

**APLICACIÓN WEB PARA FOMENTAR PENSAMIENTOS  
MATEMÁTICOS DESDE LA GEOMETRÍA PARA ESTUDIANTES  
DE CUARTO Y QUINTO DE PRIMARIA**

**Director(es)**

**Iván Rodrigo Romero Flórez**

**Dianalin Neme**

**Moreno Córdoba Diego Alexander**

**Poveda Girata Hugo Steven**

**Universidad Antonio Nariño**

**Facultad de Ingeniería.**

**Programa de Especialización en Ingeniería de Software**

**Bogotá, Colombia**

**2021**

## **Tabla de contenido**

Resumen	1
Introducción	1
Formulación y descripción del problema	2
Objetivo General	4
Objetivos Específicos	4
Marco de referencia	5
Estado del arte	5
Smartick	8
Khan academy	9
Sesame	10
Impacto	11
Componente de innovación	12
Marco teórico	13
Tecnologías	13
Lineamientos curriculares	14
Estándares Básicos de Competencias en Matemáticas	18
Metodología	21
Descripción de la implementación	22
Proceso de Software	24
Requerimientos	24

Diseño y arquitectura	46
Diagrama de despliegue	48
Caso de uso Arquitecturalmente relevante	49
Diagrama de secuencia	51
Diagrama de clases	54
Arquitectura de alto nivel	56
Construcción	57
área de actividades	57
Adaptador para Router	57
App de backend	58
Pruebas	59
Instalación y Configuración	63
Frontend	63
Backend	64
Conclusiones	66
Referencias	67
Anexo	69

## TABLA DE IMÁGENES

Imagen 1 tomada de <a href="https://www.mathlearningcenter.org/">https://www.mathlearningcenter.org/</a> .....	6
Imagen 2 tomada de <a href="https://www.quickgraph.co">https://www.quickgraph.co</a> .....	7
Imagen 3 encuesta aplicaciones .....	8
Imagen 4 tomado de (smartick, 2021) .....	9
Imagen 5 tomada de (khanacademy, 2021) .....	10
Imagen 6 tomado de (Sesameworkshop, 2021).....	11
Imagen 7 Procesos generales(Elaboración Propia). .....	16
Imagen 8 Pensamientos matemáticos (Elaboración Propia) .....	17
Imagen 9 Competencia matemática (Elaboración Propia) .....	20
Imagen 10 Árbol de Utilidad (Elaboración Propia).....	40
Imagen 11 Diseño y Arquitectura (Elaboración Propia) .....	47
Imagen 12 Diagrama de despliegue (Elaboración Propia).....	49
Imagen 13 Diagrama Casos de uso (Elaboración Propia) .....	50
Imagen 14 DS control de acceso (Elaboración Propia) .....	51
Imagen 15 DS implementación de UI (Elaboración Propia).....	52
Imagen 16 DS Realizar actividad (Elaboración Propia).....	53
Imagen 17 DS Análisis de datos (Elaboración Propia) .....	53
Imagen 18 Diagrama de clases (Elaboración Propia) .....	55
Imagen 19 Diseño alto nivel (Elaboración Propia) .....	56
Imagen 20 (Elaboración Propia) .....	58
Imagen 21 Django Modelo (Elaboración Propia) .....	59
Imagen 22 Prueba registro de usuario (Elaboración Propia) .....	60
Imagen 23 Test de integración (Elaboración Propia) .....	61

Imagen 24 Test Seguridad (Elaboración Propia) .....	62
Imagen 25 Tes Seguridad 2(Elaboración propia).....	62

## ÍNDICE DE TABLAS

Tabla 1 Iteraciones de metodología (Elaboración Propia) _____	23
Tabla 2 Requerimiento de control de acceso (Elaboración propia) _____	26
Tabla 3 Requerimiento registro de usuario (Elaboración Propia) _____	27
Tabla 4 Requerimiento de autenticación (Elaboración Propia) _____	28
Tabla 5 requerimiento de interfaz (Elaboración Propia) _____	29
Tabla 6 Requerimiento de interfaz de estudiante _____	30
Tabla 7 Requerimiento de interfaz de tutores _____	32
Tabla 8 Requerimiento de interfaz de administradores _____	33
Tabla 9 Requerimiento de actividades (Elaboración Propia) _____	33
Tabla 10 Requerimiento carga de actividades (Elaboración Propia) _____	36
Tabla 11 Requerimiento evaluación de actividades (Elaboración Propia) _____	37
Tabla 12 Requerimiento recolección y almacenamiento de actividades (Elaboración Propia) _____	39
Tabla 13 Atributo de calidad seguridad (Elaboración Propia) _____	43
Tabla 14 Atributo de calidad Escalabilidad (Elaboración Propia) _____	46
Tabla 15 Instalación de proyecto frontend (Elaboración propia) _____	64
Tabla 16 Instalación de proyecto backend (Elaboración propia) _____	65

## **RESUMEN**

Se inicia con formulación y descripción del problema, esto permite definir el objetivo general y los objetivos específicos, se analiza el estado de arte, mencionado algunas aplicaciones que se consideraron relevantes, se menciona el impacto él y componente de innovación, todo esto nos permitira tener un marco de referencia sobre el enfoque del proyecto. Luego se mencionan las tecnologías a usar y se menciona el enfoque pedagógico que orientara las actividades del sistema, posteriormente se habla de la metodología que se utilizó y como está al ser ágil se desarrolló en diferentes iteraciones, las cuales se definen. Seguidamente se mencionan todo el proceso de software, el cual abarca desde la toma de requisitos, pasando por el diseño de la arquitectura, algunos patrones de diseño que se utilizaron en la construcción y la representación de diferentes diagramas que permiten tener una mejor vista del sistema. En la parte final se menciona algunas pruebas que se ejecutaron el sistema y los parámetros necesarios para su despliegue.

## **INTRODUCCIÓN**

Actualmente la economía esta transitando de una economía basada en materia prima a una economía basada en los conocimientos y capacidades de una sociedad, además, actualmente gracias a la tecnología es posible generar grandes progresos sin

necesidad de muchos recursos, ya que esta permite generar sistemas cada vez más sofisticados y orientados a resolver necesidades específicas, todo esto no sería posible si no hubiera un conocimiento robusto de las matemáticas, es por esto, y dado que a nivel mundial los países latinoamericanos tienen un desempeño muy bajo, en comparación con otros países, con referencia a la educación matemática, que se pretende desarrollar un sistema orientado a incentivar y cultivar los pensamientos matemáticos, los cuales permiten desarrollar diferentes habilidades, todo esto buscando disminuir las brechas existentes en el uso de estas nuevas tecnologías.

## **FORMULACIÓN Y DESCRIPCIÓN DEL PROBLEMA**

Actualmente en Colombia, la educación matemática tiene grandes retos, esto se puede observar, en el último resultado que se obtuvo en las pruebas PISA, *OECD (2019)*, en donde, los estudiantes colombianos obtuvieron un puntaje promedio de 391, un pequeño avance con respecto a los 390 puntos de media obtenidos en 2015, pero aún muy distanciados del promedio de los países pertenecientes a la Organización para la Cooperación y el Desarrollo Económicos (OCDE) que es de 489 puntos, y mucho más, del primer puesto con 551 puntos.

Además, solo el 35% de los estudiantes se ubicaron en el nivel 2 o superior con una desviación de 40% con respecto a la media OCDE 75% y solo el 1% de los estudiantes alcanzaron el nivel 5 o superior con una desviación de 10% de la media de 11%.

El informe de la OCDE (2019), resalta algunos factores que influyen en estos resultados, sin embargo, sobresalen los siguientes:

- Igualdad en relación con la condición socioeconómica.
- Equidad en relación con el género.

Estos dos factores se pueden resumir en la desigualdad en oportunidades e información que existe, tanto por cuestiones económicas como por razones de género.

Por otro lado, por más de tres décadas la comunidad colombiana de educadores matemáticos ha estado implementando una serie de actividades en pro de la formación matemática de niños y jóvenes (Estándares básicos de Competencia EBC, 2006), y sobre la manera como ésta puede favorecer de forma eficaz a las grandes metas y objetivos de la educación actual.

Para esto, se han desarrollado unos documentos, orientados a responder a las nuevas demandas globales y nacionales, por ejemplo, las relacionadas con una educación para todos, la atención a la diversidad y a la interculturalidad.

Con lo anterior surge la siguiente formulación, ¿Es posible utilizar la tecnología como una herramienta que permita mejorar la calidad de la educación en matemáticas, para los estudiantes colombianos?

Tratando de responder a la pregunta anterior, se diseña una aplicación web donde se implementa un sistema que permite a la profesores definir actividades y a los estudiantes, poder realizarlas de forma un poco más práctica, para esto, el sistema permitirá mejorar la calidad del aprendizaje de los estudiantes colombianos, eliminando



esas brechas de desigualdad que existen actualmente , para ello se apoya, por un lado de la tecnología, la cual permitirá que el acceso sea para todos , y por otro lado, se tomará como referencia el trabajo que ha venido desarrollando la comunidad colombiana de educadores matemáticos, la cual nos garantiza cumplir con las necesidades que actualmente se tiene con respecto a la educación matemática.

### **OBJETIVO GENERAL**

Construir una Aplicación web, que permita aprender y estudiar conceptos de matemáticas, a nivel primaria.

### **OBJETIVOS ESPECÍFICOS**

1. Construir una interfaz donde los estudiantes de manera interactiva puedan aprender conceptos matemáticos.
2. Diseñar un interfaz que sea personalizable para los gustos de cada estudiante.
3. Realizar una interfaz para el docente, donde pueda subir diferentes actividades y obtener informes sobre los avances de los estudiantes.
4. Garantizar la seguridad de la información por medio de token de acceso que me garantice la identidad y privilegios de cada usuario.
5. Implementar una arquitectura de microservicios que permita el correcto funcionamiento de las diferentes funcionalidades del sistema.

6. Diseñar una estrategia de Testing, donde se puedan garantizar una adecuada usabilidad e integración del sistema.
7. Identificar las diferentes alternativas que existen en la web, para el estudio de las matemáticas escolares.

## **MARCO DE REFERENCIA**

### **ESTADO DEL ARTE**

Existen diversas plataformas, tanto web como para dispositivos móviles, para la enseñanza y el desarrollo de las matemáticas en los diferentes niveles. A continuación, se presentan algunas aplicaciones que se creen relevantes por la gran cantidad de usuarios que la utilizan, la primera es creada por National Science Foundation (NSF), la cual bautizaron como Math Learning Center y su misión es inspirar y permitir que las personas descubran y desarrollen su confianza y habilidad en las matemáticas (About MLC | About The Math Learning Center | Math Learning Center, n.d.), por otro lado, tenemos una aplicación, desarrollada por la empresa Educational App Store, la cual ha desarrollado el 8% de 416.000 aplicaciones puestas en la categoría de educación, y en cuanto a matemáticas realizaron una aplicación con temática de panadería para que los niños aprendieran a sumar y restar (*Math Bakery 1 - Start Counting Review* | Educational App Store, n.d.), la siguiente imagen, muestra algunas de las aplicaciones implementadas por Math Learning

Center.



¡ADMITE COMPARTIR!  
**Fractions**

La aplicación Fractions les permite a los estudiantes usar una barra o un círculo para representar, comparar y realizar operaciones con fracciones con denominadores del 1 al 100. Elija el modelo de fracción y el número de partes iguales. Use un color para seleccionar partes específicas para mostrar una fracción del todo.

[Abrir aplicación web](#)  
[Tienda de aplicaciones de Apple](#)  
[Chrome Store](#)



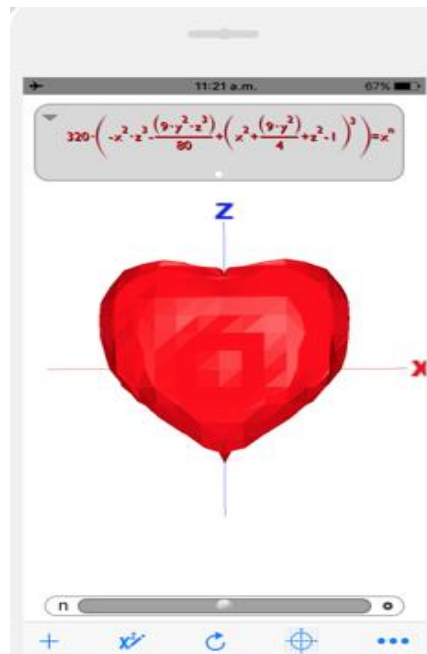
¡ADMITE COMPARTIR!  
**Geoboard**

La aplicación Geoboard es una herramienta para explorar una variedad de temas matemáticos introducidos en los grados de primaria y secundaria. Los alumnos estiran bandas alrededor de las clavijas para formar segmentos de línea y polígonos y hacen descubrimientos sobre el perímetro, el área, los ángulos, la congruencia, las fracciones y más.

[Abrir aplicación web](#)  
[Tienda de aplicaciones de Apple](#)  
[Chrome Store](#)

*Imagen 1 tomada de <https://www.mathlearningcenter.org/>*

A nivel nacional, existen diversos desarrollos para la solución de problemas matemáticos, y aunque no existe alguna que se enfoque a la enseñanza, la aplicación que ha tenido más acogida a nivel nacional e internacional es una aplicación móvil llamada “Quick Graph” disponible solamente para dispositivos con sistema operativo de iOS, y se trata de una calculadora que es capaz de realizar gráficas 2D y 3D de distintas ecuaciones y modelado de sistemas e igualdades. Gracias a su excelente puntuación y comentarios en el App Store, hacen que esta aplicación esté en el TOP de aplicaciones exitosas en este campo. Cabe destacar que tiene un costo de 5 USD (United States Dollar) y ofrece compras dentro de la aplicación (*Quick Graph En App Store*, n.d.).



*Imagen 2 tomada de <https://www.quickgraph.co>*

Teniendo en cuenta la gran cantidad de aplicaciones para fomentar algún tipo de pensamiento matemático que existen actualmente en el ecosistema, se realizó una encuesta dirigida a profesores de matemáticas, en la cual se les preguntó sobre algún tipo de aplicación de software que utiliza en sus clases comúnmente, entre las respuestas obtuvimos.



Si utiliza alguna aplicación web para enseñar alguna área de las matemáticas, cual utiliza

19 respuestas

Desmos

sesameworkshop.org

Quizziz

Geogebra, Khan academy

Probability distribution

Moodle

Classroom y todas las herramientas que ofrece google.

smartick

Classroom

*Imagen 3 encuesta aplicaciones*

Cabe resaltar que se obtuvieron como resultado algunas aplicaciones, como Geogebra (geogebra.org, 2021), que si bien, es un software con muchas funcionalidades y beneficios para el aprendizaje de diferentes conceptos matemáticos, está más orientado a un aprendizaje supervisado por un tutor. a diferencia de las tres siguientes aplicaciones que consideramos las más relevantes en la actualidad.

### **SMARTICK**

Es una aplicación de pago web y móvil española para aprender matemáticas, dirigido a niños de entre 4 y 14 años (smartick, 2021), consiste en actividades presentadas de forma muy interactiva, donde los estudiantes por medio de juegos o preguntas de opción múltiple pueden ir afianzando conceptos matemáticos.

## Matemáticas



El único programa de matemáticas online que combina programación, cálculo y lógica. Sus hijos alcanzarán su máximo potencial.

*Imagen 4 tomado de (smartick, 2021)*

Lo interesante de esta aplicación es que utiliza modelos de machine learning para identificar de forma automática el nivel de cada estudiante y adapta los ejercicios y actividades a resolver, al ritmo de aprendizaje de cada usuario (smartick, 2021).

**KHAN ACADEMY**

Es una aplicación web sin ánimo de lucro orientada hacia el aprendizaje de diferentes disciplinas a nivel escolar entre ellas Matemáticas, cuenta con más de 4.300 vídeo y utiliza algoritmo de recomendación y modelos de machine learning para mejorar la experiencia de los usuarios (khanacademy, 2021).



**Para cada estudiante,  
cada salón de clases.  
Resultados reales.**

Somos una organización sin fines de lucro con la misión de ofrecer educación gratuita de primer nivel, para cualquier persona en cualquier lugar.

*Imagen 5 tomada de (khanacademy, 2021)*

## **SESAME**

Es una organización educativa sin fines de lucro conocida como Sesame Workshop (Sesameworkshop, 2021), la cual actualmente tiene un convenio con IBM para desarrollar diferentes programas utilizando algoritmos de inteligencia artificial, orientados a hacia la enseñanza a nivel escolar.

## Unidad 3 Matemáticas / Sesión 2: Sombras



*Imagen 6 tomado de (Sesameworkshop, 2021)*

Utiliza diferentes herramientas para mejorar la experiencia de los estudiantes desde actividades y videos en su aplicación web, hasta juguetes inteligentes. Sesame e IBM tratan de implementar la computación cognitiva en el aula de clase (pulsosocial, 2016).

### **IMPACTO**



A nivel pedagógico, se espera que este proyecto tenga varios impactos positivos, uno de ellos es la capacidad de orientar pedagógicamente los procesos que viven muchos estudiantes a nivel nacional en su aprendizaje de matemáticas, mostrando un ambiente diferente al que normalmente se vive en un aula de clase, además, de un aprendizaje significativo que permita a los estudiantes medir sus conocimientos y reforzar aquellos que no están tan fuertes. En cuanto a las instituciones educativas, tendrá también un impacto en los docentes, pues este sistema se podrá usar como sistema de calificaciones y monitoreo de los estudiantes, disminuyendo un poco el análisis previo que hacen los profesores en el proceso de cada alumno, siendo el sistema capaz de avisar sobre los estudiantes que tienen deficiencias en ciertas áreas.

A nivel social, también se pretende impactar, pues al pretender aumentar el aprendizaje por las matemáticas, se quiere subir el nivel de los estudiantes de Colombia, obteniendo mejores resultados en las pruebas internacionales, demostrando así, que por medio de la aplicación desarrollada se pueden llenar algunos vacíos conceptuales que no pueden ser subsanados en el aula de clase.

### **COMPONENTE DE INNOVACIÓN**

A diferencia de las aplicaciones mencionadas anteriormente en el marco teórico, tanto nacional como internacionalmente, la aplicación que se pretende desarrollar en este proyecto tiene dos factores diferenciadores y que permiten que tenga diferentes aspectos de innovación.

El primer aspecto de innovación que tendrá este proyecto, a nivel nacional, es el enfoque de integrar actividades que fomenten el desarrollo de los estándares básicos de aprendizaje dados por el ministerio de educación, ya que estos permiten, garantizar los conocimientos mínimos que debe tener un estudiante en cierto nivel de la secundaria, es decir, las actividades que se desarrollen en la aplicación estarán orientadas a fomentar las competencias matemáticas, propuestas por el ministerio de educación, para el alcance de este proyecto y dada la gran cantidad de estándares que existen por cada pensamiento matemático, se abordaran solamente tres, los cuales estarán definidos en el marco teórico.

En segundo lugar, la interface de usuario de la aplicación será total mente personalizable para los gustos de cada estudiante, esto pretende generar una mayor acogida por parte de estos, pues hay aplicaciones con diversas temáticas, pero que, a fin de cuentas, no son del todo personalizadas, pues no permiten que las plantillas de evaluación o aprendizaje sean escogidas por el usuario según su propio gusto o criterio.

## **MARCO TEÓRICO**

### **TECNOLOGÍAS**

#### **➤ DJANGO REST FRAMEWORK**

El marco Django REST es un conjunto de herramientas muy potente y flexible para crear API web. Está basado en el marco de trabajo Django de Python, utiliza la

potencia de Django para crear Apis web utilizando la arquitectura REST, su estructura básicamente utiliza el esquema modelo, vista, serializador, donde el modelo me permite crear por medio de clases la estructura de la base de datos, la vista permite utilizar esos datos por medio de los verbos http y el serializador transforma esos objetos a formatos JSON. (Django REST framework, 2021).

#### ➤ **REACTJS**

ReactJs es una biblioteca para construir interfaces de usuario interactivas de forma sencilla. Sirve, además, para diseñar vistas simples dependiendo de cada estado de la aplicación y generalmente se usa para el desarrollo frontend consumiendo servicios expuestos desde sistemas backend, que es la manera en la que se usará en este proyecto. (React, 2021).

#### ➤ **POSTGRESQL**

Es un sistema de gestión de base de datos relacional orientado a objetos y de código abierto. En el presente proyecto se usará como herramienta para la gestión de las bases de datos y de persistencia, a fin de que, al ser desplegado en un entorno de producción, pueda ser fácilmente implementado, pues es un gestor usado en muchas partes. (Postgresql, 2021).

### **LINEAMIENTOS CURRICULARES**

Son las orientaciones epistemológicas, pedagógicas y curriculares que define el ministerio de educación nacional con el apoyo de la comunidad académica educativa para apoyar el proceso de fundamentación y planeación de las áreas obligatorias y fundamentales definidas por la ley general de educación en su artículo 23.

Los lineamientos curriculares pretenden guiar a los docentes de matemáticas en el diseño e implementación de sus actividades, para facilitar esta orientación se proponen tres aspectos:

➤ **PROCESOS GENERALES**

Que tienen que ver con el aprendizaje, tales como el razonamiento, la resolución y planteamiento de problemas, la comunicación, la modelación y la elaboración, comparación y ejercitación de procedimientos.

Los procesos de mayor importancia a los que apunta los lineamientos curriculares se presentan en la siguiente imagen (Estándares básicos de Competencia EBC, 2006).



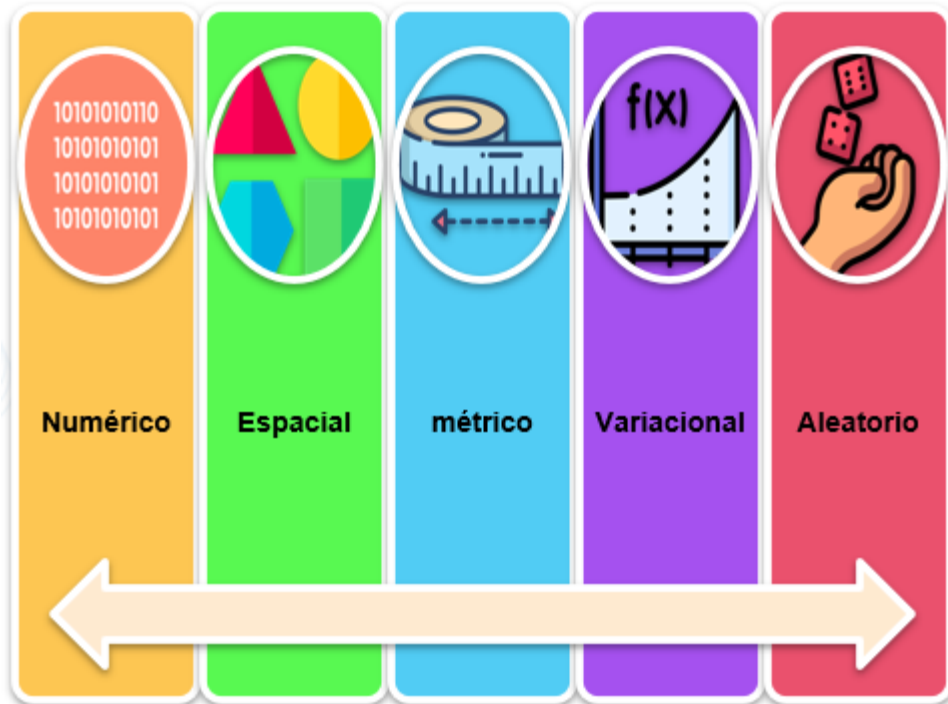
*Imagen 7 Procesos generales(Elaboración Propia).*

➤ **CONOCIMIENTOS BÁSICOS:**

Que tienen que ver con procesos específicos que desarrollen el pensamiento matemático y con sistemas propios de las matemáticas, es decir, son los procesos específicos que desarrollan el pensamiento matemático y con sistemas propios de las matemáticas. Los procesos se vinculan con el desarrollo del pensamiento numérico, el espacial, el métrico, el aleatorio y el variacional. Los sistemas son: sistemas numéricos, sistemas geométricos, sistemas de medida, sistemas de datos y sistemas algebraicos y analíticos.

Los Lineamientos Curriculares, (1998) da al desarrollo del pensamiento matemático, por medio de los pensamientos y sistemas propios de cada área, en la siguiente imagen se

presentas los cinco pensamientos, los cuales no son separados, sino más bien, se entrelazan unos con otros.



*Imagen 8 Pensamientos matemáticos (Elaboración Propia)*

Las actividades que se implementan en la aplicación están orientadas a fomentar el pensamiento Espacial y pensamiento Numérico, entendiendo estos pensamientos como.

- El pensamiento espacial y sistemas geométricos son los que permiten construir y manipular representaciones mentales de objetos del espacio, con el fin de resolver problemas relativos a ubicación, orientación y distribución de espacios (Estándares básicos de Competencia EBC, 2006).

- El pensamiento Numérico centradas en la comprensión del uso y de los significados de los números y de la numeración; la comprensión del sentido y significado de las operaciones y de las relaciones entre números, y el desarrollo de diferentes técnicas de cálculo y estimación (Estándares básicos de Competencia EBC, 2006).

- **EL CONTEXTO**

Tiene que ver con los ambientes que rodean al estudiante y que dan sentido a las matemáticas que aprende (Lineamientos Curriculares Para el área de Matemáticas, 1998), es decir, hace referencia a todos los ambientes que involucran al estudiante y donde toman relevancia las matemáticas que aprende.

Estos aspectos son relevantes dado que Lineamientos Curriculares introduce una visión de las matemáticas escolares centrada en el desarrollo del pensamiento a través de procesos y contextos.

### **ESTÁNDARES BÁSICOS DE COMPETENCIAS EN MATEMÁTICAS**

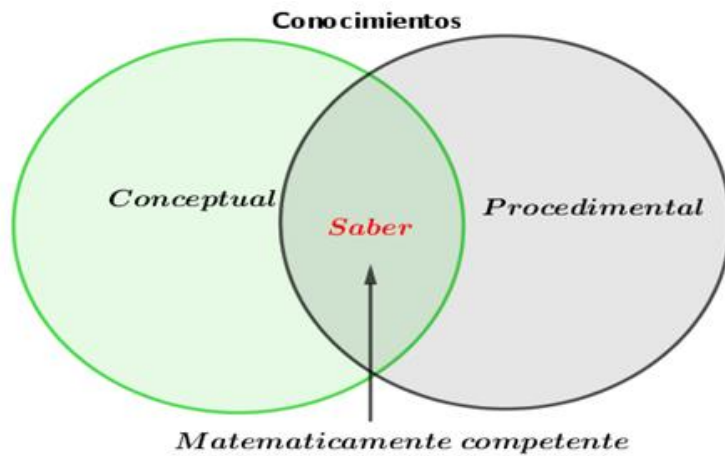
Los estándares básicos de competencia en matemáticas (Estándares básicos de Competencia EBC, 2006) se constituyen como criterios comunes de calidad que expresan una situación deseada en referencia a lo que se espera que todos los estudiantes aprendan en el área de matemáticas.

Por otro lado, en el conocimiento matemático se han distinguido dos tipos básicos en su aprendizaje y enseñanza.

- El conocimiento Conceptual el cual se caracteriza por ser un conocimiento teórico, producido por la actividad cognitiva, muy rico en relaciones entre sus componentes y con otros conocimientos; tiene un carácter declarativo y se asocia con el saber qué y el saber por qué.
- El conocimiento procedimental el cual está más cercano a la acción y se relaciona con las técnicas y las estrategias para representar conceptos y para transformar dichas representaciones; con las habilidades y destrezas para elaborar, comparar y ejercitar algoritmos y para argumentar convincentemente, por lo cual, está asociado con el saber cómo. (Estándares básicos de Competencia EBC, 2006).

Este proyecto está orientado a fomentar un pensamiento matemático en los usuarios, se podría entender el ser matemáticamente competente como la intersección entre los conceptos (sistemas) y lo procedimental (procesos).





*Imagen 9 Competencia matemática (Elaboración Propia)*

Teniendo en cuenta lo anterior, la aplicación está orientada al desarrollo (en su primera versión) de la parte conceptual, utilizando para esto un contexto amigable para los estudiantes y fomentado por medio del proceso general de razonamiento los pensamiento numérico y espacial.

Se desarrollan tres actividades en la aplicación, están enmarcadas en los siguientes estándares para los grados cuarto a quinto.

- Interpreto las fracciones en diferentes contextos: situaciones de medición, relaciones parte todo, cociente, razones y proporciones.
- Justifico el valor de posición en el sistema de numeración decimal en relación con el conteo recurrente de unidades.

Estos dos estándares están orientados a desarrollar el pensamiento numérico (Estándares básicos de Competencia EBC, 2006).

- Utilizo sistemas de coordenadas para especificar localizaciones y describir relaciones espaciales.

Este último estándar, está orientado a desarrollar el pensamiento espacial (Estándares básicos de Competencia EBC, 2006).

## **METODOLOGÍA**

La metodología que se utilizara para realizar este proyecto es Scrumban, entre los beneficios por los cuales se utilizara este marco de trabajo, se tiene (*Metodologías ágiles para la gestión de proyectos / OpenWebinars, n.d.*):

1. Es una metodología ágil que permite a equipos pequeños iniciar con lo que tenga a su disposición. (al ser solo dos integrantes esto nos permitirá enfocarnos en lo que se quiere hacer y no tanto en la estructura de la metodología).
2. Podremos tener un desarrollo continuo de nuestro producto (nos permitirá avanzar con un flujo constante de tareas sin la necesidad de un Sprint).
3. Nos permitirá tener una mayor resiliencia sin depender de estructuras sofisticadas (no permitirá adaptarnos a futuros cambios sin la necesidad de un equipo muy grande).

4. Nos permitirá trabajar de forma descentralizada y autónoma sin perder de vista el resultado en conjunto (lo cual es muy relevante dado el tamaño del equipo y el trabajo remoto).

Se implementará esta metodología utilizando software Trello, donde podremos crear el tablero Kanban, el cual tendrá tres columnas; tareas por hacer, tareas en ejecución y tareas realizadas, por cada iteración, además, las iteraciones serán de dos semanas en donde se realizarán reuniones cada dos días por medio de meet y al final de cada iteración se realizará una reunión de retroalimentación.

#### **DESCRIPCIÓN DE LA IMPLEMENTACIÓN**

La construcción del proyecto se inició el 15 de octubre del 2020 y se acordó dar como fecha limite el 15 de mayo 2021, definiéndose cuatro semanas para cada iteración se definieron los siguientes ciclos, los cuales, al ser implementados en una metodología ágil podría ser modificados dependiendo las circunstancias que se presentaran, además cada iteración está enfocada a implementar uno o varios requisitos funcionales, por otra parte, las últimas iteraciones se orientaron a definir algunas pruebas y a refactorizar el código buscando de esta forma disminuir la deuda técnica que se haya generado.

<b>Iteración</b>	<b>Fecha Inicio</b>	<b>Fecha Termino</b>	<b>Descripción</b>
------------------	-------------------------	--------------------------	--------------------

<b>0</b>	03-07-20	15-10-20	<u>Análisis requerimientos</u> y definición de arquitectura.
<b>1</b>	15-10-20	15-11-20	Control de Acceso y Autenticación.
<b>2</b>	15-11-20	15-12-20	Control de permisos y roles.
<b>3</b>	15-12-20	15-01-21	Receso por festividades de fin de año.
<b>4</b>	15-01-21	15-02-21	Diseño e implementación de las interfaces para los usuarios.
<b>5</b>	15-02-21	15-03-21	Desarrollo de los componentes para el despliegue de las actividades.
<b>6</b>	15-03-21	15-04-21	Etapas de pruebas de calidad.
<b>7</b>	15-04-21	15-05-21	Refactorización de componentes.

*Tabla 1 Iteraciones de metodología (Elaboración Propia)*

Los roles que se tomaron fueron intercambiando, sin embargo, rol de Scrum master fue tomado por Diego Alexander Moreno Córdoba, dada su gran experiencia, y el team fue integrado por diego Moreno y hugo poveda, el rol de Product owner fue tomado por todos los integrantes.

## **PROCESO DE SOFTWARE**

### **REQUERIMIENTOS**

Los requerimientos son de suma importancia en el desarrollo de software. Ya que estos, enfocan y definen lo que se quiere construir, según Fred Brooks (Pressman, 2010) sugiere que *“La parte más difícil al construir un sistema de software es decidir qué construir, ninguna parte del trabajo invalida tanto al sistema resultante si ésta se hace mal. Nada es más difícil de corregir después”*. Dicho de otra manera, los requerimientos son propiedades que debe tener el sistema, las cuales orientan la forma en que se construye dicho sistema.

Entre los requerimientos se tiene, los funcionales y los no funcionales, un requerimiento funcional es una característica de funcionalidad que debe tolerar el sistema y, por otro lado, un requerimiento no funcional es una condición sobre la operación del sistema. (Bruegge & Allen H, 2002).

### **REQUERIMIENTOS FUNCIONALES**

Estos requerimientos describen las relaciones entre el sistema y su entorno, independientemente a su implementación, el entorno incluye a los usuarios y cualquier ente externo al sistema que interactúe con él. (Bruegge & Allen H, 2002).

En este sistema se tomaron los siguientes requerimientos funcionales.

- Control de Acceso y Autenticación.

- Determinar y asegurar la identidad de las personas que acceden a el sistema.
- Definir los privilegios de acceso que tienen los usuarios.
- Garantizar la seguridad de los recursos protegidos.
- Define interfaz para los diferentes usuarios.
- Desplegar la interfaz adecuada a cada usuario.
- Garantizar la individualización de la interfaz para cada usuario.
- Actividades
- Permitir facilidad al implementar nuevas actividades.
- Evaluar de forma constante a los estudiantes.
- Recolectar la información necesaria para el análisis posterior.

Requerimiento	Control de Acceso y Autenticación
Funcionalidad	El sistema registra, identifica y asigna permisos a los diferentes usuarios.
Objetivos	<ul style="list-style-type: none"> <li>● Determinar y asegurar la identidad de las personas que acceden a el sistema.</li> <li>● Definir los privilegios de acceso que tienen los usuarios.</li> </ul>

	<ul style="list-style-type: none"> <li>● Garantizar la seguridad de los recursos protegidos</li> </ul>
Requerimientos funcionales	Registro de usuario
	Autenticar usuarios

*Tabla 2 Requerimiento de control de acceso (Elaboración propia)*

Id	1	Nombre	Registro de usuario
	Entradas	<ul style="list-style-type: none"> <li>● Nombre</li> <li>● Identificación</li> <li>● Correo</li> <li>● Edad</li> <li>● Sexo</li> <li>● Nombre de Usuario</li> <li>● Contraseña</li> <li>● Rol (estudiante, acudiente, profesor)</li> </ul>	
	Salidas	<ul style="list-style-type: none"> <li>● Notificación de registro correcto</li> <li>● Correo electrónico de verificación</li> </ul>	

	<ul style="list-style-type: none"> <li>• El sistema define el rol del usuario</li> </ul>
Precondición:	Dirigirse a la página de registro
Descripción:	El usuario diligenciará un formulario donde ingresa los diferentes datos solicitados.
Postcondición:	<ul style="list-style-type: none"> <li>• Se realizará el registro del usuario y se enviará un email de verificación.</li> <li>• El sistema definirá el rol específico y le otorgará los permisos adecuados.</li> </ul>

*Tabla 3 Requerimiento registro de usuario (Elaboración Propia)*

Id	2	Nombre	Autenticar usuarios
	Entradas	<ul style="list-style-type: none"> <li>• Nombre de usuario (o correo electrónico)</li> <li>• Contraseñas</li> </ul>	
	Salidas	<ul style="list-style-type: none"> <li>• Si la autenticación es válida el sistema enviará al usuario a su home page.</li> </ul>	



	<ul style="list-style-type: none"> <li>• Si la autenticación no es válida el sistema arrojará una alerta con la posible causa (usuario no existente o contraseña incorrecta)</li> </ul>
Precondición:	<ul style="list-style-type: none"> <li>• Estar registrado como usuario</li> </ul>
Descripción:	<p>El usuario ingresará a la page de iniciar sesión en donde ingresará el nombre de usuario y su contraseña.</p> <p>Si la autenticación del usuario no es válida en cuatro intentos consecutivos se enviará un correo para confirmar y establecer credenciales.</p>
Postcondición:	<ul style="list-style-type: none"> <li>• El sistema valida las credenciales y envía una respuesta</li> </ul>

*Tabla 4 Requerimiento de autenticación (Elaboración Propia)*

Requerimiento	Define Interfaz
Funcionalidad	<p>El sistema define e implementa las interfaces para cada usuario, dependiendo de su rol.</p> <ul style="list-style-type: none"> <li>● Para estudiantes despliega las actividades que ha realizado y las actividades por realizar.</li> <li>● Para los tutores permite llevar un seguimiento de los estudiantes que tiene a cargo.</li> <li>● Para los administradores despliega dashboard con los permisos a los que se tiene acceso.</li> </ul>
Objetivos	<ul style="list-style-type: none"> <li>● Desplegar la interfaz adecuada a cada usuario.</li> <li>● Garantizar la individualización de la interfaz para cada usuario.</li> </ul>
Requerimientos funcionales	Desplegar interfaz para estudiantes.
	Desplegar interfaz para tutores.
	Desplegar interfaz para administradores.

*Tabla 5 requerimiento de interfaz (Elaboración Propia)*

Id	3	Nombre	Desplegar interfaz para estudiantes.
Entradas		Tokens identidad	
Salidas		Interfaz personalizada	
Precondición:		Autenticación Exitosa (logueo correcto)	
Descripción:		El estudiante inicia sesión y el sistema lo autentifica, envía un token de identidad, el microservicio lo valida y envía los datos del estudiante; actividades realizadas y por realizar, progreso, sugerencias e insignias logradas.	
Postcondición:		El sistema guarda y actualiza los cambios de la interfaz.	

*Tabla 6 Requerimiento de interfaz de estudiante*

Id	4	Nombre	Desplegar interfaz para para tutores.
Entradas		Tokens identidad	
Salidas		Interfaz personalizada	
Precondición:		Autenticación Exitosa (logueo correcto)	
Descripción:		El tutor inicia sección y el sistema lo autentifica, envía un token de identidad, el microservicio lo valida y envía los datos del tutor; número de estudiantes a cargo, progreso individual, falencias individuales, sugerencias.	
Postcondición:		El sistema guarda y actualiza los cambios de la interfaz.	

Tabla 7 Requerimiento de interfaz de tutores (Elaboración Propia).

Id	5	Nombre	Desplegar interfaz para administradores.
Entradas		Tokens identidad	
Salidas		Interfaz personalizada	
Precondición:		Autenticación Exitosa (logueo correcto)	
Descripción:		El Administrador inicia sesión y el sistema lo autentifica, envía un token de identidad, el microservicio lo valida y envía los datos del administrador; permisos asociados. Interfaz para cargar actividades, eliminar datos, ver registros.	
Postcondición:		El sistema despliega una interfaz actualizada, con los datos y permisos asociados a un administrador	

--	--

*Tabla 8 Requerimiento de interfaz de administradores (Elaboración Propia)*

Requerimiento	Actividades
Funcionalidad	El sistema permite crear, actualizar y eliminar actividades, evaluar actividades y almacenar los datos recolectados en cada actividad
Objetivos	<ul style="list-style-type: none"> <li>● Permitir facilidad al implementar nuevas actividades.</li> <li>● Evaluar de forma constante a los estudiantes.</li> <li>● Recolectar la información necesaria para el análisis posterior.</li> </ul>
Requerimientos funcionales	Carga de actividades.
	Evaluación de actividades.
	Recolección y almacenamiento datos actividades

*Tabla 9 Requerimiento de actividades (Elaboración Propia)*

Id	6	Nombre	Carga de actividades.
Entradas		<p>El sistema permitirá tres tipos de actividades.</p> <ul style="list-style-type: none"> <li>● Actividad para competencias, donde el estudiante por medio de ejemplos y ejercicios conocerá un tema en específico.</li> <li>● Test, donde se evaluará un tema en particular.</li> <li>● Evaluaciones, donde se evaluará de forma general los conceptos trabajados.</li> </ul>	
Salidas		<ul style="list-style-type: none"> <li>● Actividades secuenciales (orientadas al desarrollo de los estándares básicos de competencias en matemáticas).</li> <li>● Test los cuales se realizan después de cada tema visto.</li> </ul>	

	<ul style="list-style-type: none"> <li>● Evaluaciones, las cuales se realizarán cada semana, evaluado los diferentes temas vistos.</li> </ul>
Precondición:	<ul style="list-style-type: none"> <li>● Token de identidad</li> <li>● Interfaz para carga de video</li> </ul>
Descripción:	<p>El Administrador ingresa a la interfaz asociada a carga de actividades e encontrará las opciones para ingresar las actividades del tipo que desea, específicamente el administrador podrá elegir entre.</p> <ul style="list-style-type: none"> <li>● Preguntas de respuesta múltiple</li> <li>● Preguntas de respuesta única.</li> <li>● Poner ejemplos</li> <li>● Ejercicios de arrastre</li> <li>● Ejercicios de operaciones</li> <li>● Ejercicios de conectivos</li> </ul>



	Además, cada actividad debe estar registrada bajo una categoría, que está relacionada con el pensamiento matemático que se quiere potencializar.
--	--

*Tabla 10 Requerimiento carga de actividades (Elaboración Propia)*

Id	7	Nombre	Evaluación de actividades.
	Entradas		El sistema evaluará de dos formas, y cantidad de intentos para una pregunta, las entradas serán los resultados que se obtenga en las respuestas de los estudiantes.
	Salidas		<ul style="list-style-type: none"> <li>● El programa mostrará una barra de progreso, indicando el porcentaje de aciertos en un tema específico.</li> <li>● El sistema recolecta los datos generados y los enviará para su análisis.</li> </ul>

Precondición:	<ul style="list-style-type: none"> <li>● Ingresar a la interfaz de actividades y elegir una.</li> <li>● Realizar por lo menos una actividad.</li> </ul>
Descripción:	<p>El estudiante ingresa a sus actividades. Inicia el desarrollo de las actividades y el sistema guarda las respuestas acertadas y no acertadas, mostrando una barra de progreso donde en porcentajes se muestra el porcentaje de respuestas correctas en una actividad, además expone las actividades fallidas.</p>
Postcondición:	<p>El sistema guarda y actualiza los cambios y envía los datos para su respectivo almacenamiento</p>

*Tabla 11 Requerimiento evaluación de actividades (Elaboración Propia)*

Id	8	Nombre	Recolección y almacenamiento de datos de actividades.
Entradas		Los datos suministrados por los estudiantes.	
Salidas		<ul style="list-style-type: none"> <li>• Ordena los datos y los almacena en una base de datos.</li> </ul>	
Precondición:		los estudiantes deben realizar alguna actividad para recolectar los datos.	

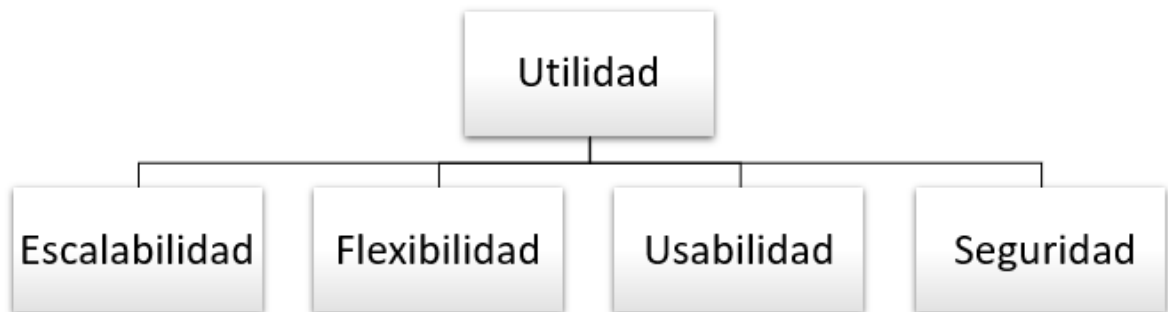
<p>Descripción:</p>	<p>el estudiante realiza una actividad, esta actividad genera los siguientes datos.</p> <ul style="list-style-type: none"> <li>● Número de aciertos.</li> <li>● Número de fallos.</li> <li>● Intentos realizados.</li> <li>● Tiempo dedicado a cada actividad.</li> <li>● Fecha y hora de realizada la actividad</li> <li>● Número de actividades realizadas sobre un mismo tema</li> </ul>
<p>Postcondición:</p>	<p>los datos deben organizarse en una base de datos y enviarse al servicio de análisis de datos.</p>
<p>Criterios De Aceptación</p>	<p>La base de datos puede ser estructurada o no estructurada.</p>

*Tabla 12 Requerimiento recolección y almacenamiento de actividades (Elaboración Propia).*

## REQUERIMIENTOS NO FUNCIONALES

Estos requerimientos definen aspectos del sistema observables y no observables por el usuario que no se relacionan de manera implícita con el comportamiento funcional del sistema (Bruegge & Allen H, 2002), además, los requerimientos no funcionales incluyen restricciones cuantitativas que aclaran criterios para indicar que tan bien satisface un sistema las necesidades de los usuarios o interesados. (Bass, 2013).

La siguiente imagen representa el árbol de utilidad de el sistema, donde cada nodo tiene la misma prioridad.



*Imagen 10 Árbol de Utilidad (Elaboración Propia)*

Teniendo nuestro arbol de utilidad se resaltan los requerimientos no funcionales.

- Escalabilidad.
- Flexibilidad.
- Usabilidad.

- Seguridad.

Estos son los requerimientos no funcionales deseables en el sistema, a continuación, se describen de forma más detallada.

<b>Atributo de Calidad:</b>	<b>Seguridad</b>	
	<b>Descripción</b>	<b>Prioridad</b>
<b>Confidencialidad</b>	<p>El sistema debe permitir autenticar a los usuarios por medio de email, al registrarse los menores deben ser registrados por medio de un acudiente autenticado previamente.</p> <p>La gestión de privilegios de cada acudiente solo le permitirá tener acceso a la información de su encargado y a información</p>	<b>Alto</b>

	<p>general, pero sin nombre detallados</p> <p>La información se almacenará y viajará de forma encriptada</p>	
<p><b>La integridad</b></p>	<p>El sistema de forma periódica debe registrar quién hace qué, cuándo y con qué información.</p> <p>Se realizarán copias de seguridad que permita proteger la información en caso de pérdida, pero además verificar estados anteriores y comprobar posibles manipulaciones.</p> <p>El sistema utilizará la información obtenida de cada estudiante solo para</p>	<p><b>Alto</b></p>

	individualizar su aprendizaje.	
<b>Disponibilidad</b>	<p>El sistema debe de estar disponible un 99% las 24 horas del día durante todo el año.</p> <p>El sistema central debe tener sistemas de tolerancia a fallas en términos de infraestructura, para entrar nuevamente en funcionamiento en menos 25 minutos en caso de una falla.</p>	<b>Medio</b>

*Tabla 13 Atributo de calidad seguridad (Elaboración Propia)*

<b>Atributo de Calidad:</b>	<b>Usabilidad</b>	
	<b>Descripción</b>	<b>Prioridad</b>



<p style="text-align: center;"><b>Usabilidad</b></p>	<p>El sistema debe tener un Interfaz de usuario amigable con los estudiantes y tutores.</p> <p>Los reportes deben ser fáciles de solicitar y deben de ser sencillos de interpretar.</p> <p>La curva de aprendizaje para el uso del sistema debe ser muy baja, su uso debe ser intuitivo.</p>	<p style="text-align: center;"><b>Alto</b></p>
<p style="text-align: center;"><b>Atributo de Calidad:</b></p>	<p style="text-align: center;"><b>Flexibilidad</b></p>	
	<p style="text-align: center;"><b>Descripción</b></p>	<p style="text-align: center;"><b>Prioridad</b></p>
<p style="text-align: center;"><b>Flexibilidad</b></p>	<p>El sistema debe tener la capacidad para agregar, modificar y quitar funcionalidad en los</p>	<p style="text-align: center;"><b>Alto</b></p>

	<p>diferentes componentes sin dañar el sistema.</p> <p>El sistema debe ser capaz de recibir extensiones sin la necesidad de cambiar la arquitectura.</p> <p>El sistema debe permitir actualizaciones de forma sencilla.</p>	
<b>Atributo de Calidad:</b>	<b>Escalabilidad</b>	
	<b>Descripción</b>	<b>Prioridad</b>
<b>Escalabilidad</b>	<p>El sistema debe poder escalar tanto vertical como horizontalmente.</p> <p>El sistema debe poder adaptarse a las diferentes cargas que se le presenten.</p>	<b>Alto</b>

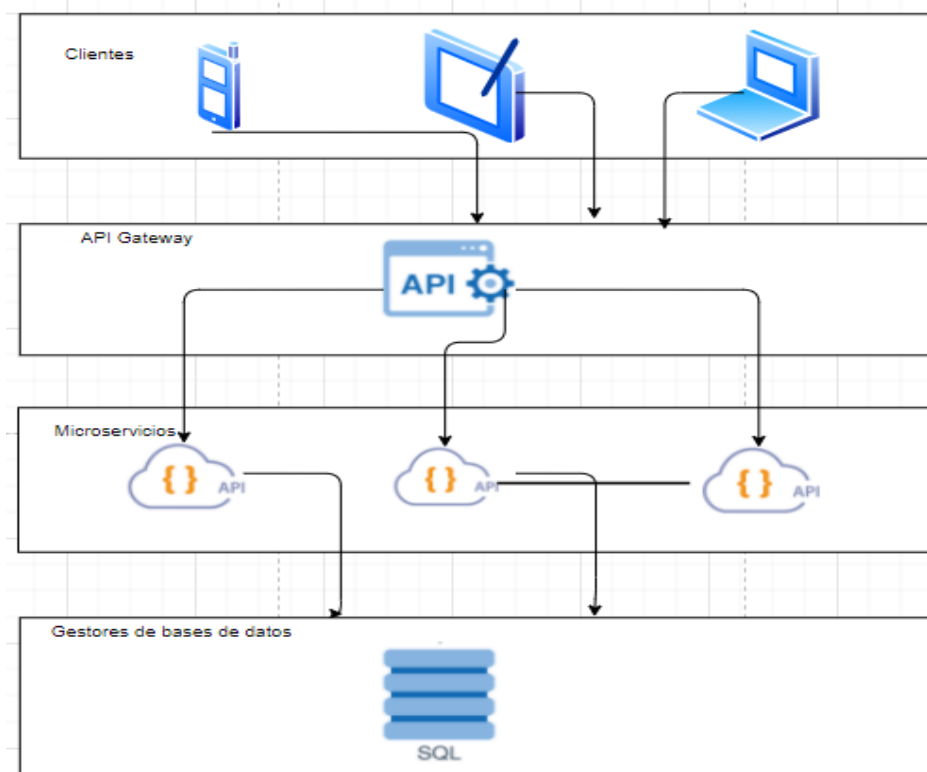
*Tabla 14 Atributo de calidad Escalabilidad (Elaboración Propia)*

## **DISEÑO Y ARQUITECTURA**

Teniendo en cuenta los requerimientos definidos se ha optado por desarrollar un estilo arquitectónico de microservicios, este modelo se puede entender como un enfoque para desarrollar una sola aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y comunicándose con mecanismos ligeros, a menudo una API de recursos HTTP. (Martin Fowler, 2014).

En otras palabras, los microservicios permiten diseñar la aplicación como una recopilación de servicios débilmente acoplados. Los microservicios siguen el modelo share-nothing (arquitectura distribuida en el que cada nodo es independiente y autosuficiente) y se ejecutan como procesos sin estado, este enfoque facilita la ampliación y el mantenimiento de la aplicación (Oracle, 2020).

Para el caso de este sistema se utilizará el siguiente esquema.



*Imagen 11 Diseño y Arquitectura (Elaboración Propia)*

➤ **CAPA CLIENTE**

Esta capa será el frontend de la aplicación y es totalmente independiente de las otras capas, para este proyecto se desarrollará con React.js.

➤ **CAPA API GATEWAY**

Esta capa será la puerta de enlace entre el frontend y los microservicios implementados, su función es la de gestionar la comunicación entre frontend y las APIs

y entre APIs, este módulo será también una API y para este caso será desarrollada con Apache APISIX.

➤ **CAPA MICROSERVICIO**

Esta capa hace referencia a los diferentes microservicios, para este caso a un conjunto de APIs que se comunican con la puerta de enlace o Api gateway, para este sistema se utilizará Django REST framework para construir cada API.

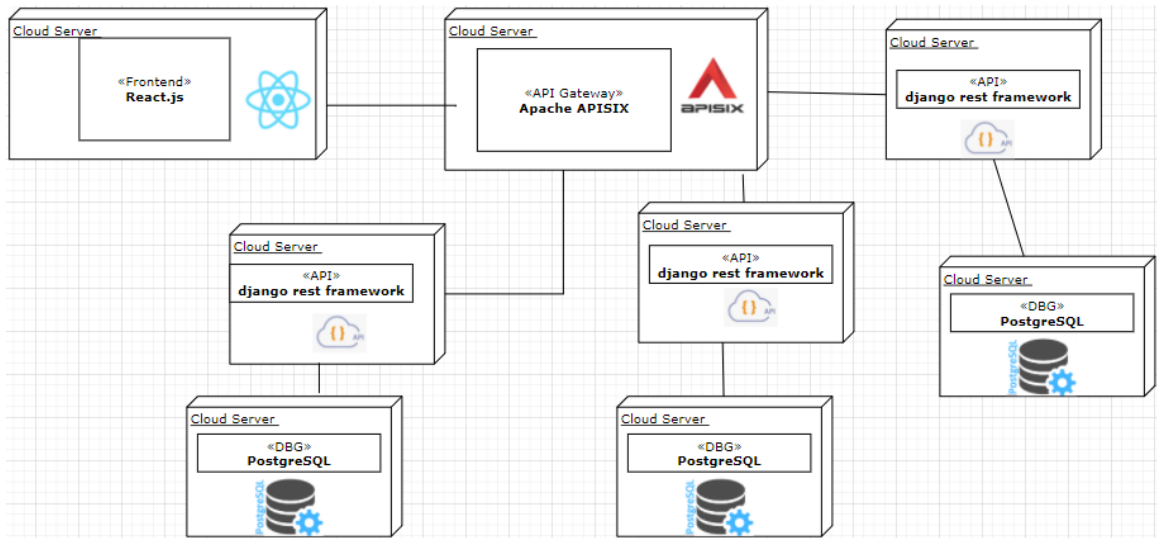
➤ **CAPA GESTOR DE BASES DE DATOS**

Para este sistema utilizaremos PostgreSQL como gestor de bases de datos relacionales, el cual se comunicará con las APIs por medio de un mapeo objeto-relacional o por sus siglas en inglés ORM, el cual está integrando de Django.

**DIAGRAMA DE DESPLIEGUE**

Entre el modelado de diseño de alto nivel se tiene el diagrama de despliegue, el cual nos permite modelar la disposición física de los artefactos de software en nodos hardware. (Carlos, 2010).

Este sistema se desarrolló bajo un modelo orientado a microservicios, por lo que sus diferentes artefactos están desplegados de forma distribuida. en este caso, su implementación se llevará en un Cloud server, la topología del sistema tendrá de la siguiente estructura.



*Imagen 12 Diagrama de despliegue (Elaboración Propia)*

### **CASO DE USO ARQUITECTURALMENTE RELEVANTE**

Los casos de uso son herramientas que modelan los requerimientos funcionales, además esclarece la interacción entre un actor y el sistema, de manera que pueda ser comprendida por cualquier interesado. (Carlos, 2010).

Para esta vista intervienen los siguientes actores: Estudiantes, Profesor, Administradores y Acudientes, estos interactúan con el sistema de la siguiente forma.



Imagen 13 Diagrama Casos de uso (Elaboración Propia)

## DIAGRAMA DE SECUENCIA

El diagrama secuencia o DS muestra la interacción de una colección de objetos en un sistema a través del tiempo, este diagrama resulta ser más cómodo para modelar el diseño, esto ocurre porque al referirnos al diseño, el tiempo adquiere mayor importancia (Carlos, 2010).

Generalmente cada caso de uso viene acompañado de su diagrama de secuencia, sin embargo, se presentan los más relevantes para este sistema. A saber:

- DS control de acceso y autenticación.
- DS implementación de las interfaces de usuarios.
- DS realizar actividades.
- DS análisis de datos.

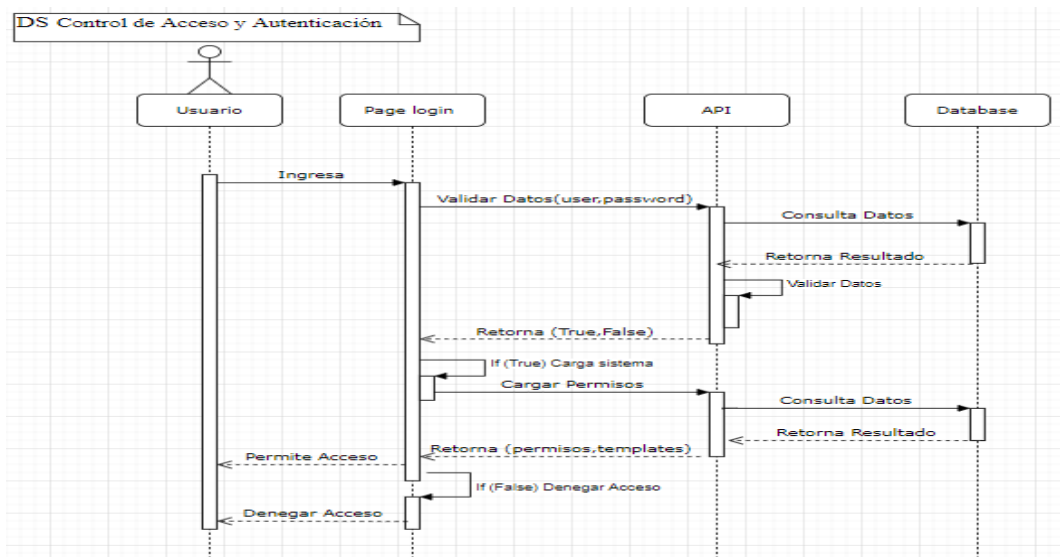


Imagen 14 DS control de acceso (Elaboración Propia)



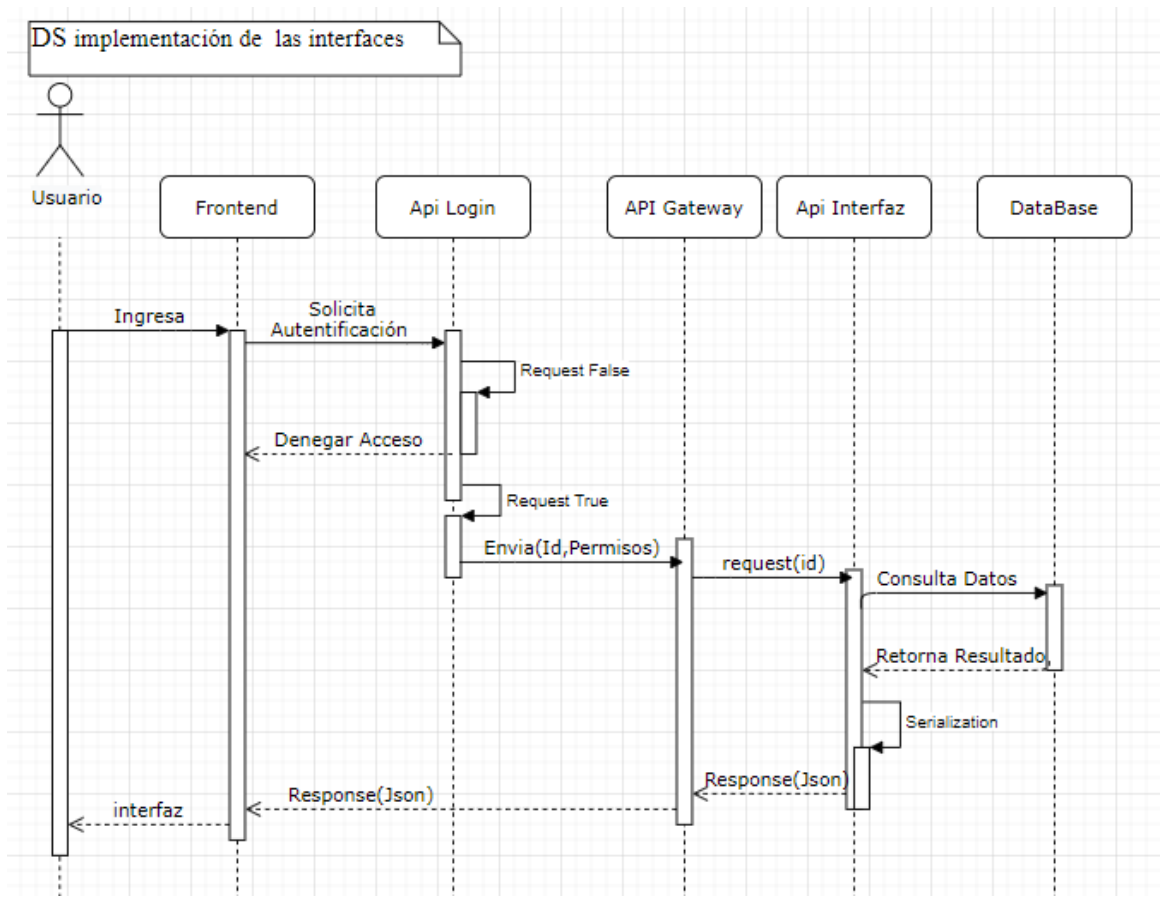


Imagen 15 DS implementación de UI (Elaboración Propia)

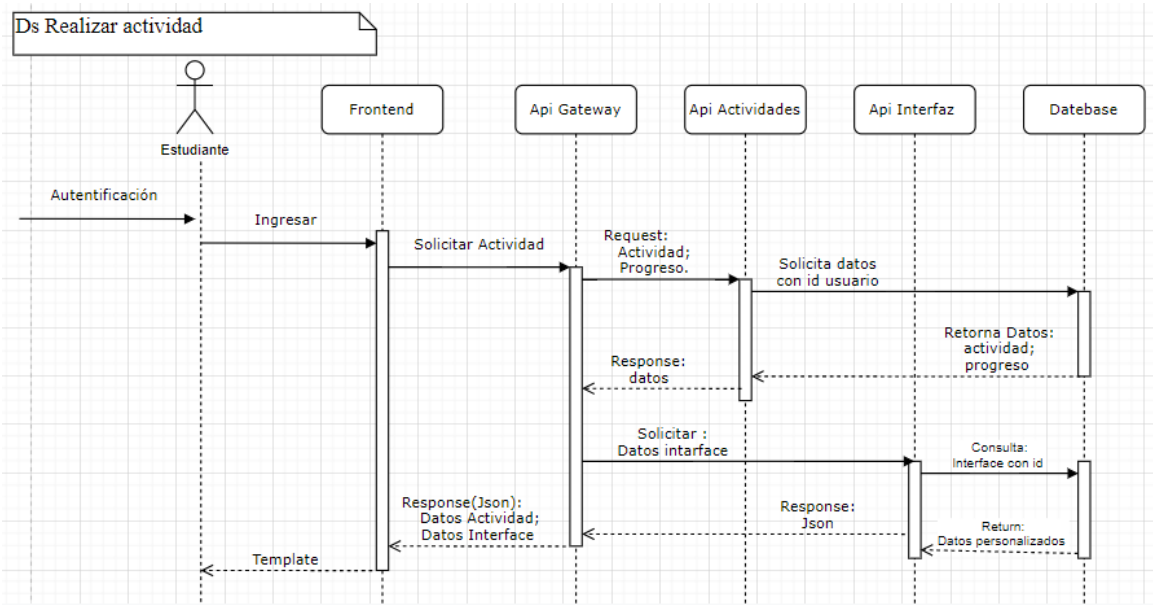


Imagen 16 DS Realizar actividad (Elaboración Propia)

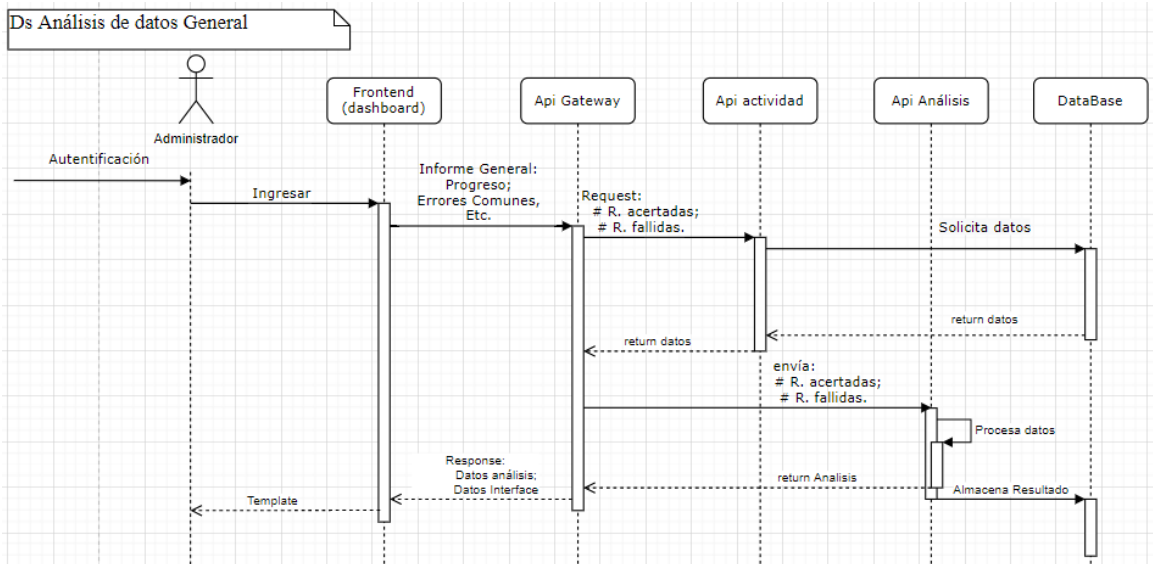


Imagen 17 DS Análisis de datos (Elaboración Propia)

## **DIAGRAMA DE CLASES**

Este diagrama representa la estructura del sistema desde el punto de vista de las clases, atributos, operaciones y asociaciones. (Bruegge & Allen H, 2002), los diagramas de clase se usan para representar modelos de objetos durante el desarrollo. En otras palabras, es un tipo de diagrama que representa el sistema exponiendo sus clases, sus atributos y métodos, y las relaciones que existen entre ellas, además, teniendo en cuenta que las clases son un pilar de la programación orientados a objetos, los diagramas de clases son una de las representaciones estructurales más importantes, a la hora de modelar un diseño detallado (Carlos, 2010).

Para este sistema se definieron las siguientes clases; Messages, User, Student, Teacher, Responsable, Administrador, interfaces, Modules, Evaluation modules, Learning modules. sus relaciones se pueden observar en el siguiente diagrama.

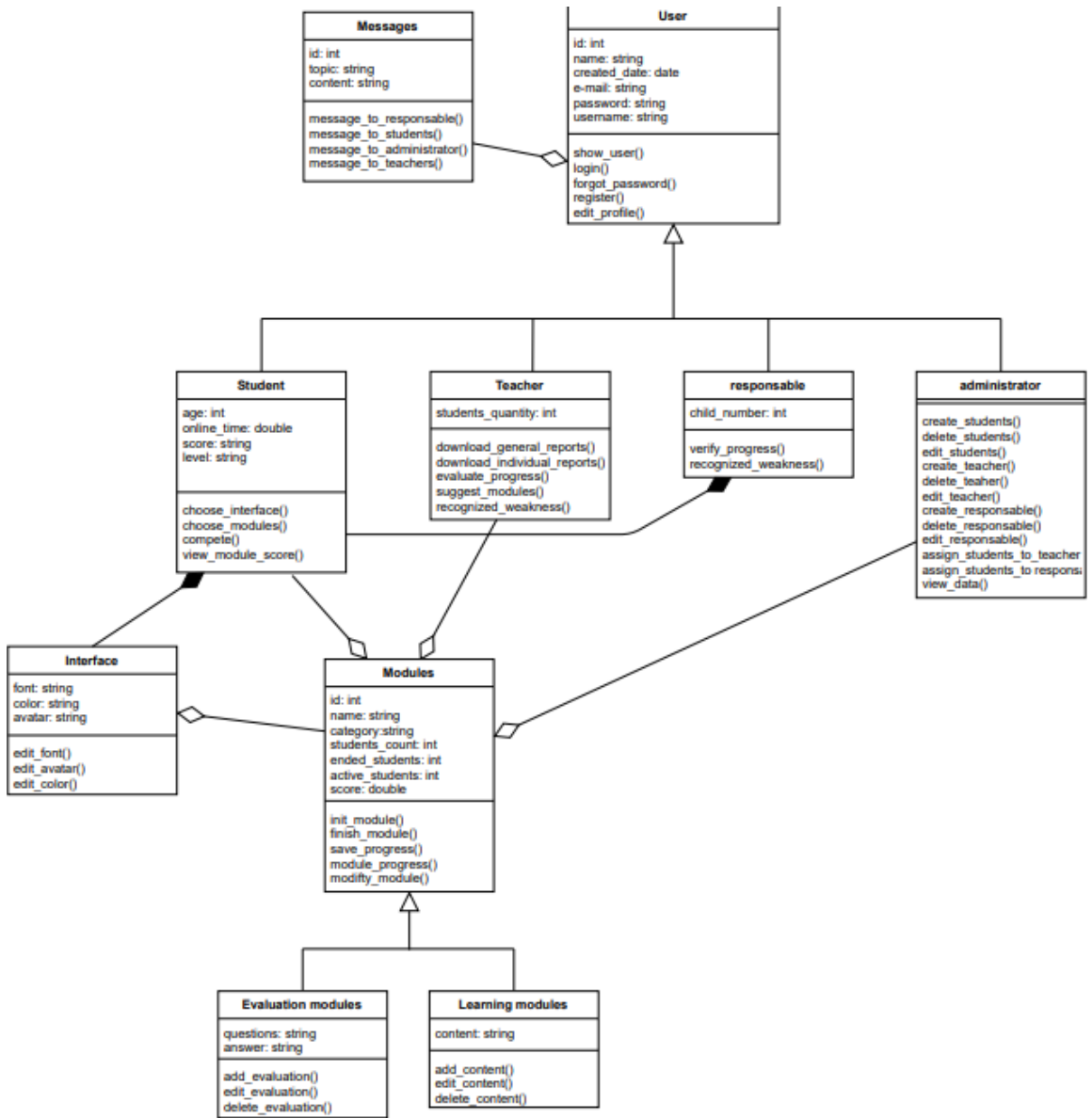


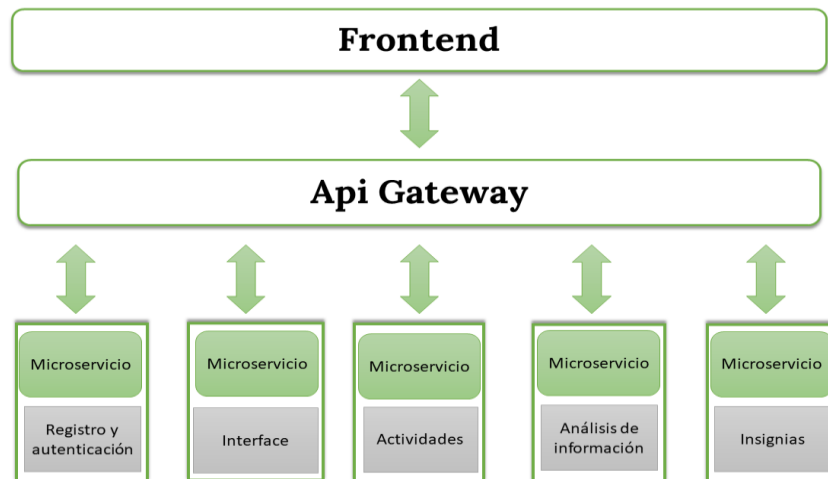
Imagen 18 Diagrama de clases (Elaboración Propia)

## ARQUITECTURA DE ALTO NIVEL

El diseño de alto nivel o diseño arquitectónico tiene como principal tarea especificar la organización del principal de los componentes de un sistema, las relaciones entre ellos y el ambiente y los principios que permiten su diseño y evolución (Fuentes, 2019).

Es decir, un diseño de alto nivel nos permite ver los componentes de nuestro sistema como un todo, poder abstraer la organización del software determinado los componentes que definirán el sistema y cómo estos se relacionan, sin especificar su implementación.

La arquitectura de alto nivel para este sistema es la de microservicios, la cual, permite el desarrollo y despliegue de aplicaciones definidas por unidades independientes, autónomas, modulares y autocontenidas (López & Edgar, 2017), el patrón básico para la arquitectura de este sistema es el siguiente.



*Imagen 19 Diseño alto nivel (Elaboración Propia)*

## **CONSTRUCCIÓN**

Este proyecto necesitó de la construcción de varias unidades, pero en aspectos generales se describe la forma como se construyeron algunos de estos componentes.

### **ÁREA DE ACTIVIDADES**

la idea de este componente se basa en una clase para construir las actividades de forma sencilla, la parte del frontend se desarrolló utilizando un componente de clase de React, este componente permite crear diferentes instancias para las diferentes actividades, la forma en que está diseñado es utilizando el patrón Bridge, en donde por un lado se creó una clase encargada de la abstracción del componente y por otro una clase encargada de implementación, además, al tener esta unidad varios componentes como botones, alertas, grid entre otros, se utilizó la librería para react Material-UI, por lo cual, se debió implementar un diseño parecido al patrón fachada, donde por medio de un objeto se pudo establecer una conexión entre los componentes de la librería Material-UI y nuestra clase para crear actividades.

En la construcción de este componente se utilizó una refactorización continua, en donde, primero se creó un modelo que fuera funcional y por medio de diferentes refactorizaciones se buscó mejorar el código sin cambiar la funcionalidad.

### **ADAPTADOR PARA ROUTER**

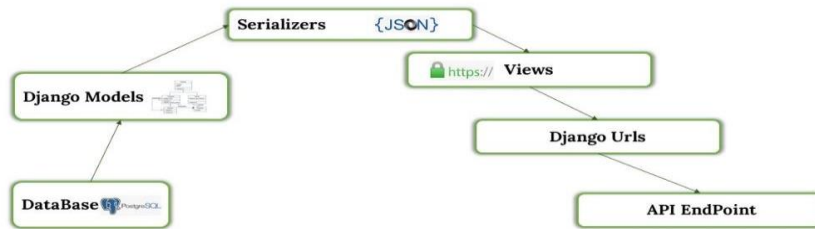
El proyecto al tener varias rutas y permisos diferentes para cada usuario en dichas rutas, se implementó un patrón de diseño entre mediador y fachada, donde por medio de un componente que interactúa con la librería react-router se gestiona todo los enrutamientos y además, toda la interacción con react-router es simplificada ya que solo se interactúa con el componente creado, de esta forma se pudo construir un componente que facilita la interacción con la librería y además con una única responsabilidad, la de administrar los routers.



*Imagen 20 (Elaboración Propia)*

#### **APP DE BACKEND**

Para el backend se utilizó el modelo que viene por defecto en Django Rest Framework el cual es el siguiente.



*Imagen 21 Django Modelo (Elaboración Propia)*

Por ejemplo, se construyó una App de Django con el nombre `custom_auth`, en esta App se creó el modelo para los usuarios, lo cual gracias al ORM interno en Django permite la transformación de nuestra clase `User`, a una tabla en nuestra base de datos relacional denominada `User`.

## PRUEBAS

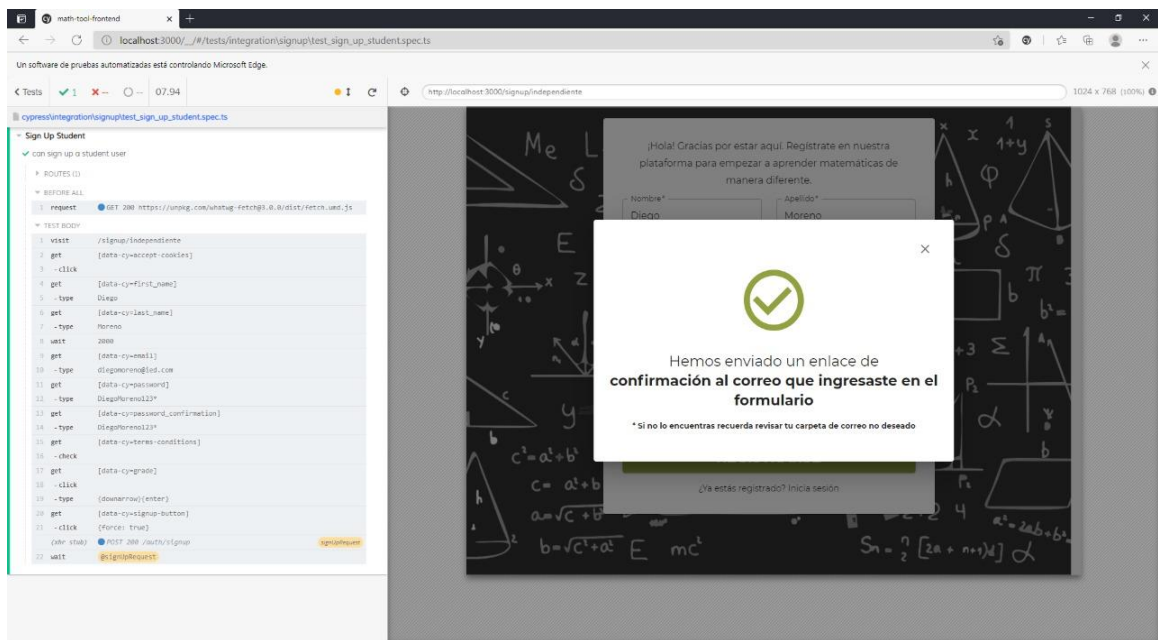
Entendiendo las pruebas como la actividad que se lleva a cabo para valorar la calidad de un producto y optimizarlo mediante la identificación de sus defectos y problemas (Bourque & Fairley, 2014). Se pueden clasificar las pruebas en dos grupos (Whittaker, 2002)

- Pruebas funcionales y estructurales, también llamadas de caja negra y caja blanca.
- Pruebas unitarias, de integración y de sistema.



Para satisfacer las necesidades de calidad, se implementaron distintas pruebas al proyecto frontend y backend, con el fin de testear la usabilidad, integración y seguridad de la aplicación. En el caso del proyecto frontend, se hicieron las pruebas de usabilidad e integración mediante la librería de javascript “Cypress” la cual, testea las peticiones y respuestas desde y hacia el servidor, lo que permitió testear el registro exitoso del usuario, su autenticación y además que el cambio de interfaz se haga correctamente.

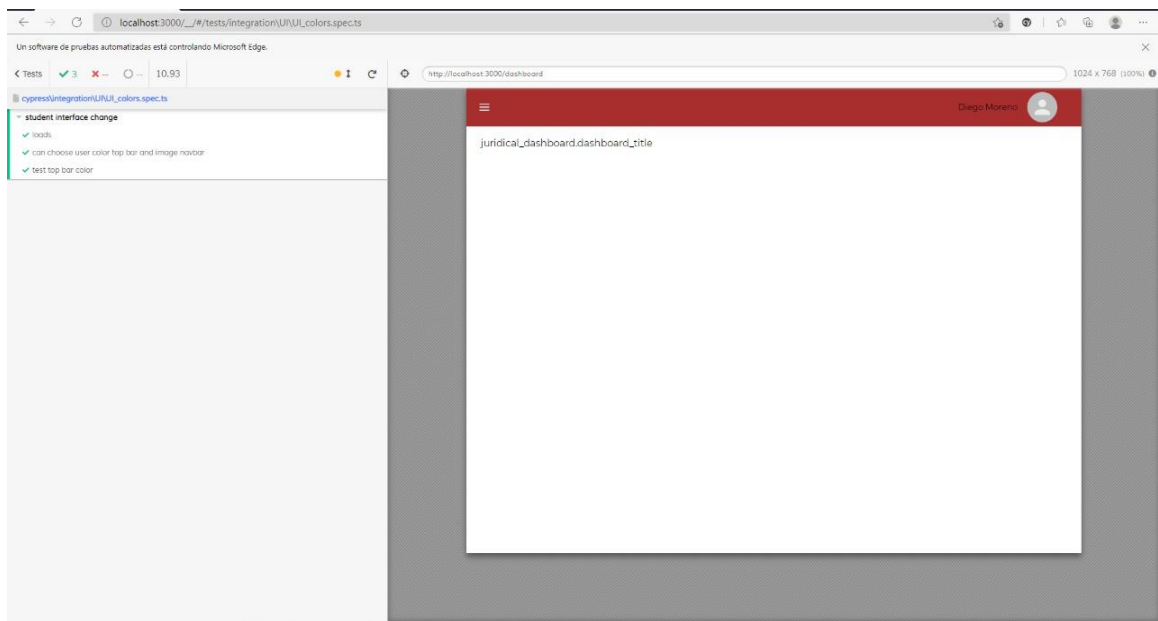
A continuación, se mostrará la prueba del registro de usuario.



*Imagen 22 Prueba registro de usuario (Elaboración Propia)*

En este caso, lo importante es que cada una de las restricciones del formulario de registro se cumplan, tales como, que el correo sea un correo válido, que se pueda elegir un curso, y que las contraseñas ingresadas coincidan.

A continuación, se muestra el segundo test de integración, el cual está dirigido a que el usuario pueda elegir la interfaz correctamente y, además, que los colores elegidos se hayan elegido correctamente como se muestra a continuación.



*Imagen 23 Test de integración (Elaboración Propia)*

Ahora, se muestra el test de seguridad del backend que se hizo mediante los verbos when-given-then y que nos ayuda a tener parámetros establecidos para los tests.

```

10 class SignUpTests(APITestCase):
11
12     def test_sign_up_student_successfully(self):
13         # given
14         Grade.objects.create(name='Cuarto')
15
16         # when
17         url = reverse('signup')
18         data = {'first_name': 'Diego', 'last_name': 'Moreno', 'email': 'diegomoreno@ied.com',
19               'password': 'DiegoMoreno123*', 'password_confirmation': 'DiegoMoreno123*',
20               'grade': {'label': "Cuarto", 'value': '1'}}
21         response = self.client.post(url, data, format='json')
22
23         # then
24         self.assertEqual(response.status_code, status.HTTP_201_CREATED)
25         self.assertEqual(response.data, {'email': 'diegomoreno@ied.com'})
26         self.assertEqual(Student.objects.count(), 1)
27         self.assertEqual(Student.objects.last().user.first_name, 'Diego')
28         self.assertEqual(Student.objects.last().user.email,
29                          'diegomoreno@ied.com')
30
31     def test_sign_up_student_is_not_successfully(self):
32         # given
33         email = 'diegomoreno@ied.com'
34         password = 'DiegoMorenod123*'
35         User.objects.create_user(username=email, email=email, password=password)
36
37         # when
38         url = reverse('signup')
39         data = {'first_name': 'Diego', 'last_name': 'Moreno',
40               'email': 'diegomoreno@ied.com', 'password': 'DiegoMorenod123*',
41               'password_confirmation': 'DiegoMorenod123*'}
42         response = self.client.post(url, data, format='json')
43
44         # then
45         self.assertEqual(response.status_code, status.HTTP_400_BAD_REQUEST)
46         self.assertEqual(response.data, {'email': ['Correo ya registrado']})

```

*Imagen 24 Test Seguridad (Elaboración Propia)*

Por último, se muestra el test de seguridad el cual se realiza en el backend, y se encarga de mostrar que la autenticación y registro se haga correctamente, y se tenga en cuenta restricciones tales como, que no pueda registrarse un usuario creado, y que los datos recibidos en la API, correspondan a los mínimos para poder tener un usuario.

```

PROBLEMS 8 OUTPUT TERMINAL DEBUG CONSOLE
(venv) PS C:\Users\Diego Moreno\Documents\proyecto math tool\math-tool-backend> python manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 0.276s

OK
Destroying test database for alias 'default'...

```

*Imagen 25 Tes Seguridad 2(Elaboración propia)*

## INSTALACIÓN Y CONFIGURACIÓN

La configuración está basada tanto para el backend como para el frontend, los cuales se podrán desplegar en diferentes maquinas.

### FRONTEND

Para implementar la parte del cliente de este proyecto se deben seguir estas instrucciones.

Paso	Nombre	Descripción
1	Requisitos	La máquina debe tener instalado Git y NodeJS $\geq 12$ , Git permitirá interactuar con el repositorio y Node ejecutar los componentes.
2	Clonación de proyecto	Por medio de los comandos de Git clonar este repositorio <a href="https://github.com/UAN-software-engineering/math-tool-frontend.git">https://github.com/UAN-software-engineering/math-tool-frontend.git</a> .
3	Configuración	Actualice el archivo de env.development acuerdo con su configuración local. De esta forma se dan los permisos para la comunicación con la IP del backend.

4	Dependencias	Estas son las librerías que se utilizan en el proyecto, para descargarlas en la maquina solo debe ejecutarse el comando yarn install o npm install.
5	Iniciar servidor	Esto permite correr el frontend en un servidor propio. Para esto se ejecuta yarn start.

*Tabla 15 Instalación de proyecto frontend (Elaboración propia)*

## **BACKEND**

Los siguientes pasos crean el servidor del backend de la aplicación

Paso	Nombre	Descripción
1	Requisitos	La máquina debe contar con Python 3.7.0 y un sistema de gestión de paquetes para Python, se recomienda pip =>18.0, además, una herramienta para crear un entorno privado de Python Virtualenv =>20.0
2	Clonación de proyecto	Por medio de los comandos de Git clonar este repositorio <a href="https://github.com/UAN-software-engineering/math-tool-backend">https://github.com/UAN-software-engineering/math-tool-backend</a>

3	Entorno Virtual	Dentro de la carpeta raíz ejecutar el siguiente Scripts  py -3 -m venv venv, esto creará un entorno virtual en la máquina.
3	Configuración	Actualice el archivo de env.development acuerdo con su configuración local. De esta forma se dan los permisos para la comunicación con la IP del frontend.
4	Dependencias	Estas son las librerías que se utilizan en el proyecto, para descargarlas en la máquina solo debe ejecutarse el comando  pip install -r requirements.txt
5	Iniciar servidor	Para inicializar el servidor debe entrar en el entorno virtual por medio venv\Scripts\activate y luego ejecutar Python  manage.py runserver.

*Tabla 16 Instalación de proyecto backend (Elaboración propia)*

## CONCLUSIONES

- La toma de requerimientos tanto funcionales como no funcionales son indispensables para el desarrollo de cualquier proyecto de software, la arquitectura que se toma para este proyecto fue de microservicios, sin embargo, una arquitectura de capas hubiera sido suficiente.
- La implementación correcta de la metodología debe ser estricta, ya que esta nos ayuda en los momentos más difíciles.
- El desarrollo de pruebas debe ir en paralelo o muy cerca a la hora de escribir código, ya que estas pruebas, primero nos dan un estado de confianza en la validez de programa y segundo no ahorra mucho tiempo en etapas más avanzadas.
- Los aspectos de seguridad deben definirse al inicio de cualquier proyecto, esto no permitiera crear sistemas más confiables para nuestros usuarios y evitar dolores de cabeza en una etapa más avanzada.
- Las habilidades blandas se deben trabajar igual o un poco más que las habilidades técnicas, ya que estas habilidades nos permitirán crear mejores proyectos.

- El sistema de software debe diseñarse pensando en el cambio. Los sistemas deben estar contruidos bajo buenas prácticas orientado a tener alta cohesión y bajo acoplamiento, ya que esto permitira a largo plazo una mejor mantenibilidad del código.

## REFERENCIAS

Bass, L. (2013). *Software architecture in practice, 3rd ed.* NJ,USA: Addison-Wesley.

Bourque, p., & Fairley, R. (2014). *Swebok V3.0 Guide to the software engineering body of knowledge.* IEEE Computer Society Press.

Bruegge, B., & Allen H, D. (2002). *Ingeniería de Software Orientado a Objetos.* México: Pearson.

Carlos, F. (2010). *UML Modelado de Software para Profesionales. (1ª Ed.)*. Buenos Aires : Alfaomega .

docs.oracle. (2020). *oracle*. Retrieved from <https://docs.oracle.com/es/solutions/learn-architect-microservice/index.html#GUID-BDCEFE30-C883-45D5-B2E6-325C241388A5>

(2006). *Estándares básicos de Competencia EBC.* bogota: MInisterio de educacion nacional .



framework, D. R. (2021, 02 15). *Django REST framework*. Retrieved from <https://www.django-rest-framework.org/>

Fuentes, M. d. (2019). *Fundamentos de ingeniería de software*. México D.F: UNIVERSIDAD AUTONOMA METROPOLITANA.

Gardner, H. (1987). *Estructuras de la ment:La teoria de las inteligencias Multiples*. México : Fondo de cultura Economica.

geogebra.org. (2021). *geogebra.org*. Retrieved from geogebra.: <https://www.geogebra.org/>

khanacademy. (2021). *khanacademy.org*. Retrieved from khanacademy.org

(1998). *Lineamientos Curriculares Para el área de Matemáticas* . Bogota: Ministerio de Educación Nacional.

*Martin Fowler*. (2014, Marzo 25). Retrieved from [martinfowler.com](https://martinfowler.com/articles/microservices.html): <https://martinfowler.com/articles/microservices.html>

Postgresql. (2021, 05 07). *Postgresql*. Retrieved from <https://www.postgresql.org/>

Pressman, R. S. (2010). *Ingeniería del software : Un enfoque práctico*. México D.F: Mc Graw Hill.

pulsosocial. (2016). *pulsosocial*. Retrieved from <https://pulsosocial.com/2016/04/27/ibm-plaza-sesamo-computacion-cognitiva-educacion-preescolar/>

React. (2021, 05 7). *react js*. Retrieved from <https://reactjs.org/docs/getting-started.html>

Sesameworkshop. (2021). *sesame*. Retrieved from <https://www.sesameworkshop.org/>

smartick. (2021). *smartick*. Retrieved from <https://co.smartickmethod.com/>

Whittaker. (2002). *How to break software: A practical guide to testing*. Addison-Wesley.

## **ANEXO**

- Document sad
- Modelo canvas