007</b> | <p>La herramienta deberá contar con una sección de recursos de apoyo al profesor, donde se explique el uso de la herramienta y se cuente con tutoriales y textos explicativos de los ejercicios por nivel, con el fin de que el profesor pueda estar capacitado.</p>  |
| <b>RQ-PP-008</b> | <p>La herramienta deberá contar con una sección de ayuda y soporte donde el profesor pueda dejar registro de los problemas encontrados, además pueda chatear con una persona de soporte técnico, para agilizar el proceso.</p>  |
| <b>RQ-PP-009</b> | <p>La herramienta deberá permitir al usuario contar con una sección de cursos, simulando un rol de estudiante. Podrá realizar todos los</p>   |

|  |   |
|--|---|
|  | ejercicios, lo cual facilitará la comprensión de los mismos y a su vez de la herramienta.   |
| <b>RQ-PP-010</b>                                     | La herramienta deberá contar con una pantalla de visualización de las preguntas dejadas sobre ejercicios por parte de los estudiantes.  |
| <b>RQ-PP-011</b>                                     | La herramienta deberá contar con un formulario de respuesta, para dar solución a las inquietudes expresadas por parte de los estudiantes  |
| <b>RQ-PP-012</b>                                     | La herramienta deberá permitir al usuario con rol de profesor contactar un estudiante a través de chat  |
| <b>Requisitos funcionales para el rol estudiante</b> |   |
| <b>RQ-PE-001</b>                                     | La herramienta deberá permitir al estudiante loguearse en la plataforma a través del nombre de usuario y contraseña. La contraseña debe ser mínimo de 6 caracteres, contener una mayúscula y al menos un número.  |
| <b>RQ-PE-002</b>                                     | La herramienta deberá permitir al estudiante visualizar a través de una interfaz los módulos asignados por el profesor.   |
| <b>RQ-PE-003</b>                                     | La herramienta deberá permitir desde la pantalla de clases, seleccionar la opción de ayuda que redireccionará a una nueva pantalla con un formulario y chat asociado al módulo seleccionado. Esto con el fin de que el usuario pueda realizar preguntas y resolver dudas de las actividades correspondientes al nivel y módulo. |
| <b>RQ-PE-004</b>                                     | La herramienta deberá mostrar en la pantalla de cursos el progreso de cada uno y el puntaje adquirido.  |
| <b>RQ-PE-005</b>                                     | La herramienta deberá mostrar al usuario con rol de estudiante un mapa de selección de las actividades a realizar en modo de juego, habilitando solo un nivel si el realizado previamente fue superado con éxito.   |
| <b>RQ-PE-006</b>                                     | La herramienta deberá contar con un espacio de juego, para que el usuario pueda realizar el nivel habilitado y desbloqueado. La descripción de los juegos del módulo se describen en el documento de diseño de juego.   |

|                  |  |
|------------------|--|
| <b>RQ-PE-007</b> | La herramienta deberá mostrar una ventana flotante de confirmación, en caso de que el estudiante decida cambiar la vista de juego por error, o desee cambiar de pantalla. Esto se debe realizar con el fin de evitar perder el progreso del nivel. Aplica solo cuando se este en la pantalla de juego.   |
| <b>RQ-PE-008</b> | La herramienta deberá guardar el avance y progreso del estudiante en cada nivel que realice con éxito. La herramienta contará criterios de evaluación, dados por una solución de base que estará definida en la herramienta. Esta contará con unos criterios de evaluación que son: revisión de la complejidad y uso de los elementos solicitados en la actividad. |
| <b>RQ-PE-009</b> | La herramienta deberá permitir al estudiante por medio de una pantalla de ayuda contactar con el profesor de cada curso. Para esto contará con una ventana de chat y un formulario, donde el estudiante podrá ingresar una pregunta.   |

#### **4. Especificación de Requisitos**

En esta sección se describen los requisitos funcionales y no funcionales, los cuales servirán de apoyo para llevar a cabo la construcción de la herramienta a consideración de las necesidades del usuario.

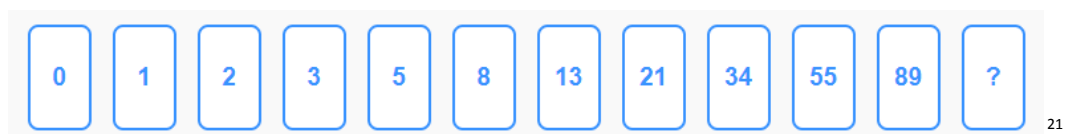
Los requisitos se describen a partir del análisis realizado a las herramientas ya existentes, encuestas y entrevistas a niños y jóvenes con experiencia y sin experiencia en este tipo de herramientas. Los resultados obtenidos se adjuntan al final de este documento.

##### **4.1. Requisitos funcionales**

Los requisitos funcionales, describirán paso a paso que deberá contener la herramienta web para la enseñanza de la programación, para así cubrir las necesidades que se presentan a nivel de ejercicios básicos de lógica computacional, fácil interacción, posibilidad de avanzar de nivel y actividades con código real avanzado.

Los requisitos funcionales se dividirán en el bloque de profesor y en el bloque estudiante, ya que las funcionalidades deben ser distintas, por lo tanto las pantallas deberán verse acorde a lo que se describe.

Cada requisito, contará con un esfuerzo asignado teniendo en cuenta la metodología scrum. Este esfuerzo no corresponde a horas, sino a dificultad en su desarrollo. Para la selección del sprint el cual se propone al final de este documento, se priorizan las actividades por prioridad y esfuerzo.



21

### Requisitos funcionales para bloque profesor

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PP-001  |
| Nombre del requisito                | Login perfil profesor  |
| Descripción detallada del requisito | La herramienta deberá contener un sistema de login por medio de correo electrónico / usuario y contraseña. La herramienta validará el correo o usuario ingresado. La contraseña deberá contener al menos 8 caracteres donde 1 elemento debe ser un carácter especial y otro una letra mayúscula. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020  |
| Fuente del requisito                | Propio del autor   |
| Versión del requisito               | 1.0.1  |
| Dependencia del requisito           | Este requisito no depende de otros   |
| Esfuerzo                            | 8  |
| Prioridad                           | Alta   |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PP-002  |
| Nombre del requisito                | Creación de una clase  |
| Descripción detallada del requisito | La herramienta deberá contar con la opción de crear una nueva clase, permitiendo el ingreso del nombre de la clase, número de estudiantes, nivel y selección del módulo. La creación de la clase se hará por medio de un formulario. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020  |

<sup>21</sup> Estimación de esfuerzo a través de la técnica planning poker : <https://planningpokeronline.com/>

|                           |  |
|---------------------------|--|
| Fuente del requisito      | Propio del autor   |
| Versión del requisito     | 1.0.1  |
| Dependencia del requisito | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción |
| Esfuerzo                  | 13   |
| Prioridad                 | Alta   |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PP-003   |
| Nombre del requisito                | Visualización de las clases y eventos de acción   |
| Descripción detallada del requisito | La herramienta deberá permitir la visualización de las clases creadas por un profesor, mostrando el nombre de la clase, número de estudiantes asignados, nivel y url de acceso. Además cada registro de la tabla donde se mostrarán las clases deberá tener las opciones de añadir estudiantes, edición de una clase, eliminar una clase y visualización de preguntas. La clase solo podrá ser eliminada si no tiene estudiantes asignados o si los estudiantes de una clase no han realizado ningún ejercicio. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020   |
| Fuente del requisito                | Propio del autor  |
| Versión del requisito               | 1.0.1   |
| Dependencia del requisito           | REQ-PP-002, es necesario tener clases creadas para mostrarlas en la pantalla de visualización   |
| Esfuerzo                            | 8   |
| Prioridad                           | Alta  |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PP-004   |
| Nombre del requisito                | Creación de estudiantes   |
| Descripción detallada del requisito | La herramienta deberá contar con una pantalla de creación de estudiantes, donde por medio de un formulario se solicitará nombre, apellido, nombre de usuario, correo electrónico y contraseña. El estudiante creado será asignado a una clase en particular y solo podrá pertenecer a una clase a la vez. Sin embargo, el |

|                                |  |
|--------------------------------|--|
|                                | estudiante podrá moverse a otra clase si lo autoriza desde el sistema el profesor. |
| Fecha de creación de requisito | 24 - Noviembre - 2020  |
| Fuente del requisito           | Propia del autor   |
| Versión del requisito          | 1.0.1  |
| Dependencia del requisito      | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción       |
| Esfuerzo                       | 13   |
| Prioridad                      | Alta   |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PP-005  |
| Nombre del requisito                | Creación de estudiantes - automático   |
| Descripción detallada del requisito | La herramienta deberá contar con una pantalla de creación de estudiantes, donde sea posible cargar un archivo de extensión .csv, el cual contenga la información de nombre de usuario, nombre, apellido, contraseña, correo electrónico, niveles y módulo al que será asignado. Esto debería crear automáticamente los usuarios en la herramienta. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020  |
| Fuente del requisito                | Propia del autor   |
| Versión del requisito               | 1.0.1  |
| Dependencia del requisito           | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción   |
| Esfuerzo                            | 13   |
| Prioridad                           | Alta   |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PP-006   |
| Nombre del requisito                | Visualización de estudiantes  |
| Descripción detallada del requisito | La herramienta deberá permitir la visualización de los estudiantes de una clase, mostrando el nombre completo, nombre de usuario y correo electrónico. Además deberá contar con las |

|                                |   |
|--------------------------------|---|
|                                | acciones de editar, eliminar y contactar por chat. Un estudiante sólo podrá ser eliminado si no ha realizado ninguna actividad o ejercicio. |
| Fecha de creación de requisito | 24 - Noviembre - 2020   |
| Fuente del requisito           | Propia del autor  |
| Versión del requisito          | 1.0.0   |
| Dependencia del requisito      | RP-PP-004 / RP-PP-005, ya que se deben crear los estudiantes para mostrarlos en la pantalla de visualización                                |
| Esfuerzo                       | 5   |
| Prioridad                      | Alta  |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PP-007   |
| Nombre del requisito                | Recursos de apoyo al profesor   |
| Descripción detallada del requisito | La herramienta deberá contar con una sección de recursos de apoyo al profesor, donde se explique el uso de la herramienta y se cuente con tutoriales y textos explicativos de los ejercicios por nivel, con el fin de que el profesor pueda estar capacitado. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020   |
| Fuente del requisito                | Resultado análisis previo de otras herramientas   |
| Versión del requisito               | 1.0.1   |
| Dependencia del requisito           | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción  |
| Esfuerzo                            | 13  |
| Prioridad                           | Baja  |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PP-008  |
| Nombre del requisito                | Recursos de ayuda y soporte técnico  |
| Descripción detallada del requisito | La herramienta deberá contar con una sección de ayuda y soporte donde el profesor pueda dejar registro de los problemas encontrados, además pueda chatear con una persona de |

|                                |  |
|--------------------------------|--|
|                                | soporte técnico, para agilizar el proceso.                                   |
| Fecha de creación de requisito | 24 - Noviembre - 2020  |
| Fuente del requisito           | Resultado análisis previo de otras herramientas                              |
| Versión del requisito          | 1.0.1  |
| Dependencia del requisito      | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción |
| Esfuerzo                       | 21   |
| Prioridad                      | Baja   |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PP-009  |
| Nombre del requisito                | Visualización de cursos y actividades  |
| Descripción detallada del requisito | La herramienta deberá permitir al usuario contar con una sección de cursos, simulando un rol de estudiante. Podrá realizar todos los ejercicios, lo cual facilitará la comprensión de los mismos y a su vez de la herramienta. |
| Fecha de creación de requisito      | 08 - Abril - 2021  |
| Fuente del requisito                | Propio del autor   |
| Versión del requisito               | 1.0.0  |
| Dependencia del requisito           | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción   |
| Esfuerzo                            | 13   |
| Prioridad                           | Media  |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PP-010  |
| Nombre del requisito                | Pantalla de ayuda al estudiante  |
| Descripción detallada del requisito | La herramienta deberá contar con una pantalla de visualización de las preguntas dejadas sobre ejercicios por parte de los estudiantes. |
| Fecha de creación de requisito      | 08 - Abril - 2021  |
| Fuente del requisito                | Resultado análisis previo de otras herramientas  |

|                           |   |
|---------------------------|---|
| Versión del requisito     | 1.0.0   |
| Dependencia del requisito | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción. |
| Esfuerzo                  | 13  |
| Prioridad                 | Alta  |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PP-011   |
| Nombre del requisito                | Solución de problemas de los estudiantes  |
| Descripción detallada del requisito | La herramienta deberá contar con un formulario de respuesta, para dar solución a las inquietudes expresadas por parte de los estudiantes. |
| Fecha de creación de requisito      | 08 - Abril - 2021   |
| Fuente del requisito                | Resultado análisis previo de otras herramientas   |
| Versión del requisito               | 1.0.0   |
| Dependencia del requisito           | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción  |
| Esfuerzo                            | 13  |
| Prioridad                           | Alta  |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PP-012   |
| Nombre del requisito                | Chat de contacto rápido al estudiante   |
| Descripción detallada del requisito | La herramienta deberá permitir al usuario con rol de profesor contactar un estudiante a través de chat. |
| Fecha de creación de requisito      | 08 - Abril - 2021   |
| Fuente del requisito                | Resultado análisis previo de otras herramientas   |
| Versión del requisito               | 1.0.0   |
| Dependencia del requisito           | RQ-PP-001, es necesario loguearse para que sea posible acceder a esta opción                            |
| Esfuerzo                            | 21  |
| Prioridad                           | Media   |

## Requisitos funcionales para bloque estudiante

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PE-001  |
| Nombre del requisito                | Login perfil estudiante  |
| Descripción detallada del requisito | La herramienta deberá permitir al estudiante loguearse en la plataforma a través del nombre de usuario y contraseña. La contraseña debe ser mínimo de 6 caracteres, contener una mayúscula y al menos un número. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020  |
| Fuente del requisito                | Propio del autor   |
| Versión del requisito               | 1.0.1  |
| Dependencia del requisito           | Este requisito no depende de otros   |
| Esfuerzo                            | 13   |
| Prioridad                           | Alta   |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PE-002   |
| Nombre del requisito                | Opción visualización cursos asignados   |
| Descripción detallada del requisito | La herramienta deberá permitir al estudiante visualizar a través de una interfaz los módulos asignados por el profesor. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020   |
| Fuente del requisito                | Propio del autor  |
| Versión del requisito               | 1.0.1   |
| Dependencia del requisito           | RQ-PE-001, es necesario loguearse para que sea posible acceder a esta opción  |
| Esfuerzo                            | 13  |
| Prioridad                           | Alta  |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PE-003                               |
| Nombre del requisito                | Pantalla de ayuda asociada al curso     |
| Descripción detallada del requisito | La herramienta deberá permitir desde la |

|                                |   |
|--------------------------------|---|
|                                | pantalla de clases, seleccionar la opción de ayuda que redireccionará a una nueva pantalla con un formulario y chat asociado al módulo seleccionado. Esto con el fin de que el usuario pueda realizar preguntas y resolver dudas de las actividades correspondientes al nivel y módulo. |
| Fecha de creación de requisito | 08 - Abril - 2021   |
| Fuente del requisito           | Análisis previo de otras herramientas   |
| Versión del requisito          | 1.0.0   |
| Dependencia del requisito      | RQ-PP-002, Es necesario que el profesor haya creado una clase previamente, con un curso asignado  |
|                                | 13  |
| Prioridad                      | Media   |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PE-004  |
| Nombre del requisito                | Progreso del usuario por curso   |
| Descripción detallada del requisito | La herramienta deberá mostrar en la pantalla de clases el progreso de cada uno y el puntaje adquirido. |
| Fecha de creación de requisito      | 08 - Abril- 2021   |
| Fuente del requisito                | Análisis previo de otras herramientas  |
| Versión del requisito               | 1.0.0  |
| Dependencia del requisito           | RQ-PE-002, Opción visualización cursos asignados   |
| Esfuerzo                            | 21   |
| Prioridad                           | Media  |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PE-005   |
| Nombre del requisito                | Selección de la actividad   |
| Descripción detallada del requisito | La herramienta deberá mostrar al usuario con rol de estudiante un mapa de selección de las actividades a realizar en modo de juego, habilitando solo un nivel si el realizado previamente fue superado con éxito. |

|                                |  |
|--------------------------------|--|
| Fecha de creación de requisito | 08 - Abril- 2021   |
| Fuente del requisito           | Stakeholders, niños y jóvenes de edades entre los 5 y 14 años                                    |
| Versión del requisito          | 1.0.0  |
| Dependencia del requisito      | RQ-PP-002, Es necesario que el profesor haya creado una clase previamente, con un curso asignado |
| Esfuerzo                       | 34   |
| Prioridad                      | Alta   |

|                                     |   |
|-------------------------------------|---|
| Código del requisito                | RQ-PE-006   |
| Nombre del requisito                | Espacio dedicado a la pantalla de juego   |
| Descripción detallada del requisito | La herramienta deberá contar con un espacio de juego, para que el usuario pueda realizar el nivel habilitado y desbloqueado. La descripción de los juegos del módulo se describen en el documento de diseño de juego. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020   |
| Fuente del requisito                | Propia del autor  |
| Versión del requisito               | 1.0.1   |
| Dependencia del requisito           | RQ-PE-005, Selección de la actividad  |
| Esfuerzo                            | 13  |
| Prioridad                           | Alta  |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PE-007  |
| Nombre del requisito                | Ventana de confirmación para salir del juego   |
| Descripción detallada del requisito | La herramienta deberá mostrar una ventana flotante de confirmación, en caso de que el estudiante decida cambiar la vista de juego por error, o desee cambiar de pantalla. Esto se debe realizar con el fin de evitar perder el progreso del nivel. Aplica solo cuando se este en la pantalla de juego. |
| Fecha de creación de requisito      | 24 - Noviembre - 2020  |

|                           |   |
|---------------------------|---|
| Fuente del requisito      | Propia del autor  |
| Versión del requisito     | 1.0.1   |
| Dependencia del requisito | RQ-PE-005, Selección de la actividad<br>RQ-PE-001, login perfil de estudiante |
| Esfuerzo                  | 5   |
| Prioridad                 | Alta  |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PE-008  |
| Nombre del requisito                | Validación de las actividades realizadas   |
| Descripción detallada del requisito | La herramienta deberá guardar el avance y progreso del estudiante en cada nivel que realice con éxito. Debido a que las actividades son de lógica y programación, cada estudiante podrá realizar una solución a partir de lo que considere, sin embargo, la herramienta validará a partir de la solución más óptima. Por cada nivel se entregarán puntos. Si se finaliza cada nivel con la lógica y estructura esperada se entregarán todos los puntos del nivel, de lo contrario se disminuirán los puntos según el tipo de solución. |
| Fecha de creación de requisito      | 08 - Abril- 2021   |
| Fuente del requisito                | Stakeholders y resultado análisis previo de otras herramientas   |
| Versión del requisito               | 1.0.0  |
| Dependencia del requisito           | RQ-PE-005, Selección de la actividad<br>RQ-PE-001, login perfil de estudiante  |
| Esfuerzo                            | 55   |
| Prioridad                           | Alta   |

|                                     |  |
|-------------------------------------|--|
| Código del requisito                | RQ-PE-009  |
| Nombre del requisito                | Pantalla de ayuda y solución de dudas  |
| Descripción detallada del requisito | La herramienta deberá permitir al estudiante por medio de una pantalla de ayuda contactar con el profesor de cada curso. Para esto contará con una ventana de chat y un formulario, donde el estudiante podrá ingresar una pregunta. |

|                                |  |
|--------------------------------|--|
| Fecha de creación de requisito | 08 - Abril- 2021   |
| Fuente del requisito           | Stakeholders y resultado análisis previo de otras herramientas |
| Versión del requisito          | 1.0.0  |
| Dependencia del requisito      | RQ-PE-001, login perfil de estudiante                          |
| Esfuerzo                       | 21   |
| Prioridad                      | Media  |

#### 4.2. Restricciones de diseño

Teniendo en cuenta que la plataforma a construir será web, se deben considerar las siguientes restricciones, antes de establecer los requisitos no funcionales que deberá tener. Las limitaciones de diseño de la plataforma web para la enseñanza de la programación se describen a continuación.

- La plataforma inicialmente se desarrollará para ser ejecutada en un ambiente web, sin embargo, la tecnología a considerar debe permitir la generación de una aplicación híbrida funcional en dispositivos móviles, principalmente celulares.
- Para la etapa inicial de lanzamiento de la plataforma, esta deberá funcionar en navegadores de uso común como lo es Google Chrome.
- Las actividades a desarrollar deben construirse bajo una tecnología soportada por navegadores y posteriormente en dispositivos móviles.

#### 4.3. Requisitos No funcionales

| Código del requisito | Descripción del requisito  |
|----------------------|--|
| RQ-NF-001            | <b>Usabilidad:</b> El tiempo de aprendizaje de la herramienta por parte del usuario (estudiante - profesor) no deberá ser mayor a una hora. La herramienta debe ser lo suficientemente intuitiva con el fin de que el usuario pueda desenvolverse sin dificultades al ejecutar acciones. |
| RQ-NF-002            | <b>Usabilidad:</b> La herramienta deberá contar con mensajes de error claros, los cuales le indiquen al usuario porque se ha generado, además de que pasos debería seguir para que no siga apareciendo el mensaje en pantalla.   |
| RQ-NF-003            | <b>Usabilidad:</b> La herramienta deberá contar con adaptación de la web a   |

|           |  |
|-----------|--|
|           | dispositivos móviles, ya que gran cantidad de usuarios navegan principalmente en internet desde un celular.  |
| RQ-NF-004 | <b>Disponibilidad:</b> La herramienta deberá estar en línea 24/7, ya que el usuario podrá acceder a la plataforma en cualquier momento, desde cualquier lugar, siempre y cuando tenga internet.  |
| RQ-NF-005 | <b>Concurrencia:</b> La herramienta deberá estar en la capacidad de permitir inicialmente al menos 100 usuarios simultáneos, interactuando no sólo con las actividades de juego, sino también en la asignación y creación de clases, o revisión de video tutoriales. |
| RQ-NF-006 | <b>Eficiencia:</b> Los datos creados, actualizados o eliminados de la herramienta, deben verse reflejados en menos de 1 segundo en el sistema.   |
| RQ-NF-007 | <b>Seguridad:</b> Los permisos al sistema (servidores), deberán ser otorgados y editados únicamente por el súper administrador.  |
| RQ-NF-008 | <b>Seguridad:</b> El sistema deberá contar con certificado SSL, para que el usuario pueda navegar seguramente en la plataforma web.  |
| RQ-NF-009 | <b>Seguridad:</b> La comunicación entre la aplicación y los servicios debe hacer uso Web Tokens o en su defecto RSA, para evitar la vulneración y consumo por parte de externos.   |
| RQ-NF-010 | <b>Mantenibilidad:</b> La herramienta deberá soportar despliegues continuos sin afectar el servicio y siendo imperceptibles para el usuario.   |
| RQ-NF-011 | <b>Confiabilidad:</b> La herramienta debe garantizar al usuario que los datos ingresados son persistentes y el avance en los niveles por parte del estudiante se almacenará correctamente.   |

## 5. Interfaces de usuario

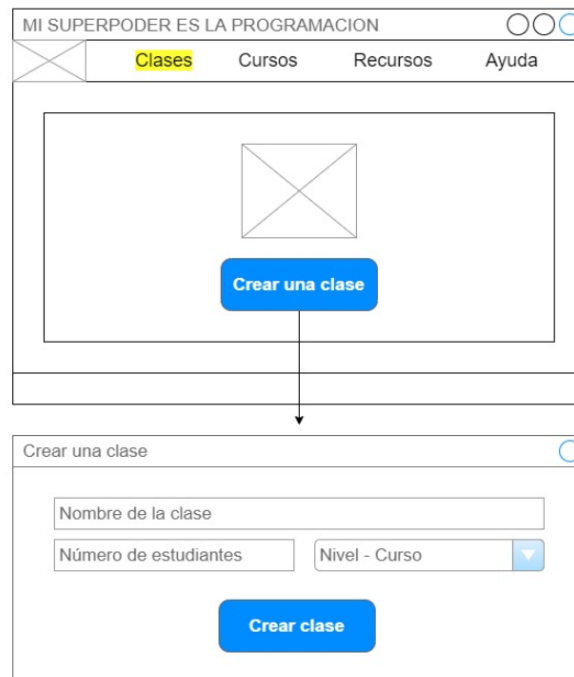
Teniendo en cuenta la definición de los requisitos, se establecen las interfaces que servirán como prototipo inicial.

### Interfaces - vista profesor

#### Pantalla 1: Login



#### Pantalla 2: Creación clase - aula



**Pantalla 3:** Listado de clases creadas

| Nombre Clase | # Students | Nivel | Enlace  | Acciones   |
|--------------|------------|-------|---------|--|
| Lorem Ipsum  | 1          | F1    | http:// | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |
| Lorem Ipsum  | 1          | F1    | http:// | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |
| Lorem Ipsum  | 1          | F1    | http:// | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |

**Pantalla 4:** Eliminar y modificar clases

Eliminar clase

¿Estás seguro que deseas eliminar la clase  
NOMBRE CLASE?

**Eliminar**

Editar clase

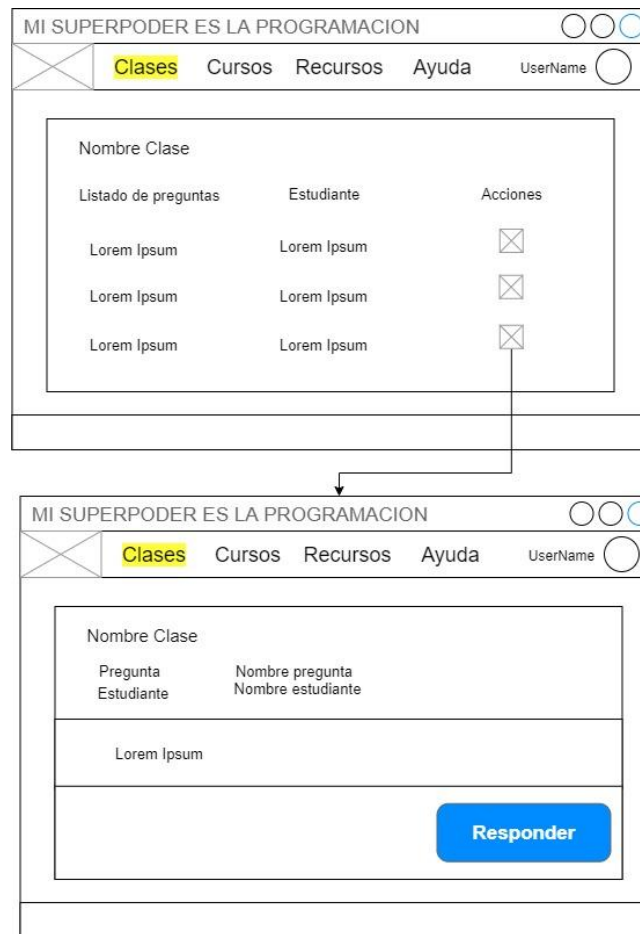
Nombre clase

# estudiantes

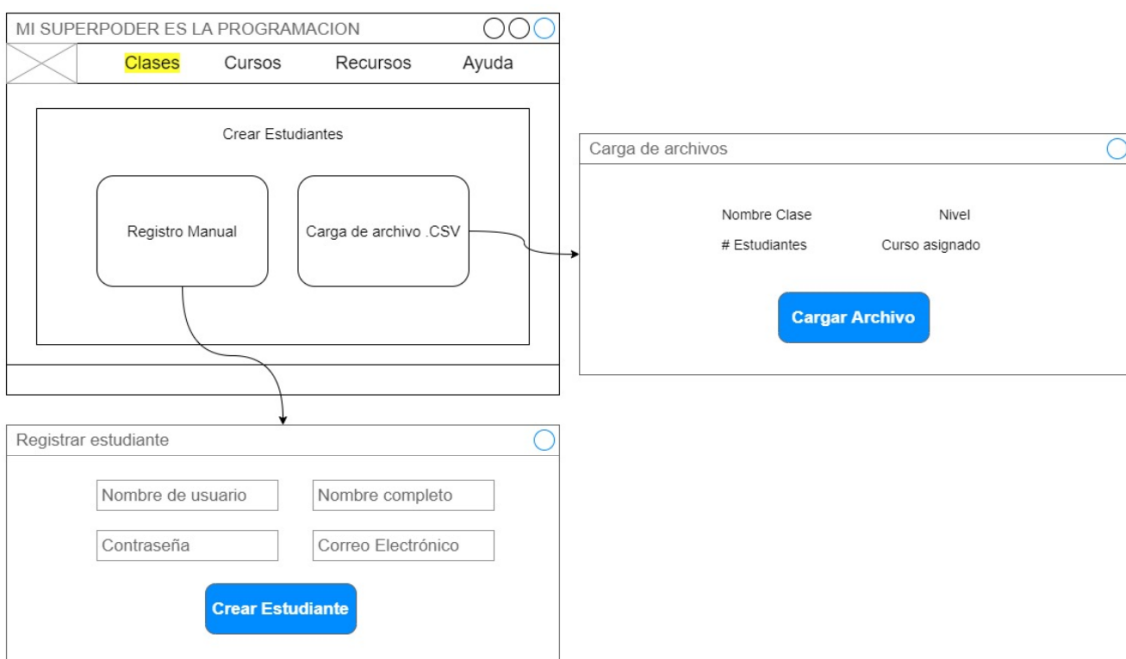
Nivel ▾

**Editar clase**

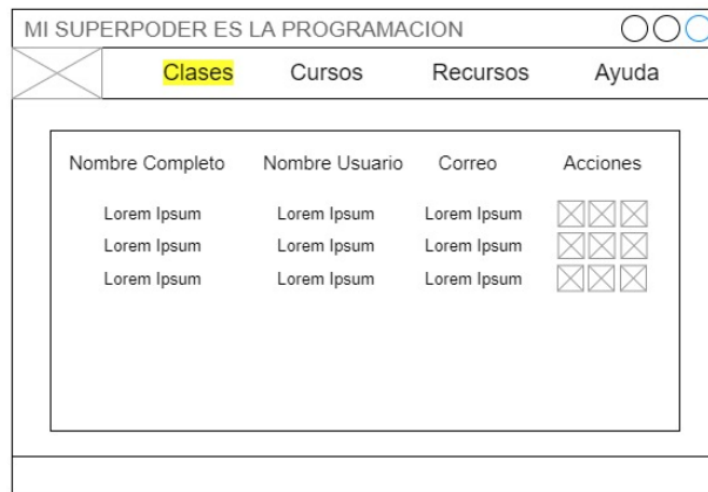
### Pantalla 5: Preguntas de la clase y sección de respuestas



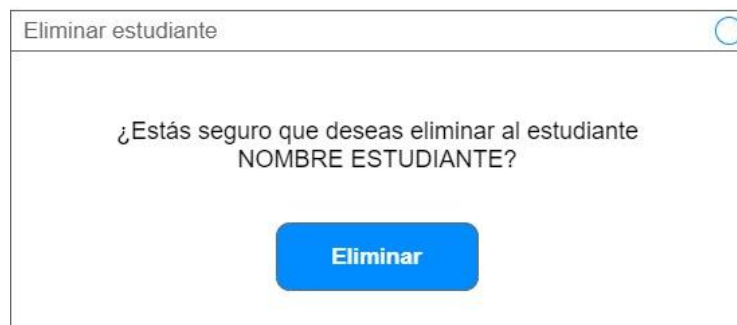
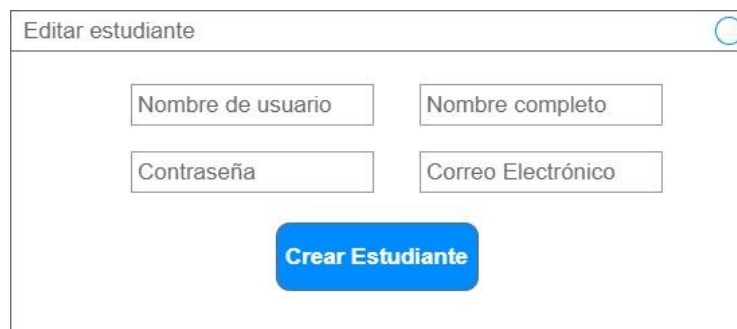
### Pantalla 6: Creación y registro de estudiantes a una clase



**Pantalla 7:** Visualización listado de estudiantes creados para una clase



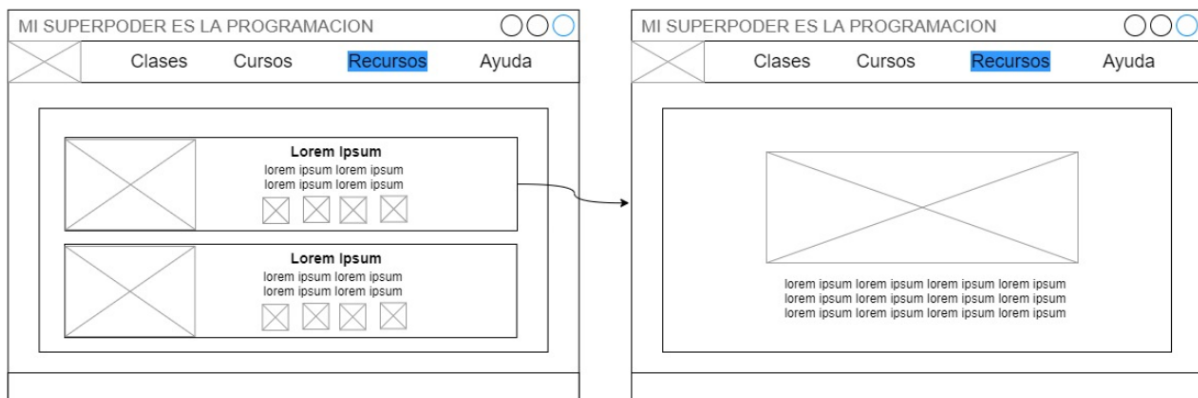
**Pantalla 8:** Modificar y eliminar un estudiante



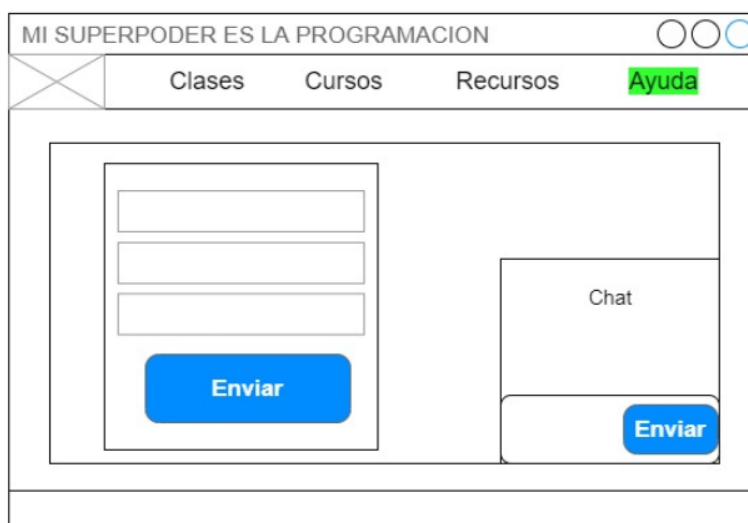
**Pantalla 9:** Visualización de cursos disponibles para asignar a una clase según el nivel



**Pantalla 10:** Recursos de ayuda, como tutoriales y documentación



**Pantalla 11:** Ventana con formulario y chat para contacto con soporte y registrar problemas de la plataforma

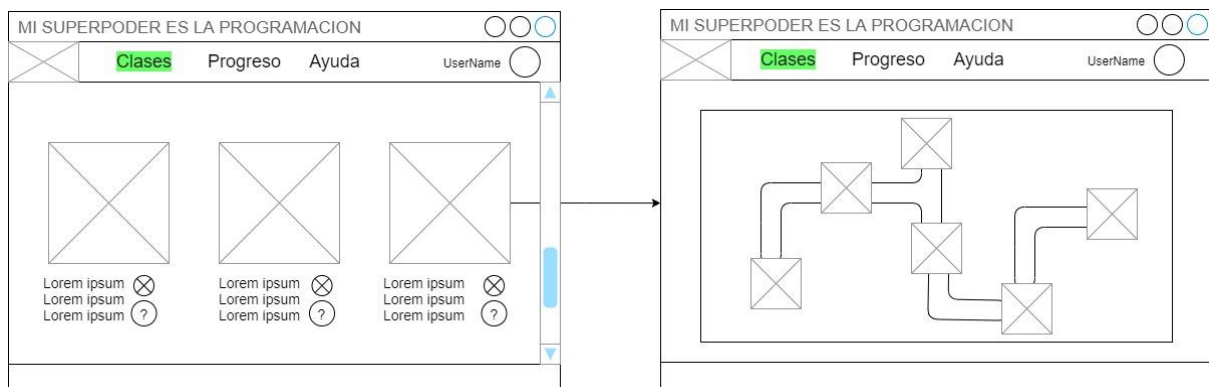


## Interfaz - vista estudiante

**Pantalla 1 - 2:** Inicio al sistema, visualización de compañeros de clase para contactarlo a través de chat, o reportar en caso de algún inconveniente presentado.



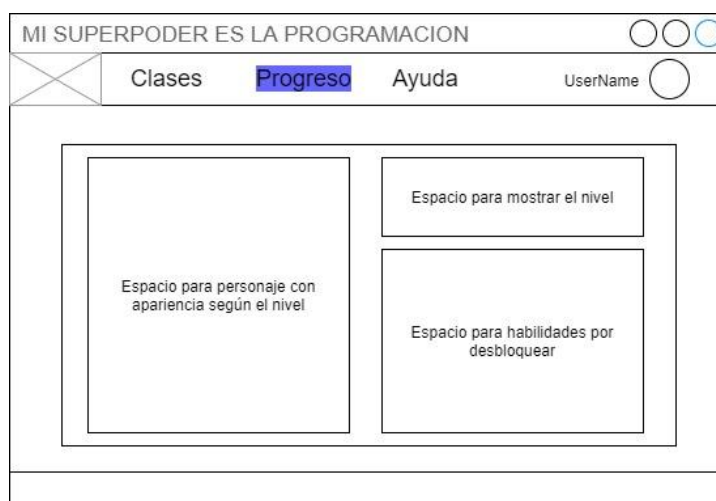
**Pantalla 3 - 4:** Visualización de cursos asignados - Interfaz de actividades disponibles a realizar según el curso seleccionado



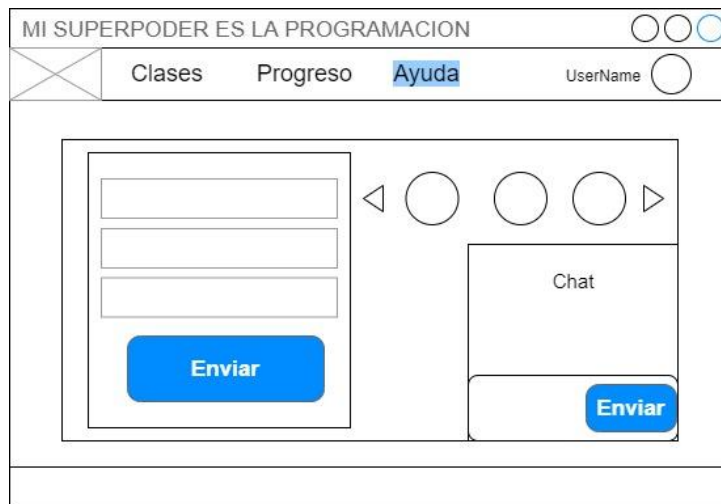
**Pantalla 5:** Interfaz de juego, el usuario podrá desempeñar sus actividades para aprender a programar. A su vez se muestra el modal de confirmación antes de salir del juego



**Pantalla 6:** Visualización del progreso del estudiante, selección de ítems para el personaje, desbloqueo de habilidades según los puntos alcanzados y el nivel



**Pantalla 7:** Ventana de soporte, por este medio el estudiante se contactará con el profesor y dejará preguntas de la clase



## 2. Game Design Document - Documento de diseño de juego



Versión: 1.5

Erika Jissel Gutiérrez Beltrán

**Game Design Documentation**

## Tabla de Contenido

### **Diseño de la Historia**

|             |   |
|-------------|---|
| Versión 1.0 | 4 |
| Versión 1.1 | 4 |
| Versión 1.2 | 4 |
| Versión 1.4 | 4 |
| Versión 1.5 | 5 |

### **Visión General**

|                 |   |
|-----------------|---|
| Introducción    | 6 |
| Objetivo        | 6 |
| Género          | 6 |
| Características | 6 |
| Plataforma      | 7 |
| Audiencia       | 7 |
| Lenguaje        | 7 |
| Sonidos         | 7 |

### **Historia**

|                      |    |
|----------------------|----|
| Outline              | 8  |
| Historia - Sinopsis  | 8  |
| Escenas              | 9  |
| Tamaño del juego     | 11 |
| Referencias visuales | 11 |

### **Mecánicas de juego**

|           |    |
|-----------|----|
| Cámara    | 16 |
| Jugadores | 16 |

|                            |           |
|----------------------------|-----------|
| Movimientos generales      | 16        |
| Configuración inicial      | 17        |
| Salud                      | 17        |
| Recompensas                | 17        |
| Obstáculos                 | 17        |
| <b>Personajes</b>          | <b>18</b> |
| <b>Gameplay detallado</b>  | <b>20</b> |
| <b>Concept art</b>         | <b>21</b> |
| <b>Desarrollos futuros</b> | <b>23</b> |

## **Diseño de la Historia**

### **Versión 1.0**

Diciembre 12, 2020

En la versión 1.0 se crea el Game Design Document (GDD). En esta primera versión se define la historia, mecánicas de juego, personajes, puntuación, salud, movimiento del personaje y habilidades.

### **Versión 1.1**

Marzo 11, 2021

En la versión 1.1 se modifican algunas características iniciales de los personajes, como nombre, personalidad y forma. Además se terminan elementos pendientes a nivel de la historia, definición de las mecánicas y escenarios por cada uno de los niveles iniciales (5).

### **Versión 1.2**

Marzo 25, 2021

En la versión 1.2 se agregan bocetos de referencia de los módulos 1 y 2, donde se indica lo que se quiere visualizar según el objetivo de cada nivel.

### **Versión 1.3**

Mayo 2, 2021

En la versión 1.3 se agregan sprints de los personajes, adicionalmente se hace un cambio a nivel de mecánica de los niveles iniciales y de la historia.

### **Versión 1.4**

Septiembre 9, 2021

En la versión 1.4 se refinan los niveles, se ajusta el personaje de violeta y se agrega el concept art del videojuego

## **Versión 1.5**

Septiembre 9, 2022

En la versión 1.5 se refinan los niveles, además de que se detallan de mejor manera los comportamientos del personaje en el videojuego.

# Visión General

## Introducción:

Mi Superpoder es la Programación, es un videojuego 2D de aventura por niveles, el cual contiene un conjunto de retos que enseñarán a los niños y jóvenes a programar, a través del uso de pseudocódigo y lenguajes de programación como Python y Javascript. Este videojuego permitirá aprender sobre conceptos básicos, desde los más fundamentales como lo son el desarrollo de la lógica a través de ejercicios de secuencia y memoria, hasta los más avanzados enfocados en la creación de elementos por medio de código.

## Objetivo:

El videojuego está enfocado en la enseñanza de la programación en niños y jóvenes. Este contará con módulos que serán asignados a través de un usuario con el rol de profesor. La plataforma adicionalmente contará con una validación, que permitirá únicamente asignar módulos avanzados a niños que han superado un bloque o módulo por completo.

Cada módulo contiene un conjunto de ejercicios, los cuales se activan automáticamente cada vez que el niño resuelve uno previo con éxito. En cada módulo se aprenderán conceptos distintos de programación proporcionando las bases necesarias que se requieren para en un futuro ser grandes programadores.

Inicialmente los jugadores deberán resolver ejercicios de secuencia y diagramas de flujo, para pasar a código en bloques y finalmente a código real. Los profesores deberán asignar cada módulo dependiendo del nivel en que se encuentre el estudiante.

El objetivo del videojuego es enseñar programación desde las bases usando secuencias, pseudocódigo y diagramas de flujo, para posteriormente en etapas más avanzadas realizar ejercicios en un lenguaje de programación en particular, desarrollando habilidades como programador y no solo como codificador.

## Género:

Educación: enseñanza de la programación en niños y jóvenes  
Aventura

**Características:**

- Videojuego en 2D
- Vista en cenital
- Único jugador
- Retroalimentación visual
- Cámara fija o cámara que sigue al personaje, dependiendo del ejercicio a realizar
- Múltiples escenarios y obstáculos que cambian según el nivel

**Plataforma:**

El videojuego será desarrollado para navegadores web. Inicialmente las plataformas objetivo son Google Chrome. Para una etapa posterior se prevé garantizar el correcto funcionamiento en navegadores como Mozilla Firefox y Safari. La interfaz del sitio será adaptable a dispositivos móviles, para facilitar la accesibilidad a la plataforma por parte de los niños y jóvenes sin importar la herramienta de la cual dispongan en casa.

**Audiencia:**

El videojuego está dirigido a niños y jóvenes de edades entre los 6 y los 12 años. Sin embargo, personas de diferentes edades que quieran jugarlo para aprender programación, pueden hacerlo sin ningún problema. Se requieren competencias básicas lectoras y conocimientos mínimos en matemáticas como lo son sumas y restas.

**Lenguaje:**

El videojuego se encontrará en el idioma español inicialmente. En etapas posteriores al desarrollo se el videojuego se podrá encontrar en el idioma inglés.

**Sonidos:**

En este videojuego la musicalización es importante para mejorar la experiencia de usuario y hacer más atractivo el videojuego. Los niveles contarán con sonidos de retroalimentación que indicarán si un evento se realizó correctamente o no, además con música de ambiente. Todos los sonidos son configurables. La música del juego será tomada de bodegas de sonidos libres.

## Historia

### Outline:

La historia del videojuego gira alrededor de Mayito, un conejo que a través de sus habilidades en programación ayuda a sus amigos a salir de problemas causados por Rocco, un conejo malvado que tiene celos de Mayito.

A lo largo de sus aventuras, deberá superar obstáculos dejados en el camino por Rocco, para rescatar a sus amigos y objetos de valor, lo que poco a poco irá otorgando superpoderes, como volar, invisibilidad, super fuerza, super cognición, invulnerabilidad y teletransportación. Para lograrlo deberá hacer uso de la lógica y la programación.

### Historia - Sinopsis:

En el bosque Katios viven Mayito, Violeta, Rocco y muchos animalitos más, los cuales cada año participan en competencias de carreras. Violeta, la tortuga y Mayito el conejo siempre compiten juntos, pues Mayito ayuda a Violeta para llegar siempre a tiempo a la meta.

El día de la competencia Mayito y Violeta avanzan en la carrera como era costumbre. Mayito tomó un poco de ventaja para esperar bajo la sombra de un árbol, pues casi no le gustaba el sol, además así podía aprovechar para descansar un poco, mientras Violeta llegaba al punto de encuentro.

En el camino Violeta se encuentra con Rocco, quien le invita a tomar un atajo para llegar más rápido al lugar donde está Mayito. En su interior Rocco quería ganar la carrera a como diera lugar. Violeta acepta tomar el atajo y se pierde en el camino. Horas más tarde Mayito se despierta confundido porque Violeta no había llegado al lugar esperado.

Mayito empieza a buscar por la ruta de la competencia sin encontrar rastros de Violeta, decide olvidarse de la carrera e iniciar una búsqueda para encontrarla. En el camino se encontrará con varios obstáculos dejados por Rocco. Allí empieza la aventura, donde será necesario utilizar la programación para resolver todos los desafíos.

## **Escenas:**

El videojuego contará con varios escenarios, dependiendo de la misión dentro del ejercicio perteneciente a un módulo y nivel del jugador. Inicialmente los módulos con los que contará el juego son:

### **- Módulo 1: Recopilando pistas - Secuencias y patrones**

Este módulo va a contener todo el primer bloque de lógica, los cuales corresponden a los pasos previos a la programación, por lo tanto los ejercicios a realizar estarán muy poco relacionados con la codificación. La temática principal serán las partes de un computador y ejercicios de secuencia, lo que hará referencia al lenguaje de máquina. Es necesario aprender sobre las herramientas a utilizar para escribir y ejecutar código.

- **Ejercicios**
  - 1 (Las huellas de Violeta)**
  - 2 (Parte I: Señales arriba)**
  - 3 (Parte II: Señales arriba)**
  - 4 (Buscando a la izquierda)**
  - 5 (Caminando hacia abajo)**
  - 6 (Combinaciones y aventuras)**
  - 7 (Caminando a la derecha)**

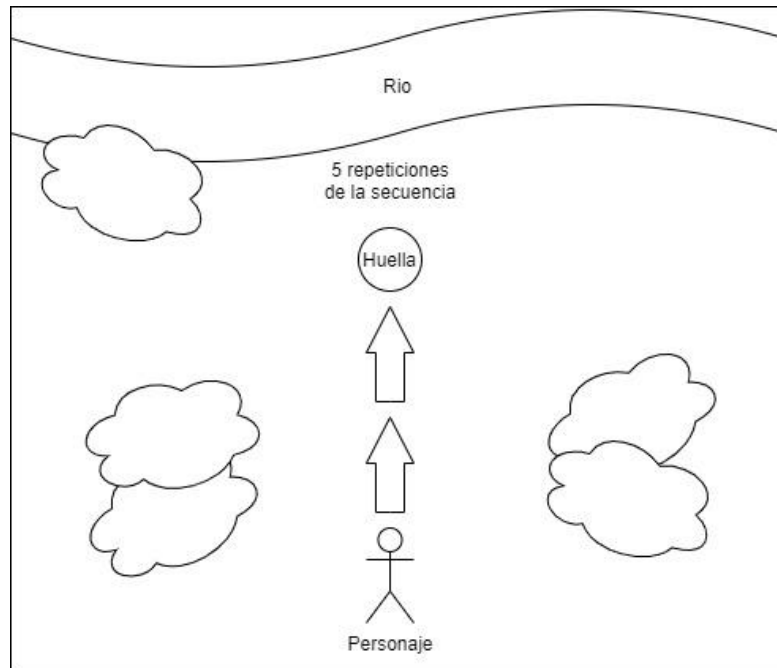
Mayito deberá buscar las señales que Violeta ha dejado en el camino, antes de que sean borradas por el viento. En este ejercicio de secuencia, el jugador deberá escoger únicamente objetos que pertenezcan a la forma de rayito.

### **Características:**

- Posición de la cámara: Cenital
- Elementos visuales: Paisaje con árboles un río donde será necesario construir un puente para cruzarlo, caminos sin salida por árboles como barrera.
- Variaciones de los escenarios: Posición de los árboles, ubicación de los obstáculos como piedras, troncos de madera, caminos sin salida y ríos

### **Descripción del módulo 1:**

Los niveles del módulo empezarán con una secuencia básica en donde el jugador deberá escoger “arriba, arriba, señal”, lo que indica que la señal estará a dos pasos por delante del personaje. Inicialmente el personaje deberá avanzar, y a medida que se hace evidente su progreso, se agregaran pasos como izquierda, derecha, abajo, arriba, diagonales.



**Figura 1:** Ejemplo nivel inicial del módulo 1. (Las flechas no serán visibles dentro del juego, solo sirven para representar los pasos que deberá dar el personaje para avanzar hasta la señal)

- **Módulo 2: Mis primeros obstáculos - Diagramas de flujo**

Rocco ha dejado muchos obstáculos en el camino, sin embargo, Mayito no se había encontrado con ellos hasta ahora. Es necesario hacer uso de diagramas de flujo, para dar las instrucciones a Mayito que le permitirán solucionar los desafíos. El camino es largo, es apenas el comienzo.

- **Ejercicios**

**8 (Parte I: Desafío anti Rocco)**

**9 (Parte II: Desafío anti Rocco)**

**10 (Toma de decisiones)**

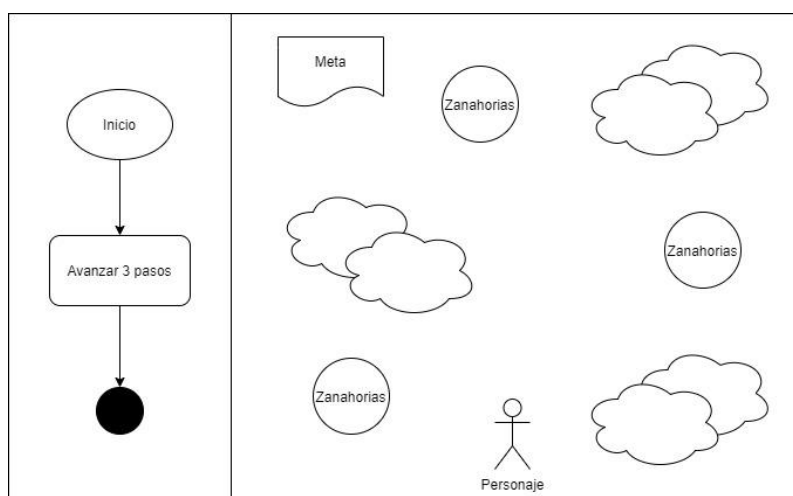
Mayito, deberá inicialmente (ejercicio 8) encontrar la salida del bosque por medio de instrucciones dadas a través de diagramas de flujo. Al salir del bosque se encontrará con un camino combinado con desierto, muchos cactus. Su objetivo será llegar al oasis para recargar energías, deberá superar obstáculos y caminos desconocidos. Se podrá encontrar serpientes, arbustos, rocas, arenas movedizas.

### Características:

- Posición de la cámara: Cenital
- Elementos visuales: Cactus, rocas, montañas de arena, plantas de desierto
- Variaciones de los escenarios: Posición de los cactus, ubicación de los obstáculos como piedras, arbustos, arenas movedizas

### Descripción del módulo 2:

Para el desarrollo de los niveles de este módulo, se hará uso de diagramas de flujo, donde el personaje deberá hacer el recorrido teniendo en cuenta el inicio, la acción y el fin.



**Figura 2:** Ejemplo inicial del módulo dos, donde se deberán recuperar las señales, dar instrucciones a través de diagramas de flujo (espacio para codificar) y evitar chocar con los obstáculos

### Tamaño del juego:

La posición de los elementos puede variar según la resolución del dispositivo. El videojuego aunque estará diseñado para web, será responsive, lo que facilitará el acceso a este.

Tamaño mínimo para celular: 2400 x 1600 (orientación horizontal)

Tamaño máximo en escritorio: 1200 x 800

### Referencias visuales

Juegos usados como referencia a nivel de concept art

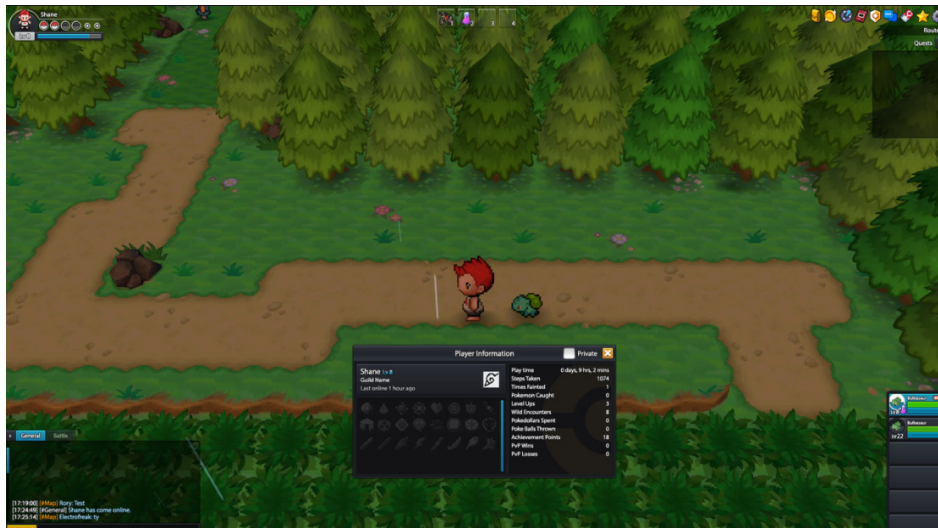
- CodeMonkey
- Pokemon Esmeralda
- Zelda Ocarina del tiempo 2D
- Journey Game



Figura 3: Referencia visual tomada de videojuego Pokemon



Figura 4: Referencia visual tomada de CodeMonkey



**Figura 5:** Referencia visual tomada de Pokéone



**Figura 6:** Referencia visual tomada del videojuego Zelda Ocarina del tiempo 2D

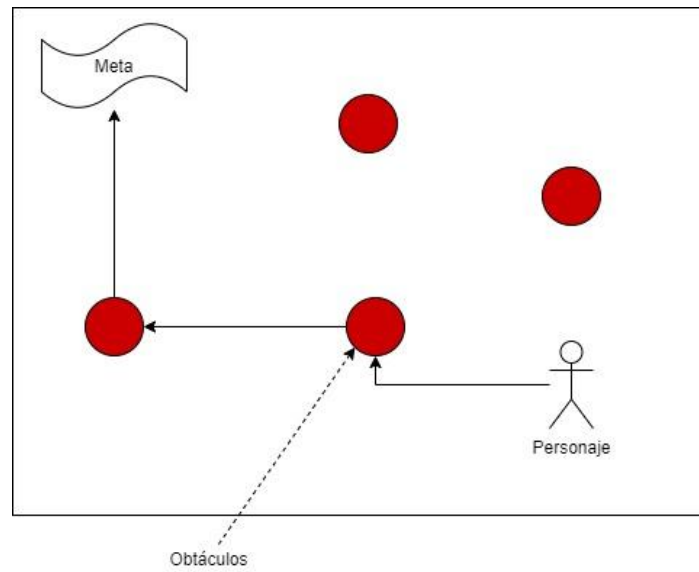


**Figura 7:** Referencia visual tomada de Journey Game

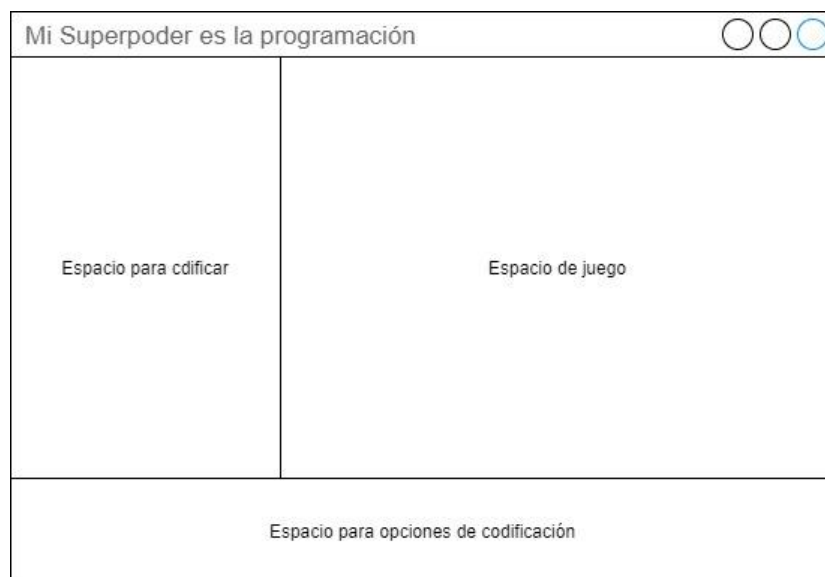


**Figura 8:** Referencia visual tomada de Stardew Valley Strange Doll

- Estructura básica del juego, aplica para todos los niveles



**Figura 9:** Ejemplo de posible conjunto de instrucciones que puede seguir el personaje



**Figura 10:** Ejemplo de pantalla de visualización de juego para escritorio (en móvil se visualizará de manera horizontal)

## Mecánicas de juego

### Cámara:

La cámara del juego estará posicionada de manera que el personaje podrá visualizarse desde arriba (vista cenital), desde la espalda o de lado dependiendo del desafío a resolver. La cámara será fija en los primeros módulos y se centrará en mostrar todos los elementos del escenario incluyendo al personaje.

### Jugadores:

Este juego está diseñado para un único jugador. Sin embargo, se espera a futuro poder interactuar en línea con otros jugadores, otorgando la posibilidad de que sea multijugador.

### Movimientos generales:

El personaje podrá subir, bajar, ir a la izquierda, derecha, saltar y utilizar un conjunto de súper poderes a medida que gana experiencia y supera desafíos. Para mover al personaje se utilizará código en el lenguaje de programación Python o Javascript, estructuras en forma de secuencia y diagramas de flujo, no podrá moverse con el teclado o mouse, a no ser que el desafío se lo indique y sea necesario en las instrucciones que se deben dar al personaje. El jugador podrá configurar cualquier letra del teclado para activar los movimientos y acciones del personaje, solo en niveles avanzados.

En el juego podrán encontrarse eventos básicos como avanzar, girar, volar, atrapar, lanzar, las cuales son funciones propias de la plataforma.

|         |  |
|---------|--|
| Avanzar | Permite al personaje avanzar ya sea para adelante o para atrás   |
| Girar   | Permite al personaje girar a la izquierda o derecha, con el fin de cambiar la posición inicial y mirar otro objetivo |
| Volar   | Permite al personaje ir por encima de ciertos obstáculos   |
| Atrapar | Permite al personaje tomar algún ítem o elemento que podrá utilizar para su defensa                                  |
| Lanzar  | Permite al personaje arrojar el objeto atrapado previamente  |

**Configuración inicial:**

El personaje principal aparecerá en un camino donde deberá superar un conjunto de obstáculos iniciales. Cada vez que un nivel sea superado con éxito, la dificultad aumentará. Los personajes secundarios como villanos o amigos, estarán relacionados a niveles avanzados, donde se tenga mayor claridad de los conceptos de programación y sea posible crear código más elaborado, con el fin de crear combates entre ellos.

**Salud:**

El personaje tendrá opciones de vida solo cuando en el nivel éste se enfrente al villano. El código creado debe derrotar al enemigo y evitar recibir impacto de los objetos lanzados y los obstáculos impuestos en el camino. Si la vida se pierde por completo, el jugador deberá plantear a nivel de código otra solución más elaborada que cubra el objetivo. En caso de superar cinco intentos y el nivel no sea superado, el jugador se devolverá al nivel anterior a modo repaso, pero manteniendo su progreso.

**Recompensas:**

Por cada nivel superado con éxito el juego otorgará puntos en forma de zanahoria, los cuales podrán ser utilizados para comprar accesorios al personaje principal. Es posible ganar de 1 punto de experiencia a 3, depende completamente de la solución descrita en el código. La solución más óptima siempre otorgará mayor puntaje.

**Obstáculos:**

Será posible encontrarse ríos sin puente, caminos rocosos, enemigos, mensajes con textos erróneos para confundir al jugador. Cada nivel es una estrategia, por lo que se debe diseñar un plan distinto para lograr una correcta solución.

Los obstáculos podrán ser esquivados, destruidos o cambiados de lugar por el personaje principal.

## Personajes

**Mayito:** Es el protagonista, es comprometido, inteligente, le gusta ayudar y compartir. Su misión es encontrar a Violeta ya que está perdida. Él deberá resolver ejercicios de lógica y análisis para alcanzar su objetivo.



**Figura 11:** Mayito

**Violeta:** Es una tortuga muy divertida. Es amigable y le gusta jugar con sus amigos. Es bastante curiosa y confiada, le encanta el color violeta y cree en la bondad de todos. Nunca duda o piensa mal de alguien, puede llegar a ser muy ingenua.



**Figura 12:** Violeta

**Rocco:** Es un conejo malvado. Es odioso, envidioso y enemigo de Mayito. No le gusta que le salgan bien las cosas a los demás, siempre quiere ganar y no acepta perder. Siempre está haciendo bromas de mal gusto y poniendo en apuros a todos, obligando a Mayito a salir al rescate de sus amigos.



**Figura 13:** Rocco

## Gameplay detallado

El personaje principal se desenvolverá en diferentes escenarios, según el nivel en el que se encuentre. Contará con diversos movimientos que van desde avanzar, retroceder, girar, saltar, volver, arrojar súper poderes y volverse invisible.

En la pantalla de juego, se tendrá un espacio para agregar bloques de diagramas de flujo, bloques de código, código real, todo se mostrará de manera progresiva según el avance del jugador. Cada nivel superado con éxito, habilitará otro de mayor complejidad.

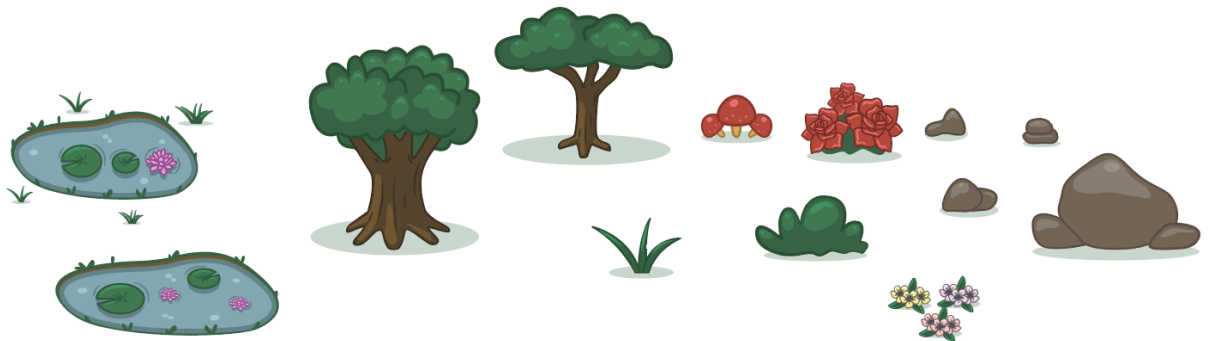
En los niveles iniciales, los cuales corresponderán a fundamentos de programación I, el jugador deberá resolver una serie de ejercicios de flujos para hacer que el personaje llegue a la meta, inicialmente deberá superar obstáculos que involucren movimientos básicos como lo son avanzar, retroceder y girar. Esto seguido de ejercicios de secuencia con el fin de desarrollar habilidades lógicas.

Al superar correctamente los ejercicios de secuencia y diagramas de flujo, la historia seguirá siendo la misma, pero el personaje deberá superar otros obstáculos en otros escenarios, con otras dinámicas, donde deberá recurrir a bloques de código que le permitan avanzar, retroceder, girar y saltar. En esta etapa el jugador deberá aprender sobre arreglos y cómo recorrerlos, esto le permitirá obtener objetos que luego podrán ser utilizados como herramientas de defensa.

En la etapa final, el jugador deberá hacer que aparte de los movimientos básicos, el personaje pueda arrojar súper poderes entre estos están disparo de rayo láser, disparo de zanahorias, capacidad de volverse invisible por un corto tiempo. El jugador deberá ordenar al personaje a través de código basado en python o Javascript, deberá indicar el paso a paso para llevar al personaje final a la meta.

Como reto final, el cual servirá posteriormente como evaluación, el jugador deberá implementar lo aprendido desde los primeros módulos hasta el final, resolviendo un reto en el cual el jugador deberá ejecutar todas las acciones permitidas a través de instrucciones dadas por diagramas de flujo, secuencia, código en bloques y código real.

## Concept art



**Figura 14:** Elementos del escenario bosque módulo 1



**Figura 15:** Ejemplo de montaje escenario, módulo 1



## **Desarrollos futuros**

Para el futuro se espera contar con al menos 5 módulos donde sea posible distribuir los ejercicios de fundamentos I - II - III a intermedio y avanzado. Cada módulo deberá contar con al menos 15 niveles. Adicional se espera incluir en uno de los módulos ejercicios sobre partes del computador para que los niños aprendan a distinguir los componentes y las diferencias de software y hardware.

### 3. Cuestionario de pruebas de la herramienta

#### Pruebas de captura de conocimiento:

Para llevar a cabo la captura de conocimientos previos se diseñó un formulario el cual permitirá obtener información sobre: 1) nivel de matemáticas, 2) nivel de lectura, conocimientos en programación (secuencias y flujos).

Para la construcción de este formulario se usa como referencia el libro "Hello Ruby Adventures in Coding" de Linda Luikas. Las gráficas utilizadas corresponden a la herramienta web Mi Superpoder es la Programación.

A continuación se presenta el formulario a utilizar en las sesiones 1 y 5 del taller. Este formulario debe realizarse con apoyo de un tutor o profesor.

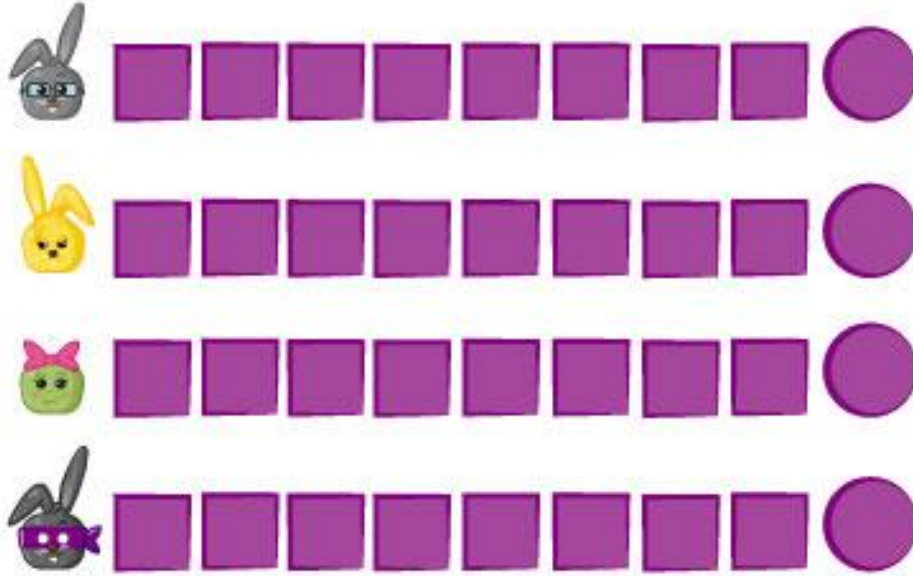
Nombre: \_\_\_\_\_

Edad: \_\_\_\_\_

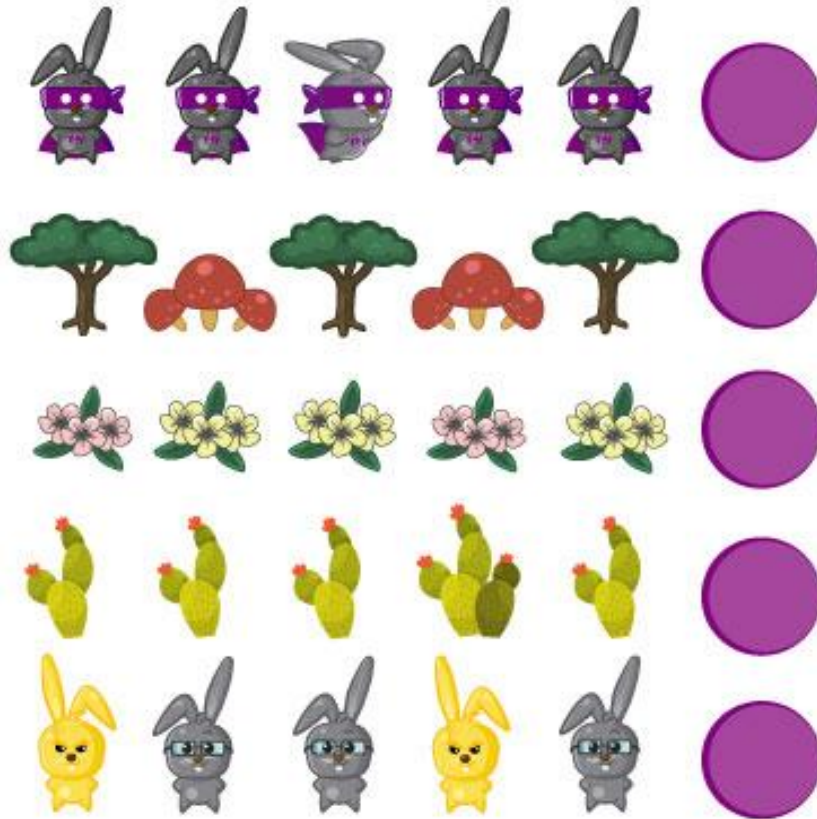
Año escolar: \_\_\_\_\_

1. Identifica los elementos iguales. Marca con una X en el cuadro que corresponda a la cara del conejo gris con gafas, tortuga con moña rosada, conejo amarillo, conejo con antifaz. Realiza un conteo de cuántos elementos hay de cada uno e ingresa el número con el total de elementos identificados en el círculo.





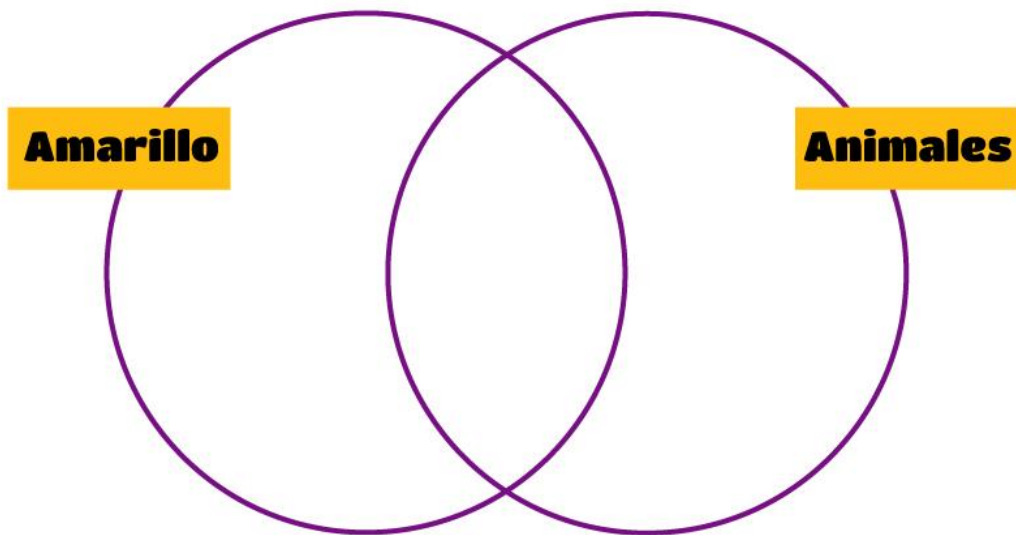
2. Continúa la secuencia, identifica qué elemento sigue dibujándolo en el círculo morado. Observa muy bien antes de completar la secuencia.

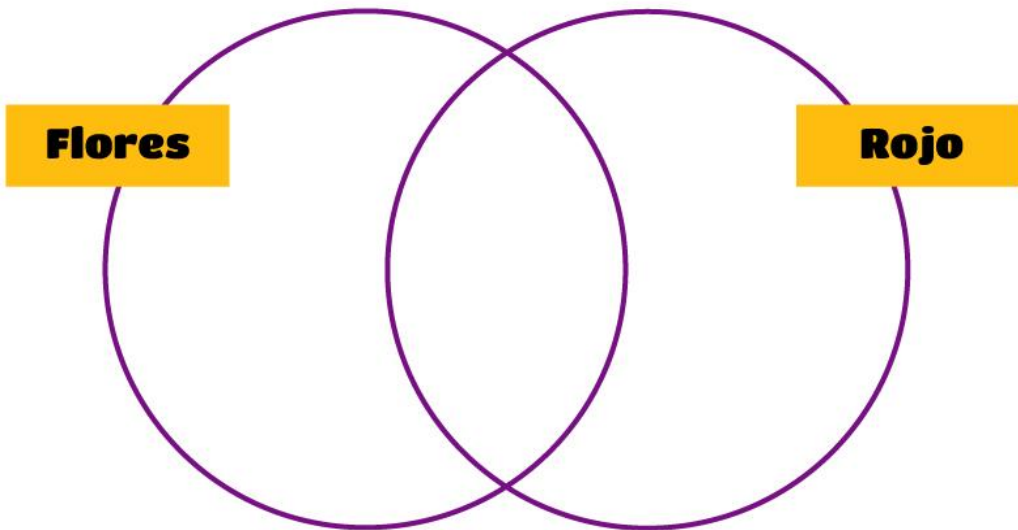
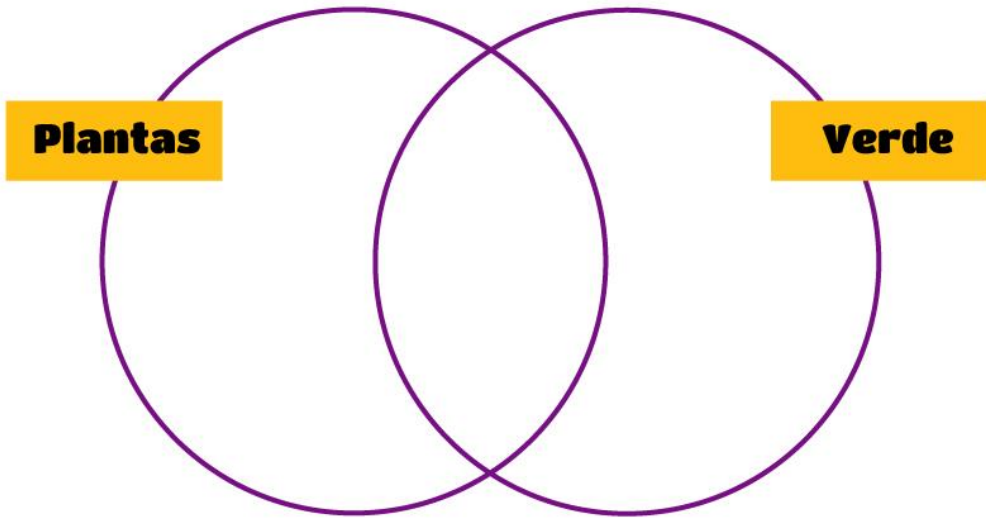


3. Ahora vamos a crear un algoritmo (o serie de pasos a realizar para completar una tarea). Crea el paso a paso para lavarte los dientes, enumerando y describiendo detalladamente cómo te cepillas los dientes.

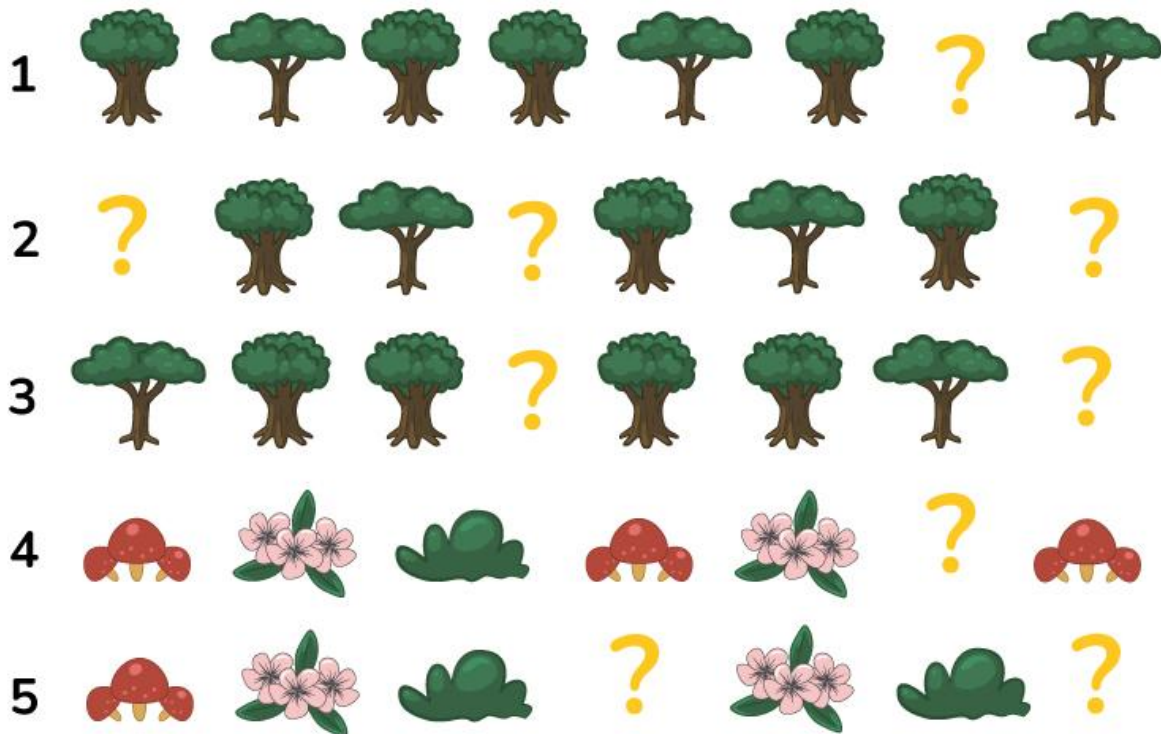


4. Vamos a agrupar elementos similares del mismo tipo o color, a esto se le conoce como conjuntos. En el círculo con la etiqueta “animales” agrega todas las figuras que representen animales, en el círculo con la etiqueta “amarillo” agrega todas las figuras de color amarillo. En la unión de ambos círculos agrega los elementos que sean animales y de color amarillo. (Así con todos los demás ejercicios, sigue las instrucciones dadas por el tutor, antes de realizar este ejercicio.)






























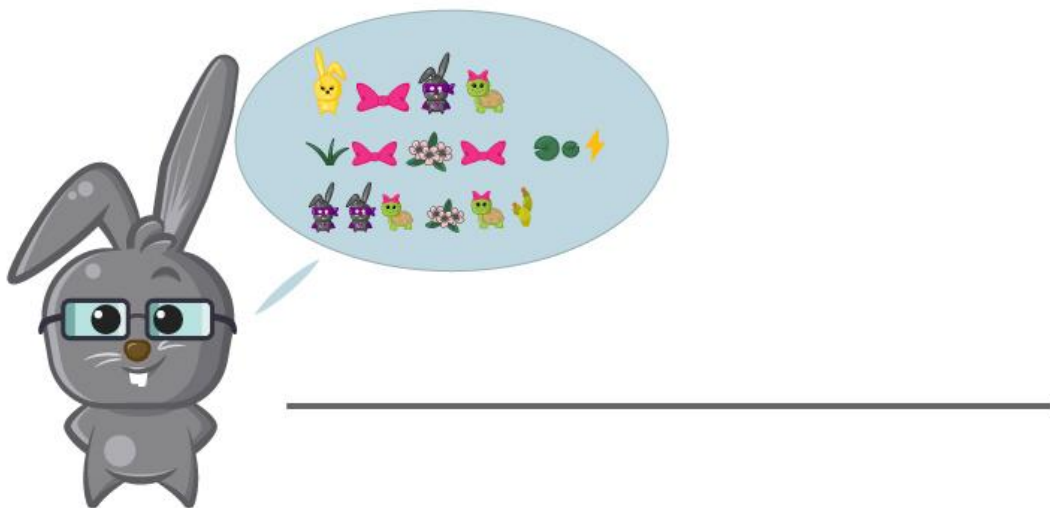


5. Vamos a continuar con las secuencias, reemplaza el signo de pregunta por la figura que hace falta.



6. El lenguaje que hablan los computadores no podemos entenderlo las personas tan fácilmente, por eso vamos a simular como sería el lenguaje de una máquina. Reemplaza cada figura por una letra de nuestro abecedario para descifrar el mensaje que intentan decirte, recuerda que las máquinas hablan un lenguaje diferente al nuestro y debemos traducirlo para poder entenderlo.

|          |   |          |   |          |   |          |  |          |   |          |   |
|----------|---|----------|---|----------|---|----------|--|----------|---|----------|---|
| <b>A</b> |  | <b>F</b> |  | <b>K</b> |  | <b>O</b> |  | <b>T</b> |  | <b>Y</b> |  |
| <b>B</b> |  | <b>G</b> |  | <b>L</b> |  | <b>P</b> |  | <b>U</b> |  | <b>Z</b> |  |
| <b>C</b> |  | <b>H</b> |  | <b>M</b> |  | <b>Q</b> |   | <b>V</b> |  |          |   |
| <b>D</b> |  | <b>I</b> |  | <b>N</b> |  | <b>R</b> |  | <b>W</b> |  |          |   |
| <b>E</b> |  | <b>J</b> |  | <b>Ñ</b> |  | <b>S</b> |   | <b>X</b> |  |          |   |






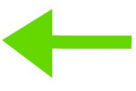
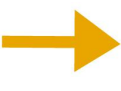

Recuerda que debes responder al mensaje descifrado llevando a cabo la traducción de tu respuesta usando las figuras para convertirlo en lenguaje de máquina.

7. Vamos a indicar los pasos que debes dar para llegar desde el inicio hasta cada uno de los personajes que se muestran en el mapa. Escribe cada paso que debes dar para llegar al personaje y finalmente saludarla. Recuerda que cada paso representa un cuadro del mapa. Puedes moverte en las direcciones arriba, abajo, izquierda y derecha.
  - Describe primero los pasos que debes dar para llegar al conejo amarillo y saludarlo.
  - Describe los pasos que debes dar para llegar a la tortuga y saludarla.
  - Describe los pasos que debes dar para llegar al conejo gris y saludarlo.

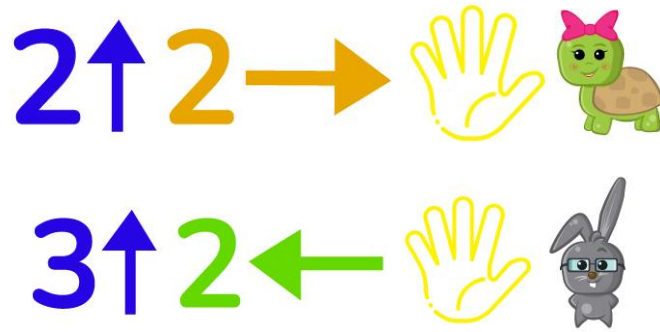


**Parte 2, después del uso de la herramienta:**

Describe los pasos que debes dar para llegar a cada uno de los personajes utilizando flechas según la dirección. Una flecha representa un cuadro del mapa. Dirígete hasta cada personaje desde el punto de partida (inicio) y saluda a todos los amiguitos del mapa.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| Inicio  | Arriba  | Abajo   | Izquierda   | Derecha   | Saludar   |

8. Teniendo en cuenta el ejercicio anterior, diseña el camino que deberías recorrer para dirigirte desde el inicio hasta el conejo amarillo.
  
9. Vamos a interpretar las instrucciones que se muestran en la imagen y a representarlas en el mapa.



10. **Punto a realizar después del uso de la herramienta.** Desarrolla el punto 8 y 9 utilizando diagramas de flujo. Al finalizar crea un diagrama de flujo nuevo para el punto 9 donde se agregue una condición que permita saludar a la tortuga si esta tiene moño, en caso de no tener moño finalizar el flujo. Para el caso del conejo gris, agregar una condición que permita saludarlo solamente si tiene gafas, si no tiene gafas también se dará fin al flujo.
11. Para seguir con la construcción de algoritmos, vamos a enumerar e indicar uno a uno de los pasos que utilizas al momento de tomar una ducha o un baño.



12. Punto para después del uso de la herramienta. Describe por medio de un diagrama de flujo los pasos para bañarte, válida si el agua está fría o caliente, si el agua está caliente te lavarás el cabello, si está fría, seguirás con el resto del proceso y no te lavarás el cabello.

#### 4. Políticas de privacidad



Versión: 1.0

Erika Jissel Gutiérrez Beltrán

**Políticas de privacidad**

## Políticas de privacidad

Para la captura de la información a través de la técnica de observación mientras los niños y jóvenes interactúan con la herramienta, se diseñaron las siguientes políticas de privacidad, con el fin de proteger los datos de todos los menores de edad que hagan parte de las pruebas. Estas buscan proteger la identidad de los niños y jóvenes, ya que se tiene interés en capturar fotos y videos que sean de apoyo y complemento para las pruebas y la sustentación final del proyecto de grado de maestría. Este material será exclusivo para uso académico y se prohíbe su reproducción total o parcial en otros escenarios.

La política de privacidad se construye teniendo en cuenta la ley de protección de datos personales de Colombia o ley 1581 de 2012, además del decreto 1074 de 2015 el cual precisa:

“Artículo 2.2.2.25.2.9. Requisitos especiales para el tratamiento de datos personales de niños, niñas y adolescentes. El Tratamiento de datos personales de niños, niñas y adolescentes está prohibido, excepto cuando se trate de datos de naturaleza pública, de conformidad con lo establecido en el artículo 7 de la Ley 1581 de 2012 y cuando dicho Tratamiento cumpla con los siguientes parámetros y requisitos:

1. Que responda y respete el interés superior de los niños, niñas y adolescentes.
2. Que se asegure el respeto de sus derechos fundamentales.

Cumplidos los anteriores requisitos, el representante legal del niño, niña o adolescente otorgará la autorización previo ejercicio del menor de su derecho a ser escuchado, opinión que será valorada teniendo en cuenta la madurez, autonomía y capacidad para entender el asunto. Todo responsable y encargado involucrado en el tratamiento de los datos personales de niños, niñas y adolescentes, deberá velar por el uso adecuado de los mismos. Para este fin deberán aplicarse los principios y obligaciones establecidos en la Ley 1581 de 2012 y presente capítulo. (...).”

### **¿Qué es la política de privacidad y cómo protegerá los datos recogidos al momento de hacer las pruebas?**

La política de privacidad suministra información que permite entender cómo los datos recogidos de los niños y jóvenes serán manipulados, pero a su vez cómo serán protegidos, garantizando que no serán distribuidos si no se aceptan estas políticas.

Todos los niños y jóvenes deben estar de acuerdo en cómo se capturará la información, también se debe tener en cuenta el consentimiento del representante legal, pues es esta persona quién tiene el criterio final.

Los datos no serán publicados siempre y cuando así lo desee el representante legal, niños y jóvenes. Sin embargo, si las políticas de privacidad son aceptadas, la información capturada será utilizada únicamente con fines académicos, es decir, no será posible por ningún motivo, compartir la información en redes sociales o sitios web de dominio público.

La información capturada a través de formularios, fotografías y video podrán utilizarse únicamente para estudios estadísticos y evaluación de la herramienta. La proyección de las imágenes capturadas podrán utilizarse únicamente en el recinto universitario y por ningún motivo podrán salir de este.

Estas políticas aplican para:

- Formularios de pruebas
- Fotografías y videos capturados mediante la ejecución de las pruebas
- Presentación de los resultados de manera académica
- Sitio web de [misuperpodereslaprogramacion.com](http://misuperpodereslaprogramacion.com)
- Sitio web de la herramienta [clases.misuperpodereslaprogramacion.com](http://clases.misuperpodereslaprogramacion.com)

### **¿Qué información recopilamos sobre tí?**

- Información pública:  
Nombre completo, grado escolar, edad, nombre del colegio en donde se estudia.
- Información privada:  
Fotos y videos del participante mientras interactúa con la herramienta o participa de las entrevistas

### **¿Es necesario recopilar esta información?**

Sí, los datos que sean suministrados a través de las pruebas son de gran importancia ya que, permitirán analizar la funcionalidad de la herramienta web Mi Superpoder es la Programación. Sin estos datos no será posible llevar a cabo las estadísticas y comparaciones con otras herramientas, además de conocer los conocimientos previos y los adquiridos luego de la interacción y realización de las actividades de la herramienta. Sin esta información no será posible tener un análisis completo y no será posible tener información concreta en la sustentación de los resultados.

### **¿Cómo usamos la información recopilada?**

Los datos recopilados serán utilizados para evaluar la herramienta a nivel de experiencia de usuario y usabilidad, además permitirá capturar información sobre los conocimientos previos en programación y los adquiridos luego del uso de la herramienta.

La información permitirá a nivel estadístico conocer si la herramienta cumple su objetivo frente a otras ya existentes enfocadas en la enseñanza de la programación.

### **¿Cómo capturamos la información?**

La información será capturada a través de un formulario donde se obtendrá información pública personal como nombre, edad, grado escolar y colegio. También se capturará información a través de entrevistas que serán almacenadas en formato imagen y video al momento de aplicar la técnica de observación mientras se hace uso de la herramienta web.

## 5. Interfaces de alta fidelidad, diseño UX



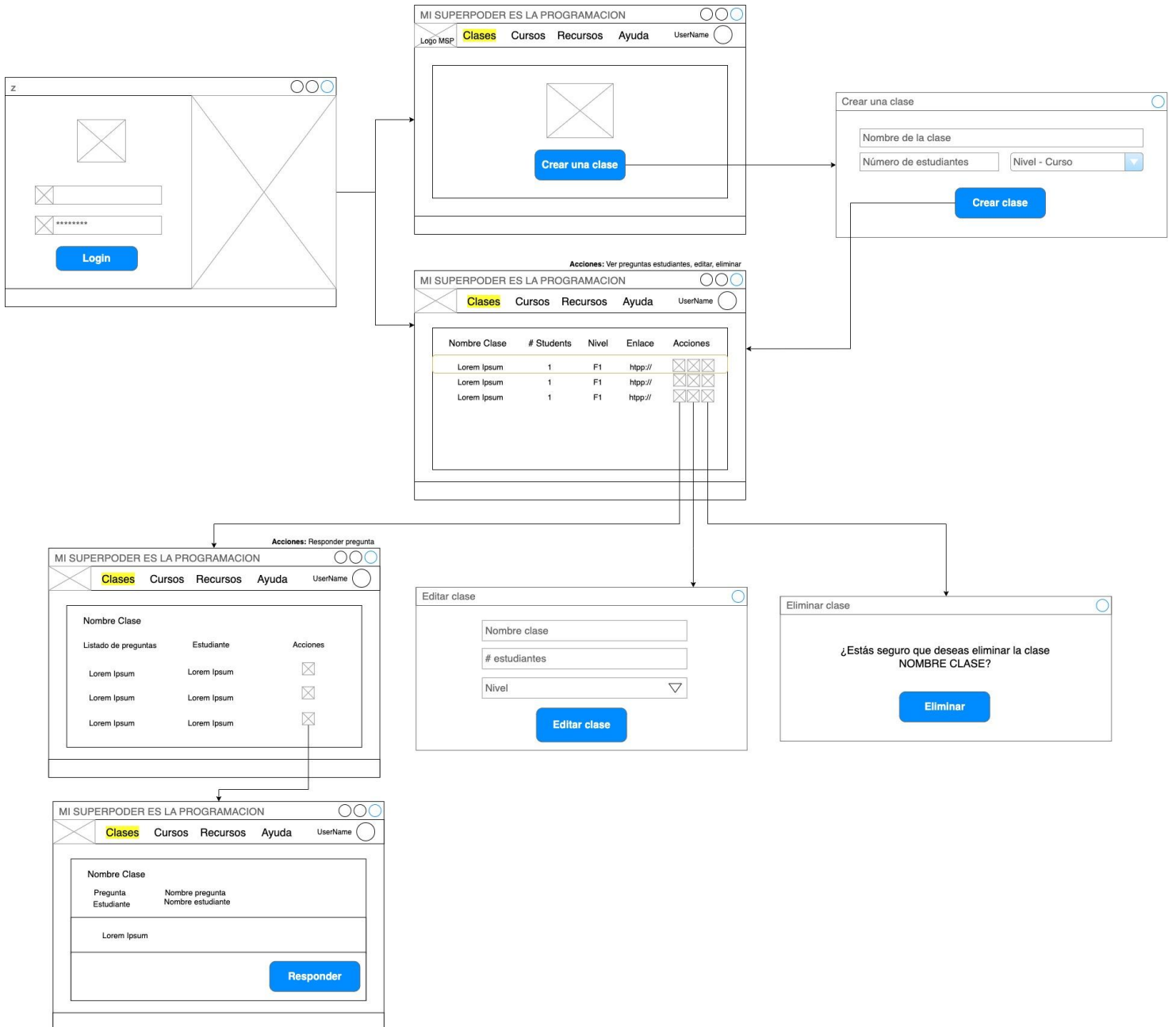
Versión: 1.0

Erika Jissel Gutiérrez Beltrán

**Interfaces, mockups y diseño UX**

## Mockups y flujo interfaz profesor

- Pantalla login, creación y visualización de una clase

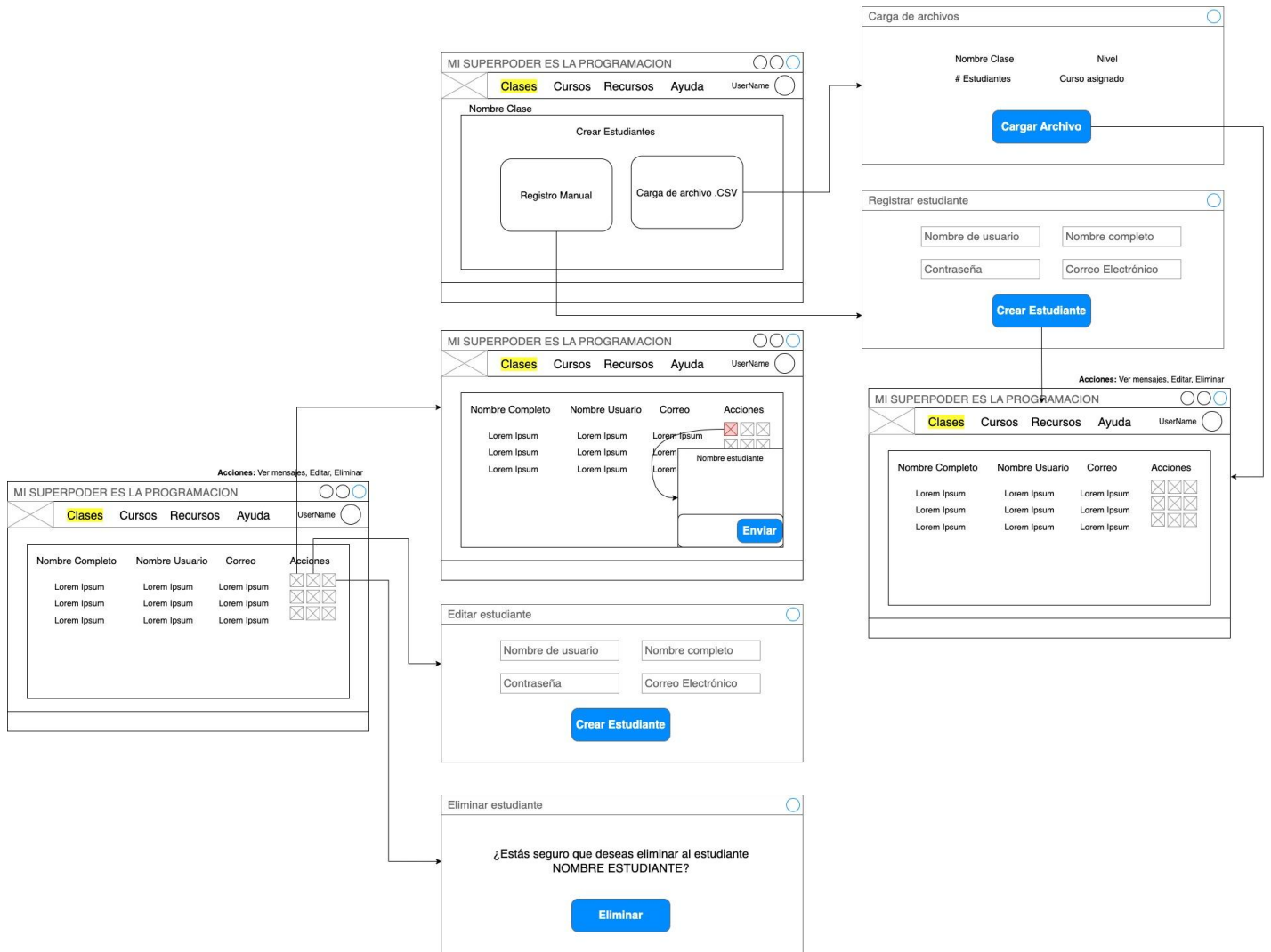


22

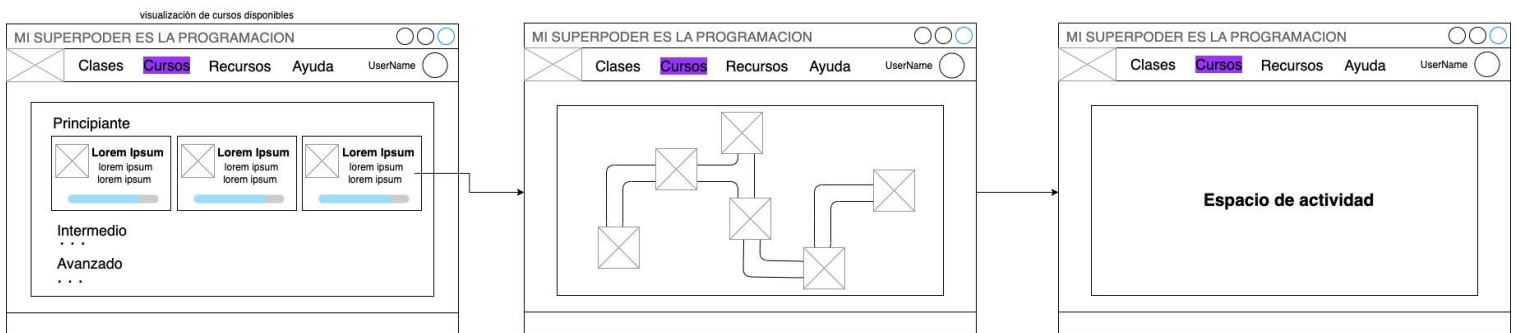
22 Mockups pantalla del profesor:

<https://drive.google.com/file/d/1z4KMEixyH86z4Hc9juqLY-dwzVvve1Dr/view?usp=sharing>

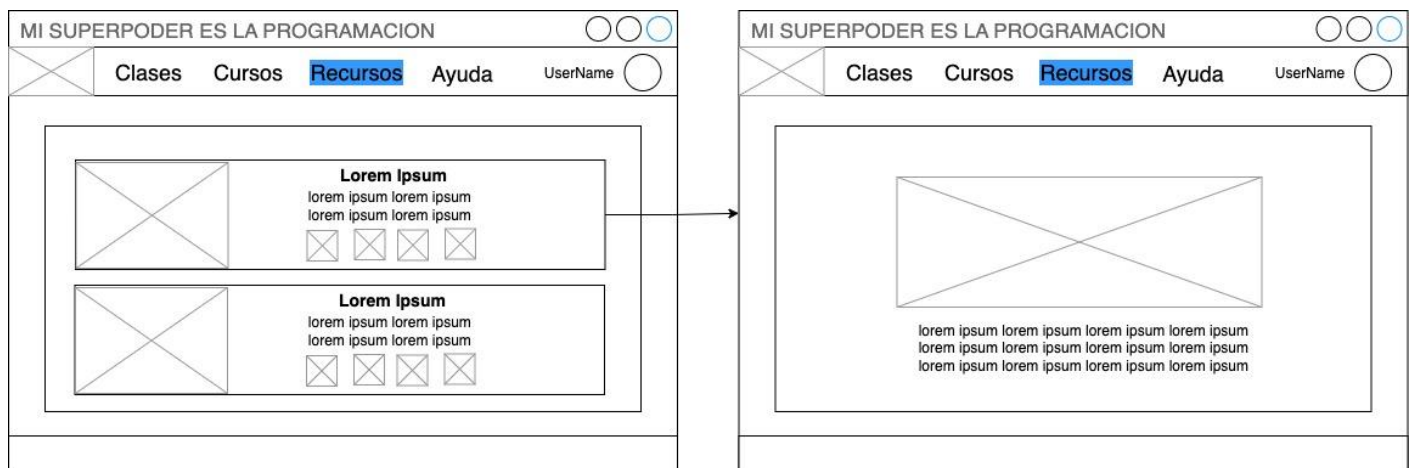
- Pantalla creación y visualización estudiante, ventana de chat



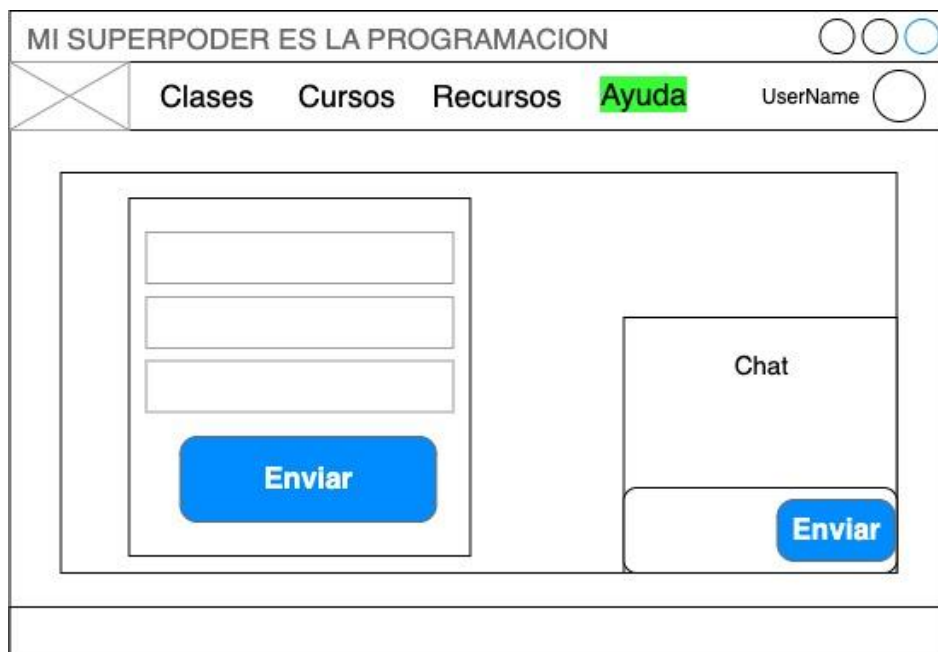
- Pantalla de visualización de módulos y espacio de juego del nivel o actividad



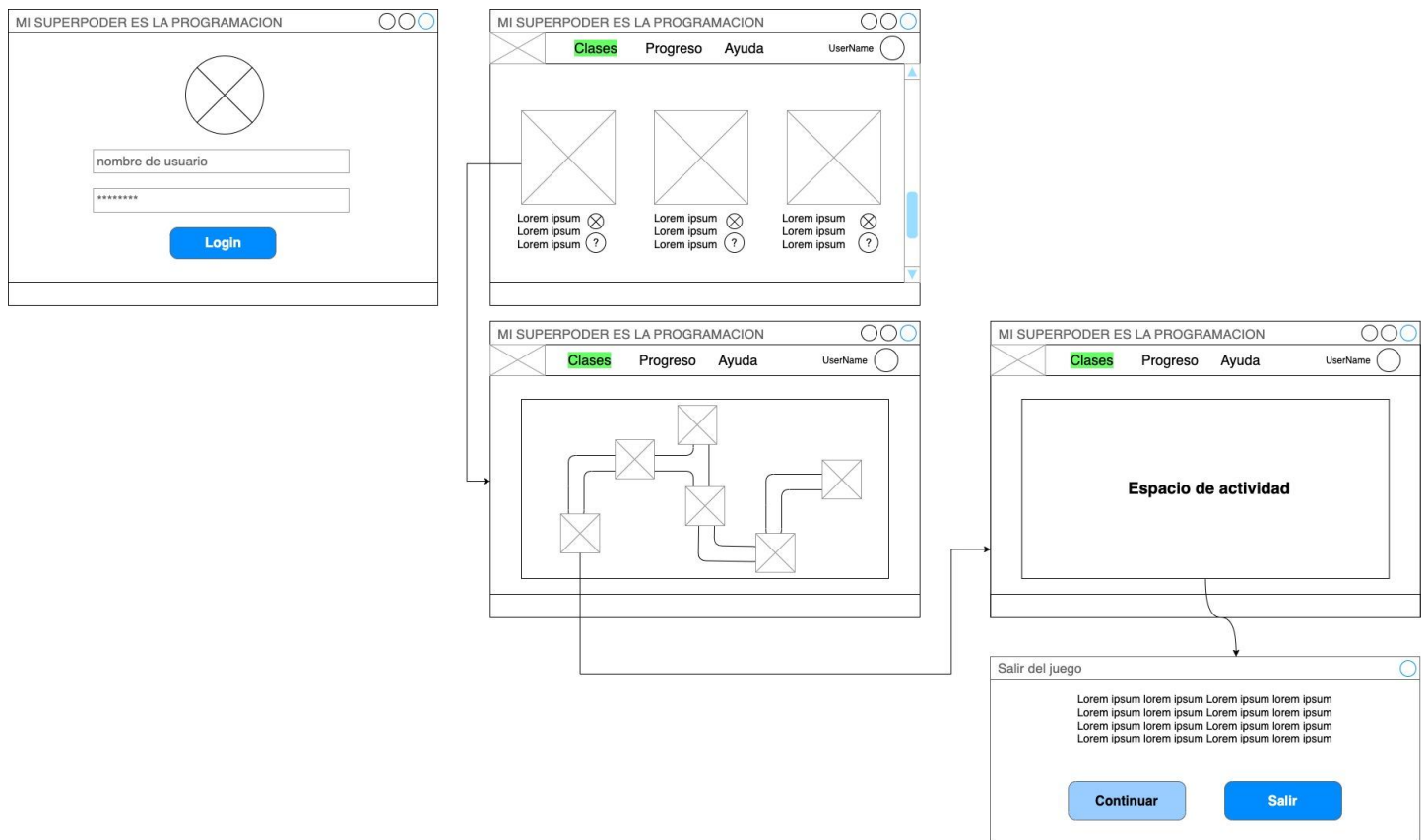
- Pantalla de visualización de recursos para capacitación de los profesores



- Pantalla de ayuda para recibir soporte técnico o responder solicitudes de los estudiantes

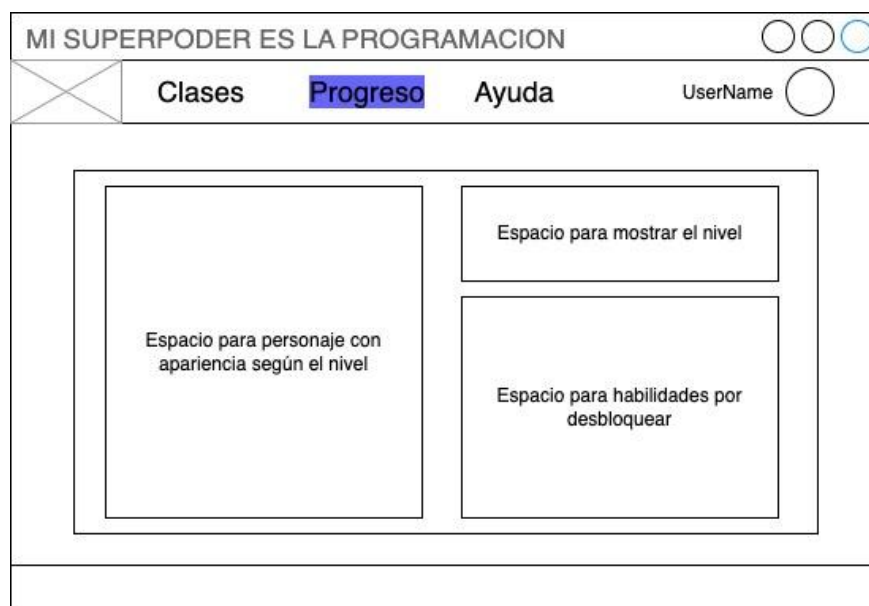


- Pantalla login, visualización de módulos asignados, progreso y espacio de juego



23

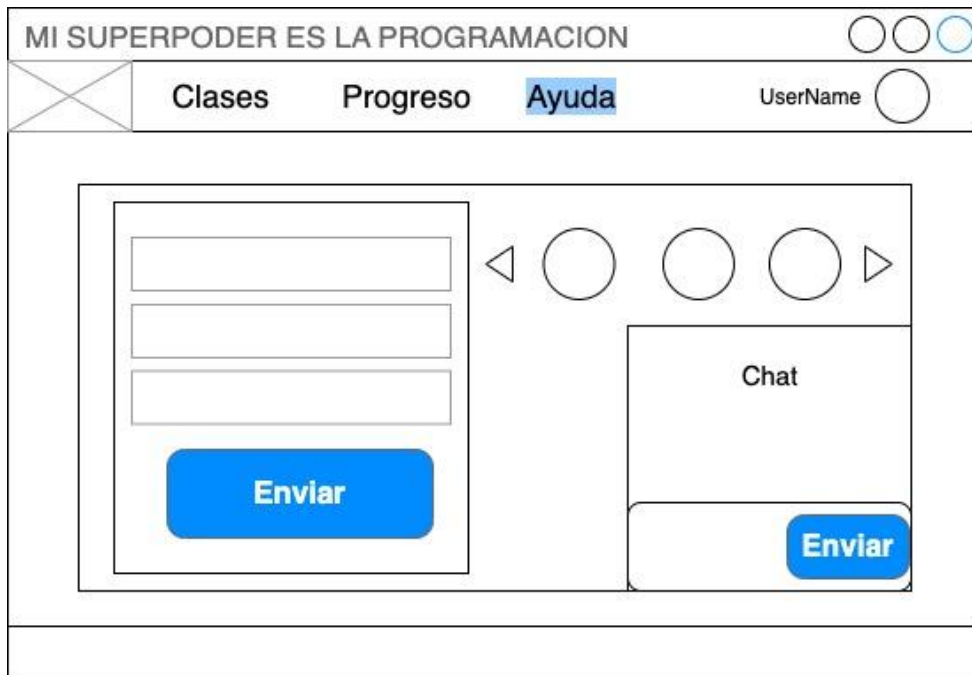
- Pantalla de progreso, visualización de los puntos y porcentaje de avance



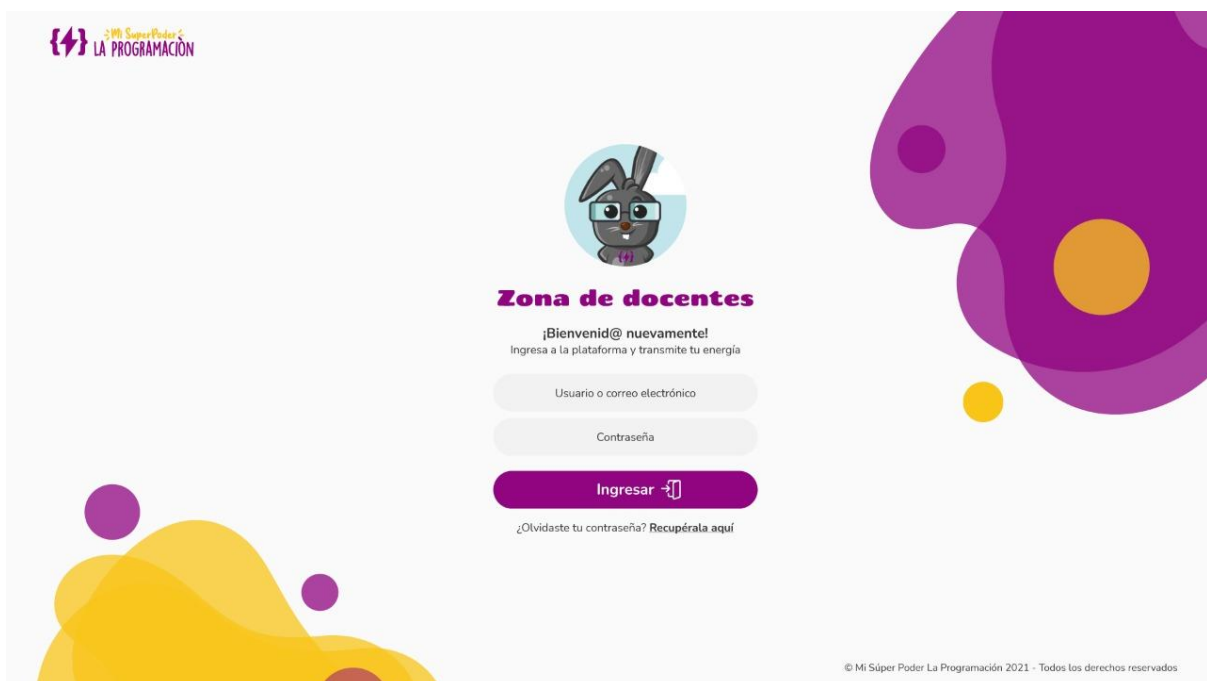
<sup>23</sup> Mockups pantalla del estudiante:

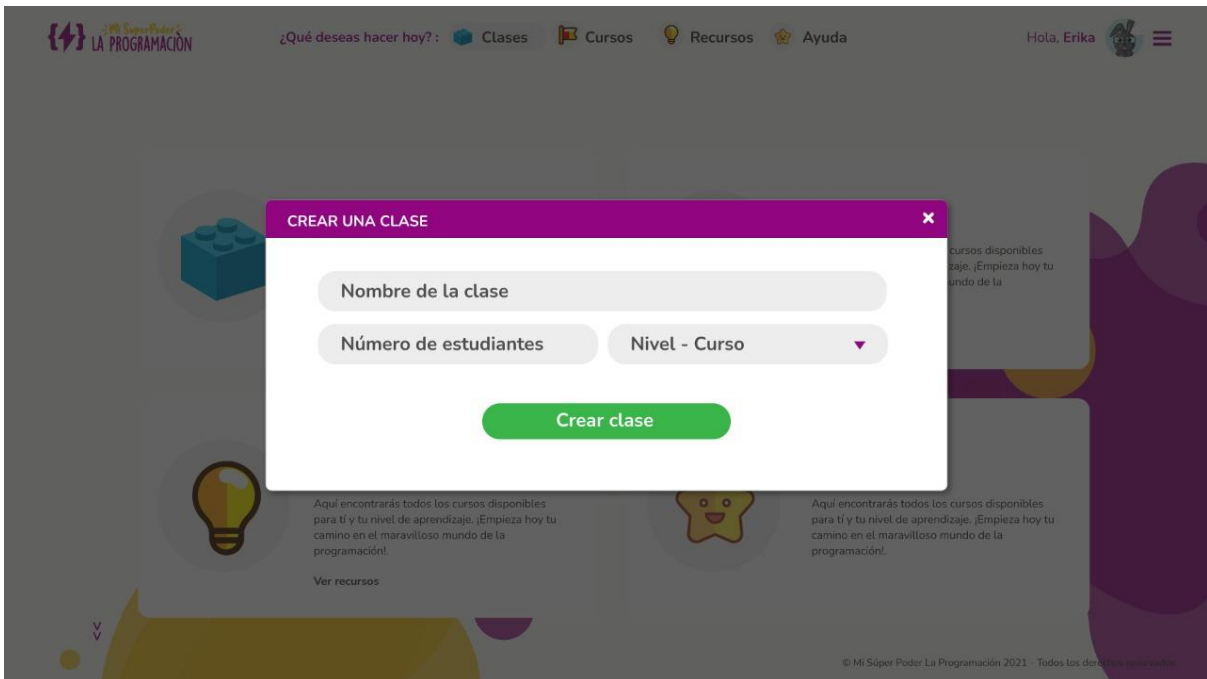
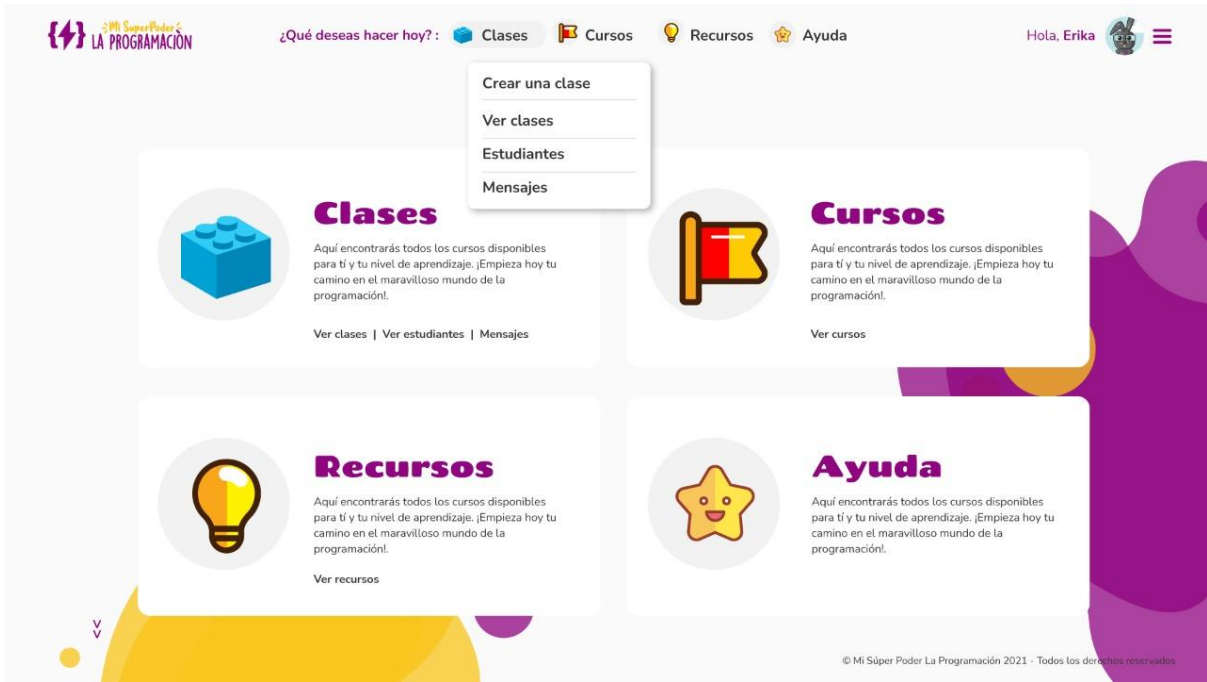
<https://drive.google.com/file/d/1z4KMExiyH86z4Hc9juqLY-dwzVvve1Dr/view?usp=sharing>




- Pantalla de ayuda, solicitud de asesoría y explicación a los profesores



## Diseño UX pantallas rol profesor








¿Qué deseas hacer hoy? :
[Clases](#)
[Cursos](#)
[Recursos](#)
[Ayuda](#)
Hola, Erika



## Cursos


[NIVEL 1](#)
[NIVEL 2](#)
[NIVEL 3](#)
[NIVEL 4](#)
[NIVEL 5](#)




Aventura de programación  
Parte I: Fundamentos



Aventura de programación  
Parte I: Fundamentos






Aventura de programación  
Parte I: Fundamentos



Aventura de programación  
Parte I: Fundamentos

© Mi Súper Poder La Programación 2021 - Todos los derechos reservados


¿Qué deseas hacer hoy? :
[Clases](#)
[Cursos](#)
[Recursos](#)
[Ayuda](#)
Hola, Erika




## Estudiantes


[Crear un estudiante](#)
[Subir lista estudiantes](#)

| NOMBRE COMPLETO | NOMBRE DE USUARIO | CORREO | ACCIONES |
|-----------------|-------------------|--------|----------|
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |
| Mayito Alberto  |                   |        |          |

**SUBIR LISTA DE ESTUDIANTES** ✕

ARRASTRAR ARCHIVOS AQUÍ

 Lista numero 1 .csv


















[<< Anterior](#)
[2 de 20](#)
[Siguiente >>](#)


© Mi Súper Poder La Programación 2021 - Todos los derechos reservados

## Estudiantes

Crear un estudiante

Subir lista estudiantes

| NOMBRE COMPLETO                | NOMBRE DE USUARIO | CORREO                         | ACCIONES  |
|--------------------------------|-------------------|--------------------------------|---|
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |    |

<< Anterior 2 de 20 Siguiente >>



## Estudiantes



| NOMBRE COMPLETO                | NOMBRE DE USUARIO | CORREO                         |
|--------------------------------|-------------------|--------------------------------|
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |
| Mayito Alberto Gutierrez Pérez | Mayito            | mayito-elmejor2022@outlook.com |

### Mensajes de Mayito

Lorem ipsum dolor sit amet, consete

Lorem ipsum dolor sit amet, consete

Enviar

 ¿Qué deseas hacer hoy? : [Clases](#) [Cursos](#) [Recursos](#) [Ayuda](#) Hola, Erika 

## Ayuda


A través del siguiente formulario podrás solicitar ayuda a nuestros asesores con respecto al uso de la plataforma. Responderemos a la mayor brevedad.

Tema con el cual necesitas ayuda



Tipo de ayuda que necesitas

Cuéntanos más al respecto...

[Solicitar ayuda](#)



© Mi Súper Poder La Programación 2021 - Todos los derechos reservados

 ¿Qué deseas hacer hoy? : [Clases](#) [Cursos](#) [Recursos](#) [Ayuda](#) Hola, Erika 

### ELIMINAR UNA CLASE ✕

¿Está seguro que desea eliminar esta clase?

[Cancelar](#) [Eliminar clase](#)

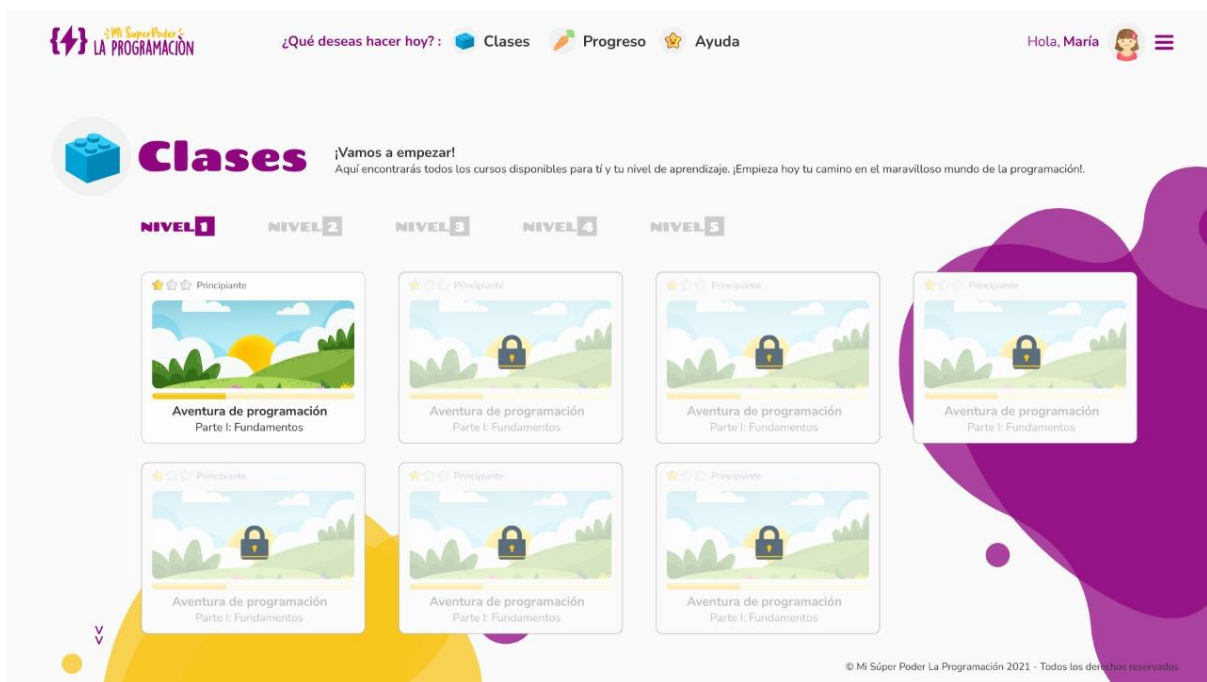
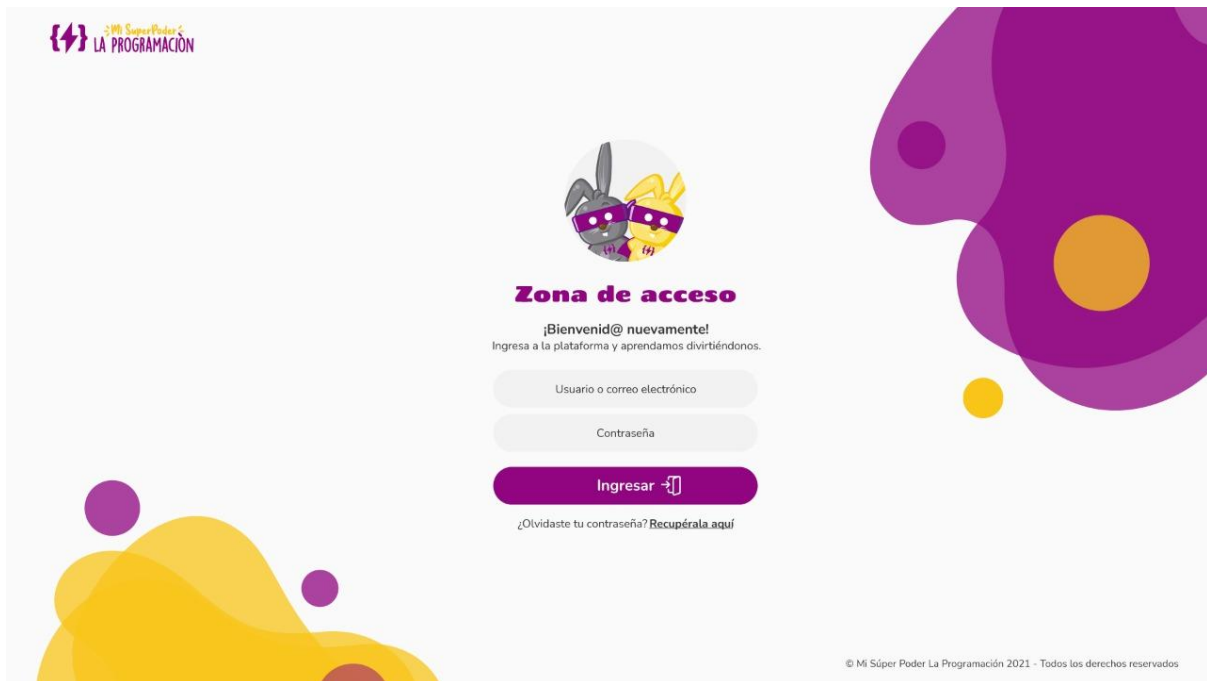
Aquí encontrarás todos los cursos disponibles para ti y tu nivel de aprendizaje. ¡Empieza hoy tu camino en el maravilloso mundo de la programación!

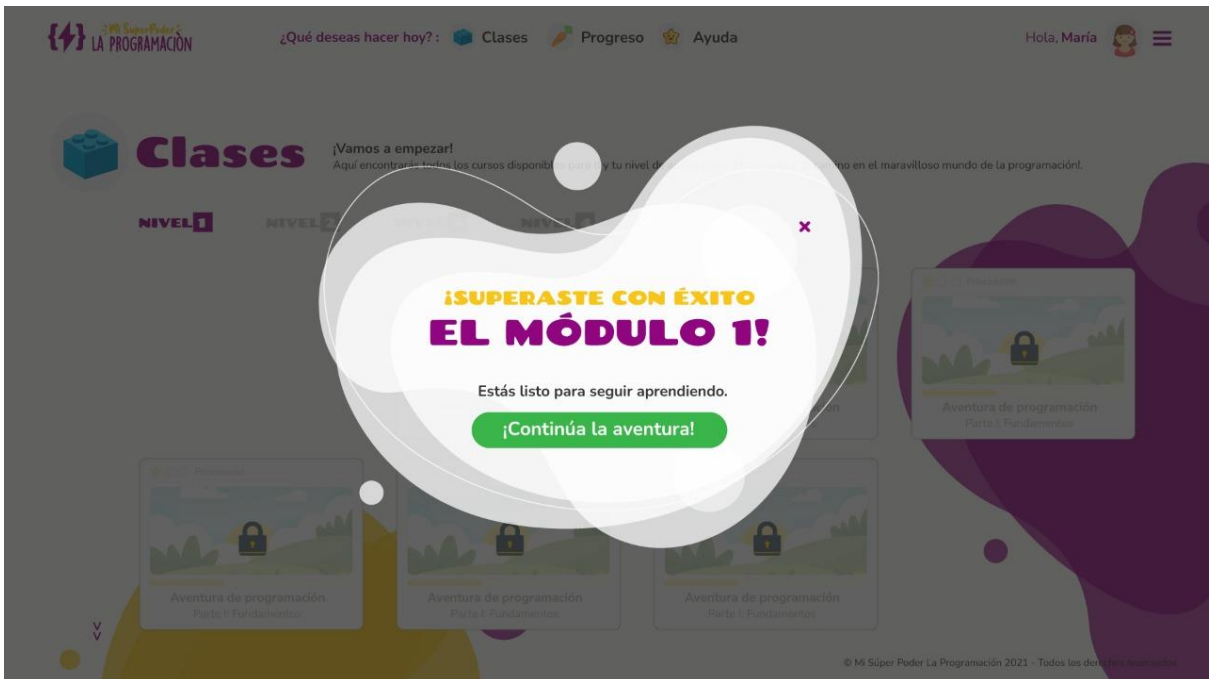
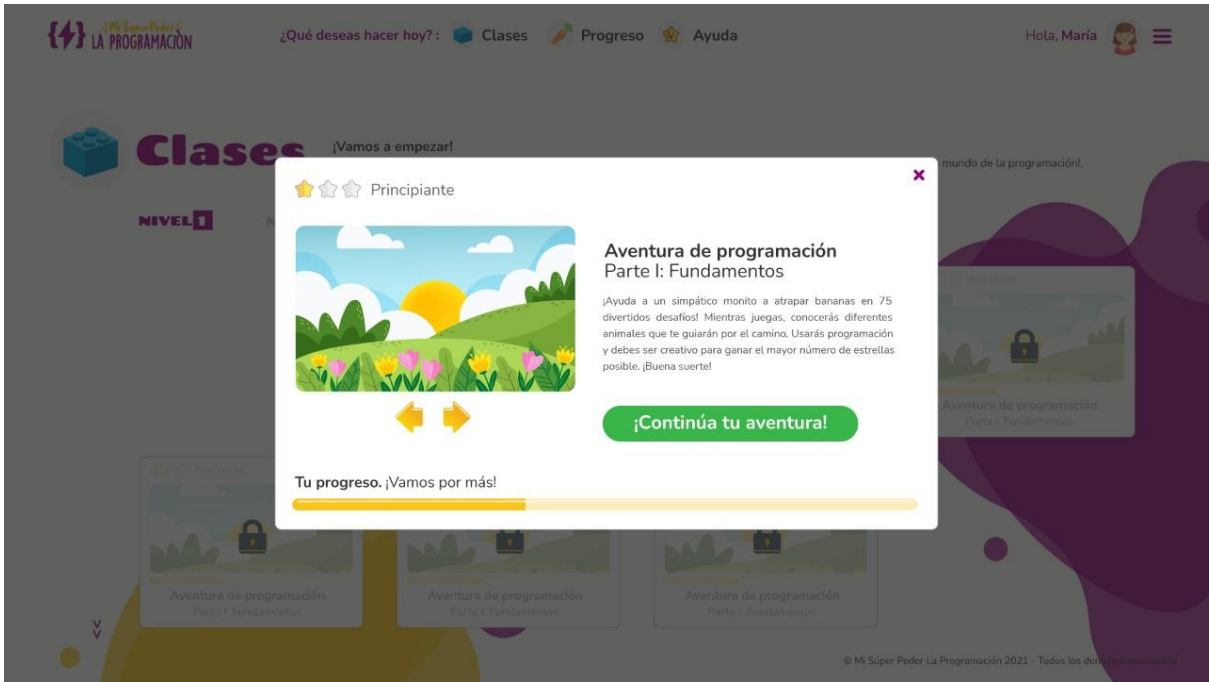
Ver recursos

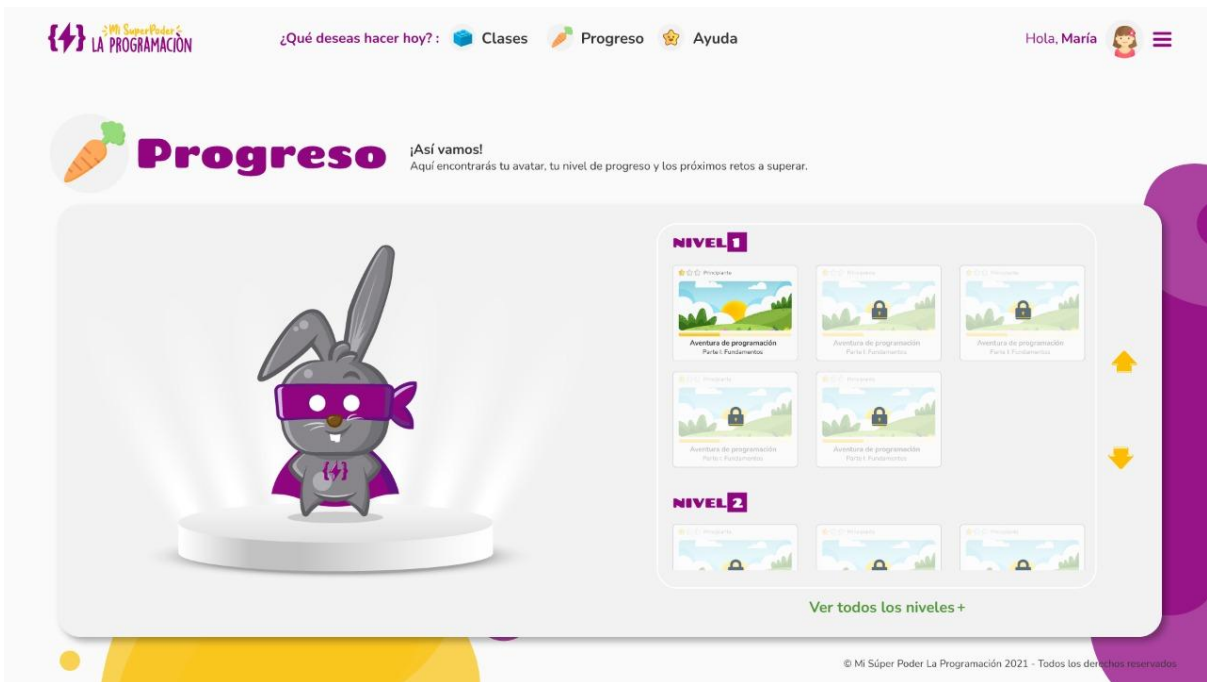
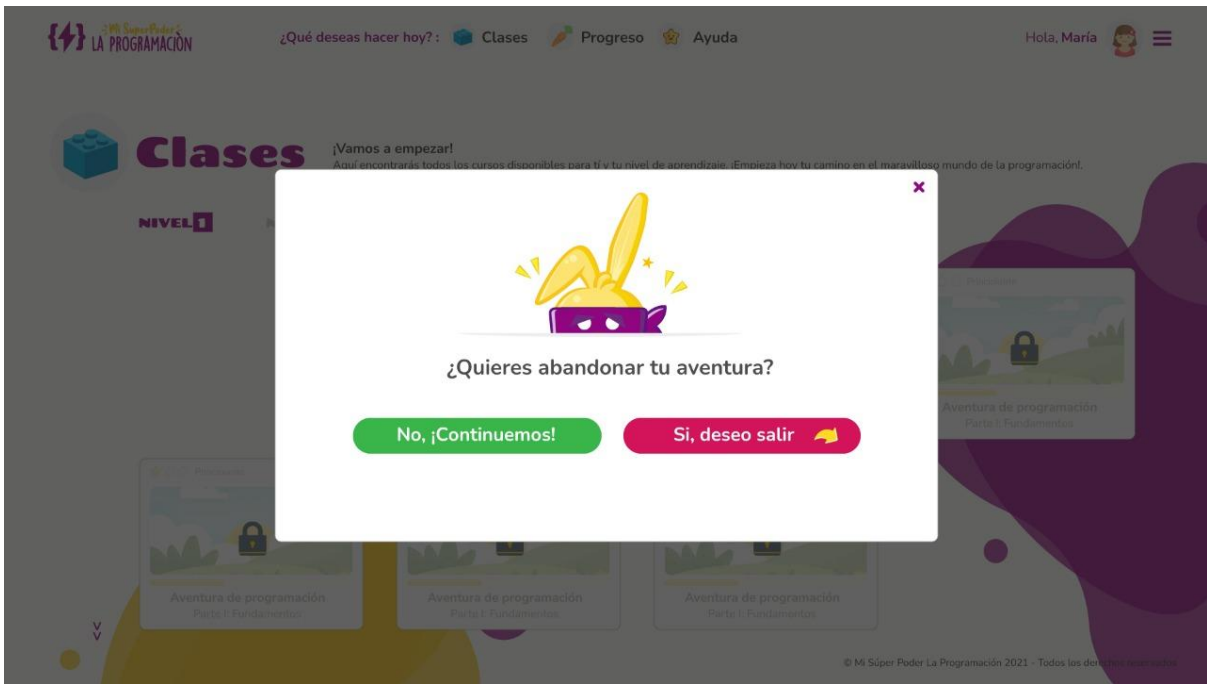
Aquí encontrarás todos los cursos disponibles para ti y tu nivel de aprendizaje. ¡Empieza hoy tu camino en el maravilloso mundo de la programación!

© Mi Súper Poder La Programación 2021 - Todos los derechos reservados

## Diseño UX pantallas rol estudiante







# Ayuda

¿Tienes algún problema con la plataforma?  
Aquí estamos para ayudarte y hacer de tu experiencia de aprendizaje la más divertida.

## Envíanos un mensaje

Tu nombre

Correo electrónico

Tema

Inquietud

ENVIAR

## Preguntas frecuentes

- ¿Cómo puedo registrarme? +
- ¡Es muy fácil!. problema, me podrías ayudar problema, me podrías ayudar problema, me podrías ayudar problema, me podrías ayudar problema, me podrías ayudar problema, me podrías ayudar problema, me kjghabdf jfbfweef.
- ¿Pregunta 2? +
- ¿Pregunta 3? +
- ¿Pregunta 4? +

## ¡Háblanos!

 **María**  
Tengo un problema, me podrías ayudar?

**Operador**  
Claro que sí, cuéntame. 

María está escribiendo...

ENVIAR