



Aplicación web que permite la búsqueda centralizada y actualizada de los colegios a nivel Bogotá, Busco colegio

Luis Andrés Olarte Zabala

10892116988

Luis Eduardo Velasco Acero

9434724

Universidad Antonio Nariño

Especialización en Ingeniería de Software

Facultad de Sistemas

Bogotá, Colombia

2021

Busco Colegio

Luis Andrés Olarte Zabala, Luis Eduardo Velasco Acero

Proyecto de grado presentado como requisito parcial para optar al título de:

Especialización en Ingeniería de Software

Director (a):

MSC. Dianalin Neme Prada

Codirector (a):

MSC. Iván Romero Flórez

Universidad Antonio Nariño

Especialización en Ingeniería de Software

Facultad Sistemas

Bogotá, Colombia

2021

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

Cumple con los requisitos para optar

Al título de _____.

Firma del Tutor

Firma Jurado

Firma Jurado

Bogotá, Día Mes 2021

Contenido

	Pág.
Contenido.....	4
Índice de ilustraciones	6
Índice de tablas.....	6
Resumen.....	7
Abstract.....	8
Formulación y descripción del problema	11
Objetivos	13
Objetivo general	13
Objetivos específicos	13
Marco de Referencia.....	14
Estado del Arte.....	15
Impacto	17
Componente de Innovación.....	19
Marco teórico	21
Tecnologías.....	21
Buscador Asistido	22
Metodología	24

Estados:.....	24
Niveles de prioridad:	24
Roles: 24	
Reuniones	25
Proceso de software.....	26
Requerimientos funcionales	27
Requerimientos no funcionales	32
Diseño y arquitectura	36
Diagrama de despliegue.....	37
Diagrama de clases.....	38
Arquitectura de alto nivel	39
Instalación y Configuración.....	40
Pruebas y Calidad de Código.....	44
Conclusiones	46
Referencias Bibliográficas.....	47

Índice de ilustraciones

Ilustración 1: Variación de porcentaje de matriculados nivel nacional	22
Ilustración 2: Tablero Kanban del proyecto	30
Ilustración 3: Casos de uso arquitecturalmente relevante	31
Ilustración 4: Diagramas de secuencia para los padres y acudientes	36
Ilustración 5: Diagrama de secuencia colegios	37
Ilustración 6: Diagrama de despliegue	41
Ilustración 7: Diagrama de clases	42
Ilustración 8: Diagrama de arquitectura de alto nivel	43
Ilustración 9: Diagrama de configuración y tecnologías	46

Índice de tablas

Tabla 1: Buscar colegios por asistente	31
Tabla 2: Registrar datos de colegio	32
Tabla 3: Registrar datos de colegio	32
Tabla 4: Enviar correo con token de seguridad	33
Tabla 5: Validar token de seguridad	33
Tabla 6: Registrar datos de colegio	34
Tabla 7: Registrar auditoría	34
Tabla 8: Cargar datos iniciales de colegios	34
Tabla 9: Seguridad	38
Tabla 10: Usabilidad	38
Tabla 11: Desempeño	39

Resumen

El presente documento detalla el proyecto “Busco Colegio” para la Especialización En Ingeniería De Software de la Universidad Antonio Nariño que tiene por objetivo desarrollar un buscador de colegios de forma centralizada y detallada de los colegios a nivel Bogotá, donde se busca encontrar el colegio ideal para el estudiante apoyado en un buscador guiado.

El proyecto es desarrollado bajo metodología Scrumban que ha permitido la gestión de las tareas de forma remota y eficiente. Se implementó en entornos JavaScript usando Nodejs para el desarrollo del backend, para todo lo referente con el frontend se desarrolló con el framework Angular.

Abstract

This document details the project "Busca Colegio" for the Specialization in Software Engineering at Antonio Nariño University, the objective is to implement a search at the Bogotá level of the schools, we seek to find the ideal school for the student using a step search.

The project is developed with Scrumban methodology for ease of management works. It was implemented with Nodejs for backend and Angular for frontend.

Introducción

Históricamente la búsqueda de colegios se ha realizado mediante referencias personales de amigos y conocidos o por cultura general de los colegios tradicionales la situación restringía de una forma muy grande la búsqueda y obligaba a las personas a actuar de una forma muy conformista con las condiciones de las instituciones.

Con el paso de los años y la llegada de tecnologías de la información se evolucionó a la búsqueda por internet a través de los buscadores más reconocidos, pero estos no se especializan en estas búsquedas, muestran las instituciones más populares y restringen algunos resultados que pueden ser muy relevantes para usuario específico.

También algunas instituciones cuentan con páginas propias que proporcionan algunos datos y la información de contacto, generalmente estas instituciones son de carácter privado y sus funcionarios no cuentan con los conocimientos técnicos para modificar los datos, cuando estos cambian, por lo tanto, se desactualizan con frecuencia.

Las instituciones gubernamentales tienen compilados estos datos que se manejan de forma privada, pero existen otros que se manejan de forma pública en las que se encuentran datos básicos de los colegios y se encuentran distribuidas bajo diversos criterios como colegios privados, colegios públicos o por zonas. Estas bases de datos están disponibles de forma pública, pero no son accesibles para un usuario sin conocimientos técnicos.

Por lo anteriormente descrito se realizó una exploración de aplicaciones y páginas en internet que tuvieran relación con el tema las cuales son descritas a continuación.

Sumado a lo mencionado anteriormente, hoy en día la educación ha venido cambiando, buscando otros enfoques, explotando de otras formas las habilidades de los estudiantes y esto ha hecho que los padres de familia y/o acudientes generen unas preguntas

diferentes a la hora de consultar los colegios, dejando en un segundo plano otros puntos de interés como ubicación, costos.

La aplicación plantea una herramienta que les permita a los padres de familia y/o acudientes encontrar un colegio de la ciudad de Bogotá adecuado a las necesidades de los estudiantes y que cumpla con los criterios facilitando el filtrado y definición de aspectos como la metodología, infraestructura, religión, énfasis, actividades extracurriculares buscando de esta forma satisfacer las necesidades generadas.

Formulación y descripción del problema

Tradicionalmente la búsqueda de colegios para los estudiantes es una tarea tediosa, en ocasiones complicada y frustrante, ya que debemos destinar tiempo, dinero y paciencia para investigar y conocer los colegios que puedan ofrecer y suplir las necesidades de los padres/acudientes.

Sumado a lo anterior, las circunstancias en las que hoy se encuentra la sociedad luego de todo lo sucedido por el COVID 19, la vida en general ha sufrido grandes cambios y muchos de ellos han llegado para quedarse, la educación no ha sido ajena a estos cambios y la virtualidad ha jugado un papel importante en todo este proceso donde ya los padres de familia y/o acudientes han ido cambiando la manera tradicional de buscar los colegios para sus hijos y ahora estas búsquedas también se realizan de forma virtual, por esta razón se ve la necesidad de tener una herramienta que brinde una serie de funcionalidades a los usuarios que permita tanto ofrecer como buscar información sobre colegios donde pueden filtrar por diferentes características como metodologías de enseñanza, infraestructura, ubicación y a su vez registrar dicha información ofreciendo seguridad en todo el proceso de almacenamiento manejando una verificación de cuentas por medio de tokens para evitar suplantaciones y sabotajes en la información que se ingrese en la aplicación.

Adicionalmente en el mercado no se encuentra una aplicación que ofrezca con detalle y completitud información que cumpla con las expectativas que pueden llegar a tener los padres de familia y/o acudientes en cuanto al tema académico, de infraestructura, recursos y metodologías de enseñanza, lo mismo que para los colegios, donde ellos puedan ofertar toda su información partiendo de una aplicación segura y de fácil manejo.

La herramienta tiene como principio centralizar la información de todos los colegios en la ciudad Bogotá y ofrecer un sistema de búsqueda para los padres de familia y/o acudientes sin necesidad de realizar ningún desplazamiento y poderlo hacer desde cualquier dispositivo como Android, IOS o Windows, así facilitar esta tarea que como se mencionaba anteriormente es tediosa y compleja en ocasiones.

Objetivos

Objetivo general

Implementar una aplicación web que permita la búsqueda centralizada y actualizada de los colegios a nivel Bogotá.

Objetivos específicos

- Permitir que los padres de familia y/o acudientes puedan encontrar el colegio que desean por medio de una búsqueda guiada que irá mostrando las opciones paso a paso.
- Ofrecer a los colegios una aplicación web que les permita dar a conocer sus características y ventajas competitivas de tal forma que permita divulgar de mejor forma su información.
- Brindar seguridad en el registro de la información de los colegios mediante un mecanismo que permita verificar la autenticidad de la cuenta.
- Ofrecer una aplicación web que sea escalable en su infraestructura con el fin de garantizar siempre una alta disponibilidad.
- Brindar un indicador que muestre las características más populares entre los padres de familia y/o acudientes en sus búsquedas de colegios.
- Permitir a los padres y colegios dar a conocer su opinión o evaluar la aplicación.

Marco de Referencia

Históricamente la búsqueda de colegios se ha realizado mediante referencias personales de amigos y conocidos o por cultura general de los colegios tradicionales la situación restringía de una forma muy grande la búsqueda y obligaba a las personas a actuar de una forma muy conformista con las condiciones de las instituciones.

Con el paso de los años y la llegada de tecnologías de la información hemos evolucionado a la búsqueda por internet a través de los buscadores más reconocidos, pero estos no se especializan en estas búsquedas, muestran las instituciones más populares y restringen algunos resultados que pueden ser muy relevantes para usuario específico.

También algunas instituciones cuentan con páginas propias que proporcionan algunos datos y la información de contacto, generalmente estas instituciones son de carácter privado y sus funcionarios no cuentan con los conocimientos técnicos para modificar los datos, cuando estos cambian, por lo tanto, se desactualizan con frecuencia.

Las instituciones gubernamentales tienen compilados estos datos que se manejan de forma privada, pero existen otros que se manejan de forma pública en las que se encuentran datos básicos de los colegios y se encuentran distribuidas bajo diversos criterios como colegios privados, colegios públicos o por zonas. Estas bases de datos están disponibles de forma pública, pero no son accesibles para un usuario sin conocimientos técnicos.

Por lo anteriormente descrito se realizó una exploración de aplicaciones y páginas en internet que tuvieran relación con el tema las cuales son descritas a continuación.

Estado del Arte

Actualmente existen varios listados de los colegios a nivel Bogotá en las bases datos del gobierno que buscamos a través de la página www.datos.gov.co tras realizar una búsqueda detallada se encontró que estas están divididas bajo diversos criterios y que necesitan ser unificadas y a través de un algoritmo de comparación tomando los datos más actuales ya que hasta el momento no se ha encontrado una base de datos con fecha a este año.

Al realizar una búsqueda de aplicaciones que relacionen colegios se hallaron algunas similares a la propuesta descrita en este documento, sin embargo, sus nichos de mercado se encuentran fuera del país o están dirigidas a un público distinto a los padres y estudiantes de Bogotá, la que tiene más similitudes y enfocada en un concepto cercano a la aplicación web propuesta plantea una búsqueda de colegios bajo diversos criterios, pero está relaciona los clientes de una aplicación que también promueven en la misma página.

Educación virtual es un tema muy común en este momento, no sabemos a ciencia cierta cuál es el futuro de la educación, pero hay una visión general que nos dice que viene para quedarse y que no puede ser 100% automatizada debido a que solo las personas podemos apoyar el proceso cognitivo de cada estudiante, este concepto nos lleva a que los colegios no van a desaparecer, pero si deben cambiar por lo tanto un sistema que centralice el contacto con los colegios es un paso necesario para alcanzar el nuevo esquema de educación sin importar la modalidad que al final predominó.

Por otra parte, en la actualidad existe una fórmula que ha funcionado con diversas aplicaciones conocidas en las cuales se asocia cualquier persona que esté en capacidad de cumplir con un requerimiento necesario para un servicio específico, fomentando la sana

competencia y ofreciendo una mayor cantidad de opciones para elegir. Este tipo de políticas finalmente benefician al cliente; este es el caso de aplicaciones como Trivago y Rappy las cuales a pesar de ser catálogos de servicios reflejan un gran éxito con esta fórmula.

Después de verificar en multitud de las aplicaciones que brindan un servicio similar se encuentra que la mayoría tienen gestores de búsqueda muy sencillos que apenas permiten validar un par de criterios como por ejemplo calendario, género, privado o público. En este proyecto planteamos una búsqueda que le permita al colegio especificar con un nivel de detalle superior las características. Permitiendo a solicitud del usuario una búsqueda más detallada guiándose por una serie de preguntas como por ejemplo la infraestructura, las metodologías de enseñanza y las actividades extracurriculares llegando así a definir sus preferencias a la hora de elegir un colegio.

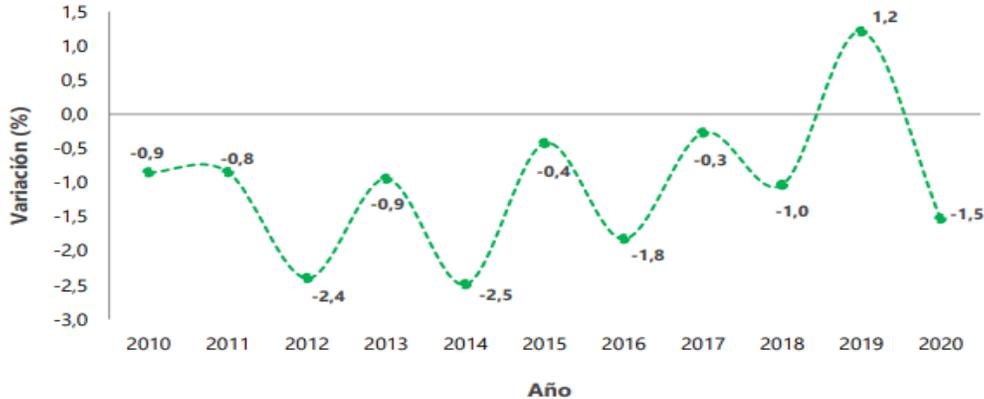
Luego de la investigación realizada y los resultados encontrados, se puede realizar un análisis del impacto que puede llegar a tener la aplicación web propuesta el cual se describe a continuación.

Impacto

La aplicación plantea una herramienta que les permita a los padres de familia de la ciudad de Bogotá encontrar un colegio adecuado a las necesidades de los estudiantes y que cumpla con los criterios de los padres facilitando el filtrado y definición de aspectos como la metodología, infraestructura, énfasis y actividades extracurriculares buscando con esto satisfacer las necesidades de toda la familia.

El reporte nacional del DANE muestra la variación de estudiantes matriculados durante el periodo de pandemia se evidencio una disminución del 2.7 % con respecto al año anterior, se espera que este porcentaje sea restaurado en los años venideros con la estabilización del virus.

Ilustración 1: Variación de porcentaje de matriculados nivel nacional



Nota. DANE, Educación formal (2021)

Se busca que con la aplicación los colegios puedan ofertar sus servicios de una manera más eficiente, que permita compararse de una manera sencilla con sus competidores, que conozcan de primera mano qué es lo que más buscan los usuarios permitiéndoles buscar

mejoras en sus servicios y así atraer más estudiantes mejorando el número de matrículas el cual se ha visto afectado como se mencionó anteriormente.

Componente de Innovación

Actualmente hay variedad de buscadores de colegios enfocados en diferentes criterios y para diversos públicos, sin embargo, la propuesta que se presenta en este documento es incluir instituciones oficiales y privadas con información más detallada la cual abarcara diferentes factores como educativos, sociales, económicos, culturales enfocados en las necesidades educativas del estudiante con el fin que los padres/acudientes puedan encontrar el colegio que más se adapte a sus condiciones y expectativas.

Los padres/acudientes van a ser guiados por todo el proceso por una serie de preguntas básicas hasta encontrar los colegios que ofrezcan todas los servicios y beneficios que están buscando y finalmente podrán ver un detalle enriquecido de cada uno de los resultados encontrados.

Se busca ayudar a las instituciones educativas a tener un medio por el cual puedan divulgar de mejor manera su información, sus beneficios, cualidades y/o características sobresalientes al público, por medio de una página web que ofrezca una gran experiencia de usuario al momento de ingresar la información, garantizando de manera segura el manejo y exposición de la información aplicando diferentes métodos como validaciones de cuenta para así evitar suplantaciones o alteraciones no deseadas a los datos de los colegios.

Se ofrecerá información de los colegios que, en los indicadores, han sido más encontrados o que características son las más buscadas por los usuarios, con esto los colegios podrán analizar en dónde pueden mejorar u ofrecer más servicios con el fin de tener más visibilidad.

Se ofrecerá un método de seguridad muy sencillo a los colegios para evitar suplantaciones o vulnerabilidades de su información, se tendrá un listado de correos electrónicos obtenidos de datos.gov.co (Datos abiertos Colombia 2019-2021) con el fin de tener un listado oficial. Por medio de tokens de verificación los cuales serán enviados a los correos oficiales de los colegios ellos podrán acceder a la aplicación web para registrar, modificar o eliminar su información sin tener que generar cuentas de acceso o abrir brechas de seguridad.

La aplicación será construida como una Aplicación web progresiva PWA, lo cual se detalla más adelante en el marco teórico, pero en este apartado se resalta la ventaja competitiva de que la aplicación sea multiplataforma teniendo acceso a un público más amplio y permitiendo la descarga y almacenamiento de resultados en el dispositivo.

Marco teórico

Se brindará una aplicación PWA que entrega diferentes ventajas sobre las páginas web tradicionales, como por ejemplo la apariencia nativa ya que la manera de navegar e interactuar con el usuario es bastante similar a la de una aplicación nativa. Otra ventaja que ofrece es la velocidad de respuesta tanto en carga como en navegación, ya que se apoyan en el almacenamiento en caché.

Una característica importante es la navegación sin conexión a internet ya que las aplicaciones PWA cuentan con una característica que permite almacenar la información en caché del lado del cliente y así mostrar información del sitio previamente cargado. Los usuarios tendrán la posibilidad de generar un acceso rápido por medio de un icono que se ubicará en el menú o en la pantalla de inicio, así se obtendrá una mayor visibilidad dando una mayor relevancia a la página web para los usuarios y todo esto logrado sin necesidad de pasar por una tienda de aplicaciones ni realizar procesos de instalación, tareas o acciones que pueden resultar en ocasiones tediosas o que ocupen más espacio del deseado o necesario.

Tecnologías

En cuanto a las tecnologías que van a ser usadas en el proyecto podemos mencionar lo siguiente. Nodejs, va a ser nuestro lenguaje de programación para realizar el backend, una de las características que ofrece esta tecnología es bastante eficiente en su respuesta, permite tener una alta escalabilidad, puede extender sus funcionalidades añadiendo código de NPM, lo que nos ofrece una gran cantidad de librerías en caso de ser necesarias.

Por otro lado, va a ser utilizado el framework Angular para todo lo referente con el frontend, que ofrece ventajas como el lenguaje TypeScript, maneja todo por componentes

lo cual elimina el patrón MVC que es muy costoso y pasa a una especie de rompecabezas donde se van a incluir los componentes desarrollados.

Como motor de DB va a ser implementado Mongo DB, el cual es un complemento perfecto para javascript es una herramienta que consume pocos recursos; a su vez ofrece otras ventajas como escalabilidad, alta disponibilidad y gran flexibilidad.

Se debe mencionar Docker Swarm como tecnología para disponibilizar la aplicación, lo que va a permitir crecer o decrecer la cantidad de nodos o réplicas con las que se cuenta para disponer la aplicación, así ofrecer de manera fácil una alta disponibilidad en caso de ser requerida y disminuir costos cuando esta no necesite recursos.

Finalmente es muy importante mencionar que todo el uso de estas tecnologías va a estar apoyado por buenas prácticas de programación, usando como base todo lo relacionado con el código limpio, siguiendo todas las recomendaciones como nombramientos de funciones, variables, tener un código lo suficientemente claro para evitar comentarios incensarios, siguiendo las indicaciones SOLID.

Buscador Asistido

Esquema de búsqueda amigable que asista al usuario y lo guíe hasta encontrar los resultados, se plantea un buscador que ofrezca al usuario una experiencia agradable al momento de interactuar con la aplicación, con el objetivo de salir de los buscadores rígidos o formularios complejos y tediosos de diligenciar.

Seguridad

La seguridad será implementada en los servicios para el caso de actualización de datos por parte del colegio, debido a que esta información es delicada y propensa a daños de terceros, se plantea un esquema de validación con verificación del correo oficial del colegio,

el cual será parametrizado durante la carga inicial de las bases de datos del gobierno. A este correo le será enviado un token cifrado en AES-256 y tendrá que ser usado para la autorización de actualización de datos.

Protección de datos

Para la aplicación no es necesario almacenar datos de los usuarios que consultan debido a que estos no tendrán contacto directo con alguno de los colegios a través de la aplicación. Se planea almacenar los metadatos de la consulta esto con el fin de generar estadísticas de uso y mostrar características más populares y búsquedas de los colegios.

Los datos de los colegios serán proporcionados voluntariamente por cada uno de ellos buscando mostrar sus características de una forma adecuada y en busca de captar más atención de las familias. Los datos básicos del colegio son proporcionados por las bases de datos públicas de los colegios y para este caso no se encuentra la restricción legislativa.

Metodología

En el desarrollo del proyecto la metodología que se aplica es Scrumban debido a que reúne la capacidad de mejora, simplicidad y facilidad para adoptar de Kanban y la naturaleza perceptiva, adaptabilidad de Scrum. Permitiendo un enfoque dinámico y pragmático en la mejora continua y la priorización de tareas con sus plazos de entrega.

La configuración de capacidad para la realización de tareas en el proyecto es de 2 tareas por estado del tablero para los dos miembros del equipo, todo esto manejado en iteraciones de 1 semana.

Estados:

- To Do: Actividades pendientes para agregar a un Sprint.
- Pending: Actividades estimadas para el Sprint pero que aún no se han iniciado.
- In Progress: Actividad iniciada y en proceso.
- Review / QA: Actividad en proceso de pruebas y certificación.
- Done: Actividad terminada.

Niveles de prioridad:

- Sprint: Actividades estimadas e incluidas para entrega en el Sprint.
- Solicitud de cambio: Actividad para la cual se solicitó un cambio pero que está actualmente en producción.

Roles:

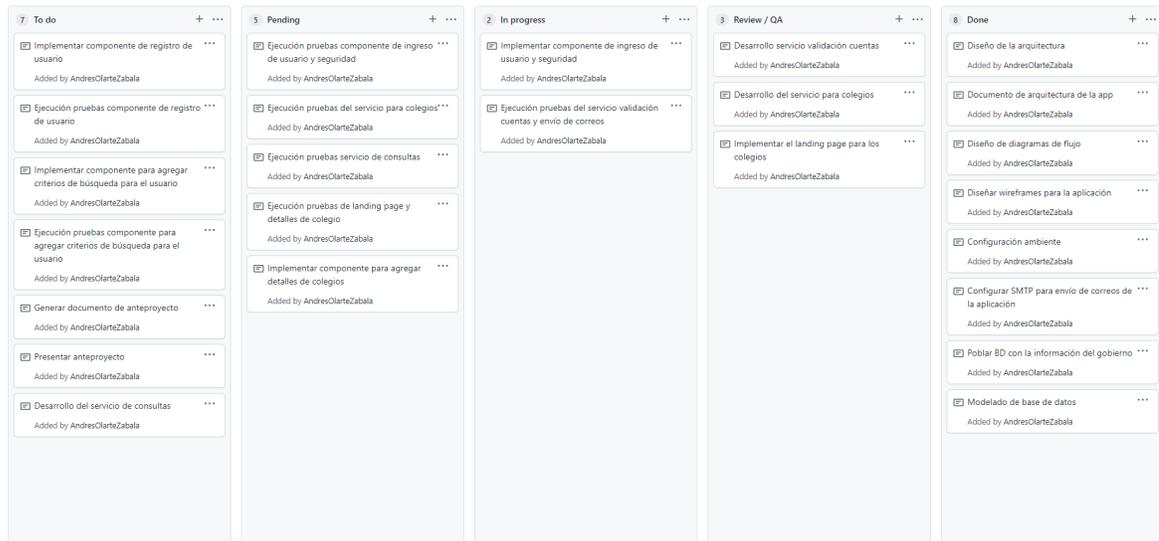
- Dueño del producto

- Equipo

Reuniones

- Planeamiento: Reunión realizada en el inicio del Sprint.
- Diaria: Reuniones diarias de sincronización y solución de dudas.
- Retrospectiva: Reunión realizada al final del Sprint para verificar el trabajo realizado y oportunidades de mejora.

Ilustración 2: Tablero Kanban del proyecto



Nota. Creación propia (2021) - Imagen del tablero manejado por el equipo.

Proceso de software

Todo se inició con el levantamiento y análisis de los requerimientos, ya con el problema definido y acotado de manera clara, se empezó a relevar todo el detalle de las funcionalidades, los actores que iban a estar involucrados en todos los flujos. Seguido de este relevamiento se crearon las historias de usuarios, dando detalle de todo lo requerido, definiendo los criterios de aceptación para cada una de ellas.

La siguiente etapa del proceso consistió en todo el diseño de la arquitectura y el desarrollo de los diagramas, si bien la arquitectura tuvo algunas iteraciones y durante ellas sufrió algunos cambios, desde el inicio se concretaron las tecnologías y en su mayoría la esencia de lo que sería el proyecto. Para el desarrollo del backend fue utilizado NodeJs que es un entorno de ejecución real que utiliza javascript, es bastante liviano y rápido que se acomoda a las necesidades del proyecto. En lo que respecta al frontend fue utilizado Angular v12, un framework que permite modularizar las funcionalidades, separando en componentes cada una de las implementaciones y permite crear aplicaciones PWA, que es uno de los puntos mencionados como componente de innovación.

Para la persistencia de datos se utilizo MongoDB como motor de bases de datos, que permite manejar documentos lo cual fue de bastante utilidad al momento de almacenar la información de los colegios, pues se requieren bastantes datos por la cantidad de información almacenada dado el contexto de las preguntas y el enfoque de las búsquedas, al no tener relaciones marcadas con otros datos fue la elección ideal para el proyecto.

Posterior a las etapas de definición, análisis y generación de documentación con los diagramas, se definió la metodología de trabajo eligiendo a SCRUMBAN, creando los sprint y las tareas que se debían encarar en cada uno de ellos.

Dando inicio con los sprint se inició el desarrollo de la aplicación, durante todo el proceso se implementaron pruebas cruzadas mitigando en parte posibles errores a la hora de hacer las pruebas de caja negra.

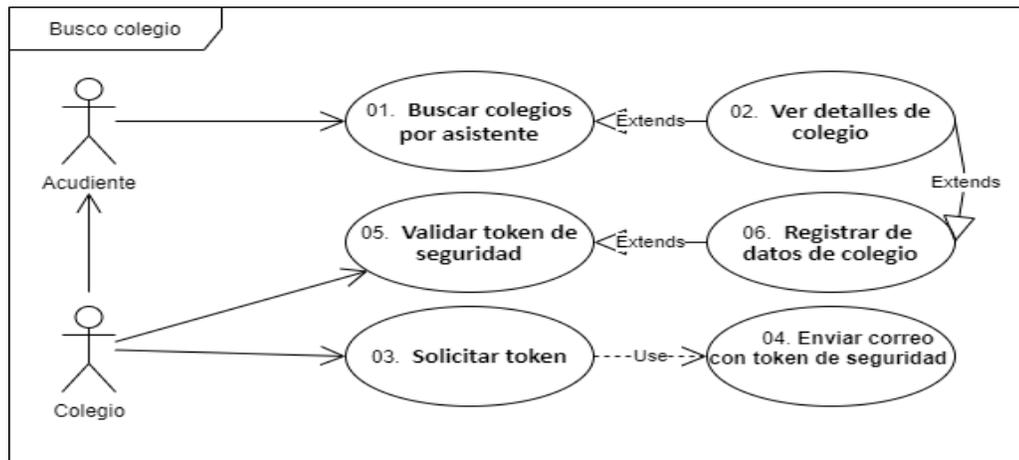
Durante todo el proceso, existieron circunstancias que generaron algunas dificultades para llevar a cabo el proyecto, quizás el factor mas determinante fue el tiempo, ya que por las diversas actividades fueras del entorno académico se perdía la fuerza de trabajo para el proyecto, este punto se aminoró con sobre esfuerzos los cuales rindieron frutos y el producto pudo ser terminado, cumpliendo con los objetivos pactados. Otra dificultad que se presentó durante todo el proceso de software fue la parte de diseño y experiencia de usuario ya que la formación de las personas involucradas en el desarrollo no tenía esa formación, la forma que se atacó ese problema y de alguna manera se solventó, fue validando con personas de ese rubro quienes asesoraron y dieron algunos consejos para tener un buen resultado.

Todo lo referente a las buenas practicas de desarrollo, fueron tenidas en cuenta durante todo el proceso lo que dejó una gran satisfacción al final, pues durante todo el desarrollo los integrantes del equipo fueron abordando los requerimientos y el código tuvo siempre un mismo formato, como si hubiese sido escrito por una sola persona.

Requerimientos funcionales

Los requerimientos funcionales describen todos los comportamientos y funcionalidades de la aplicación, a continuación se explican los casos de uso y los requerimientos en historias de usuario que contempla la aplicación web Busca Colegio.

Ilustración 3: Casos de uso arquitecturalmente relevante



Nota. Creación propia (2021) - En la figura se puede observar los casos de uso de la aplicación.

Tabla 1: Buscar colegios por asistente

Código	HU001		
Requerimiento	Buscar colegios por asistente		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	Como un Padre/Acudiente necesito una funcionalidad que me permita definir la infraestructura tecnológica o física para el colegio que se acerca a mis criterios, el método de enseñanza que más cercano a mis criterios, las actividades extracurriculares para el estudiante, en la búsqueda de los colegios así podré ver solamente los colegios que se ajustan a los criterios que seleccione de la lista.		

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Tabla 2: Registrar datos de colegio

Código	HU002		
Requerimiento	Ver detalles de colegio		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	<p>Como un Padre/Acudiente necesito una funcionalidad que me permita ver los datos detallados y formateados del colegio que seleccione así podré:</p> <ul style="list-style-type: none"> ● Definir correctamente el colegio. ● Validar si aplica para mis criterios. ● Ver los datos de contacto. 		

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Tabla 3: Registrar datos de colegio

Código	HU003		
Requerimiento	Solicitar token		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	<p>Como un Colegio necesito una funcionalidad que me permita verificar mi identidad en el sistema mediante el correo certificado que se encuentre configurado al momento, así podré ingresar al sistema para actualizar mi información.</p>		

Nota. Creación propia (2021) – Historia de usuario principal del sistema

Tabla 4: Enviar correo con token de seguridad

Código	HU004		
Requerimiento	Enviar correo con token de seguridad		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	Como Colegio necesito que al momento de solicitar un token se envíe un correo electrónico que se encuentre configurado para mí como colegio y así podré verificar que el correo me pertenece y validar mi identidad.		

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Tabla 5: Validar token de seguridad

Código	HU005		
Requerimiento	Validar token de seguridad		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	<p>Como un Colegio necesito un mecanismo que me permita verificar la identidad de los colegios mediante un token que le es entregado al correo configurado inicialmente así podré</p> <ul style="list-style-type: none"> ● Modificar los datos de ser necesario ● Verificar la autenticidad de los datos ● Ver todos los datos configurados 		

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Tabla 6: Registrar datos de colegio

Código	HU006		
Requerimiento	Registrar datos de colegio		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	Como un Colegio necesito una funcionalidad que me permita ver y editar mis datos así podré actualizar mi información cada vez que sea necesario		

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Tabla 7: Registrar auditoría

Código	HU007		
Requerimiento	Registrar auditoría		
Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción	
Fuente	Equipo de desarrollo		
Prioridad	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
Descripción	Como un Colegio necesito un mecanismo de registro de log de cambios que se realicen a mis datos así podré ver un historial de cambios realizados y si es necesario corregirlos		

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Tabla 8: Cargar datos iniciales de colegios

Código	HU008		
Requerimiento	Cargar datos iniciales de colegios		

Tipo	<input checked="" type="checkbox"/> Historia. usuario	<input type="checkbox"/> Restricción
Fuente	Equipo de desarrollo	
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	Como un Colegio necesito un mecanismo de carga inicial de los datos básicos relativos a mi ubicación e información de contacto así podré verificar la coincidencia de los datos básicos de mi colegio para que permanezcan actualizados.	

Nota. Creación propia (2021) – Historia de usuario principal del sistema.

Requerimientos no funcionales

En los requerimientos no funcionales se detallarán las características generales, atributos de calidad de la aplicación, donde se ve la manera en que los usuarios podrán editar la información, en que dispositivos podrá ser usada la aplicación y el desempeño que debe tener la aplicación de cara al usuario final.

Tabla 9: Seguridad

Atributo de Calidad	Seguridad	
Confiabilidad	Descripción	Prioridad
Autorización	La aplicación solo permitirá registrar información o editarla una vez se haya verificado la propiedad de la cuenta ingresando el token que se enviará a la cuenta oficial.	Alta

Nota. Creación propia (2021)

Tabla 10: Usabilidad

Atributo de Calidad	Usabilidad	
	Descripción	Prioridad
	La aplicación debe ser multiplataforma, permitiendo que los distintos usuarios puedan darle uso desde cualquier dispositivo, debe ser intuitiva al momento de registrar la información y con interfaces sencillas.	Media

Nota. Creación propia (2021)

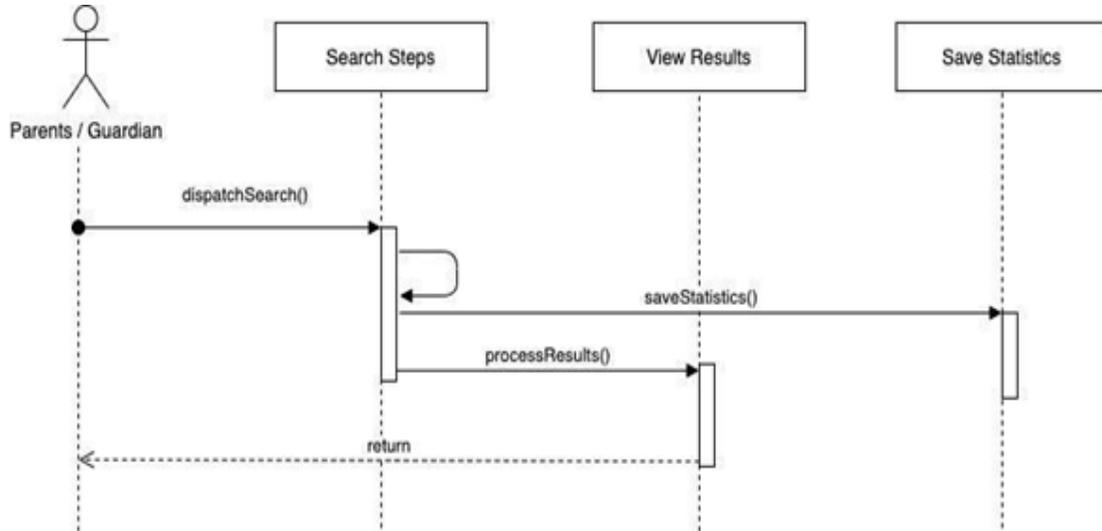
Tabla 11: Desempeño

Atributo de Calidad	Desempeño	
Escalabilidad	Descripción	Prioridad
	La aplicación debe poder escalar y desescalar para atender una mayor o menor demanda según la necesidad, así siempre estará disponible.	Alta

Nota. Creación propia (2021)

Diagrama de secuencia

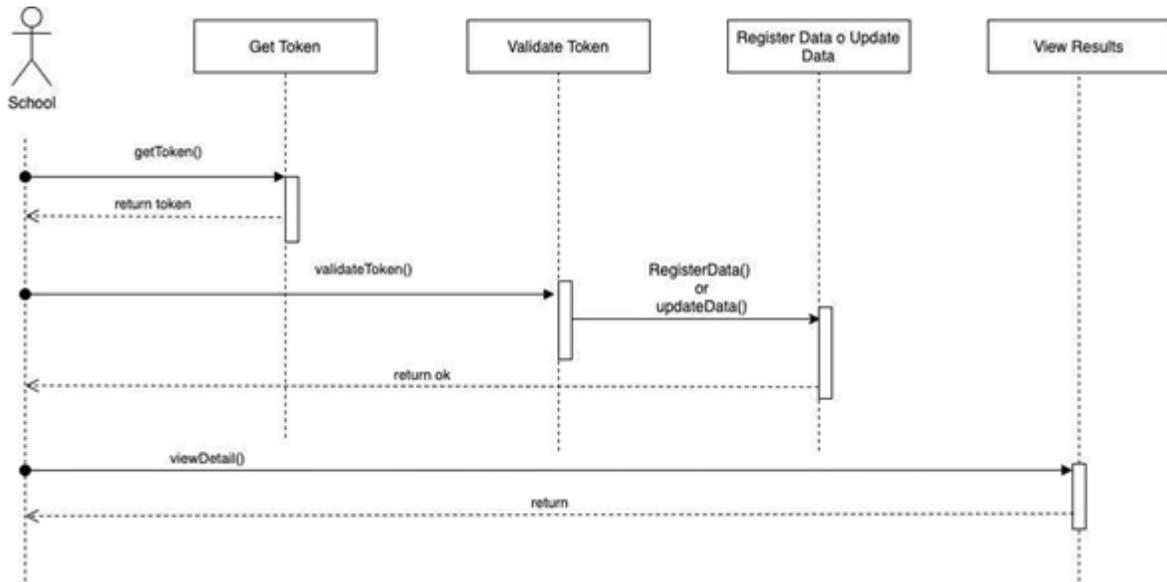
Ilustración 4: Diagramas de secuencia para los padres y acudientes



Nota. Creación propia (2021) - En la figura se puede observar los flujos principales de la aplicación para los padres de familia y/o acudientes, donde se tiene un buscador por pasos, un listado de resultados luego de realizar las búsquedas y un log que va a ir registrando todos los movimientos de los usuarios para ofrecer unas estadísticas que indiquen que es los que mas buscan los padres de familia y/o acudientes.

1. El Padre/Acudiente inicia la búsqueda de colegio
2. El Padre/Acudiente realiza paso a paso el llenado del formulario de búsqueda
3. El sistema en la selección de cada uno de estos criterios los metadatos son almacenados para generar futuras estadísticas.
4. El sistema después de procesar los criterios de búsqueda genera los resultados
5. El sistema retorna los resultados al Padre/Acudiente

Ilustración 5: Diagrama de secuencia colegios



Nota. Creación propia (2021) - En la figura se observa el diagrama de flujo para el flujo de carga de datos del colegio, donde se valida el proceso que debe seguir cada colegio, obteniendo un token que va a ser enviado por correo electrónico para poder acceder al sistema para editar la información y finalmente ver la información que va a estar disponible para los usuarios.

1. El colegio solicita un token para la edición de los datos.
2. El sistema retorna un token para realizar el proceso.
3. El colegio envía el token para validarlo.
4. Si el token es válido permitirá registrar los datos de actualización.
5. El sistema actualiza los datos y retorna el resultado.
6. El colegio podrá verificar los datos actualizados.

Diseño y arquitectura

Se plantea la arquitectura microservicios implementados en lenguaje Javascript sobre el servidor NodeJS con base de datos MongoDB permitiendo una implementación sencilla y ágil que da libertad al equipo de desarrollo de concentrarse en la usabilidad de la aplicación con alto rendimiento, apoyado con un token encriptado AES-256 para la edición de los datos.

Se definen cinco grupos funcionales para los microservicios que son:

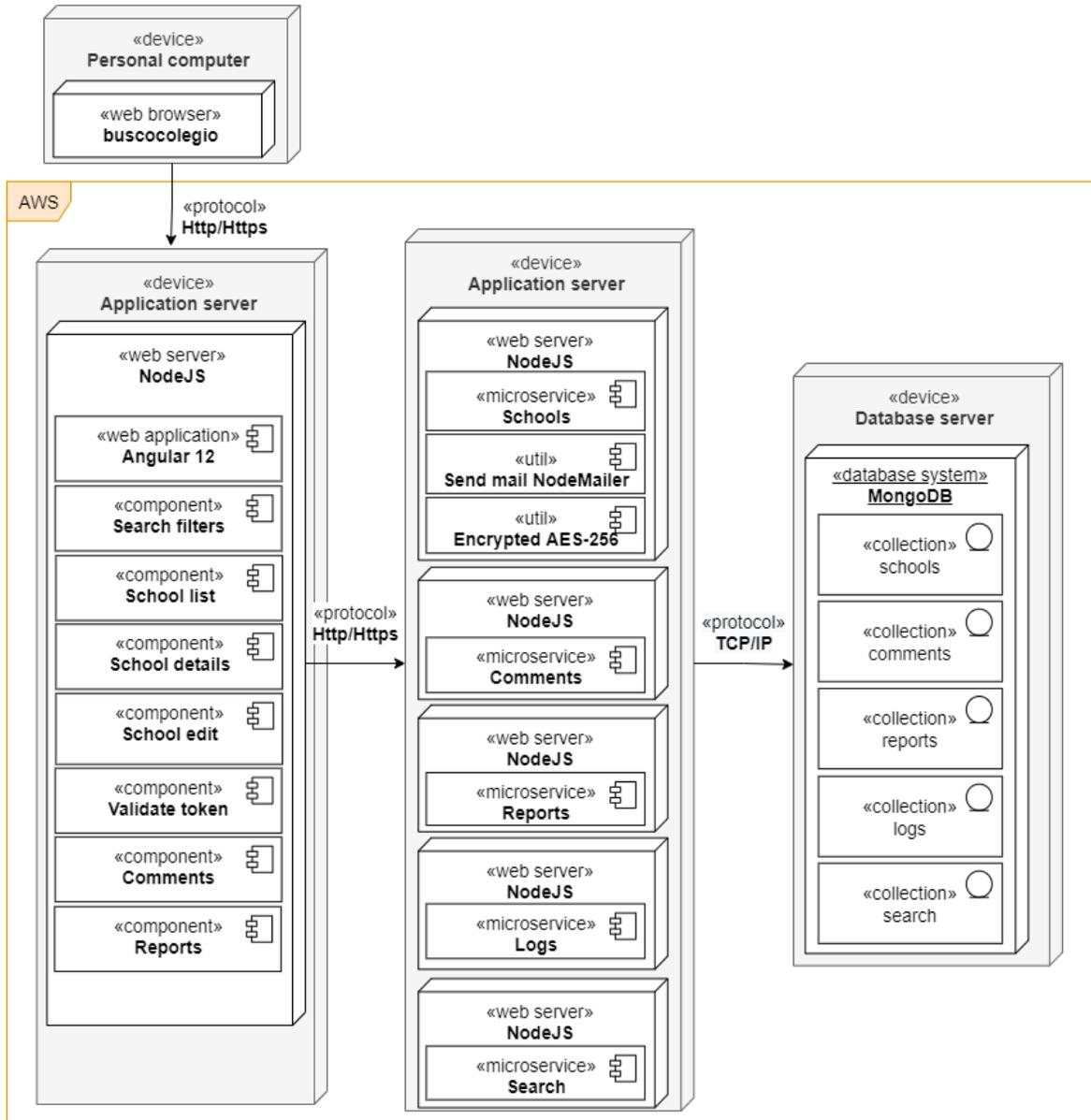
- **Schools:** Microservicio que administra los datos de los colegios.
- **Search:** Microservicio que administra los metadatos de las búsquedas para almacenarlos y su posterior análisis.
- **Reports:** Microservicio que administra los datos reportados a los colegios con relación a las búsquedas, los analiza y genera los índices de búsquedas.
- **Logs:** Microservicio que administra los cambios realizados a los colegios a modo de historial.
- **Comments:** Microservicio que administra las opiniones de los usuarios y calificaciones a la aplicación.

Cada uno de los microservicios cuenta con su respaldo en una de las colecciones de base de datos respectivamente, esta base de datos será implementada en MongoDB.

El front de la aplicación está implementado en Javascript usando el framework Angular en su versión 12, basados en el principio de responsabilidad única, cada funcionalidad está modularizada permitiendo tener un alto desacoplamiento en toda la implementación, también se usó el patrón Redux para todo el manejo de conexión con los servicios de backend y manipulación de la información que se le muestra al usuario.

Diagrama de despliegue

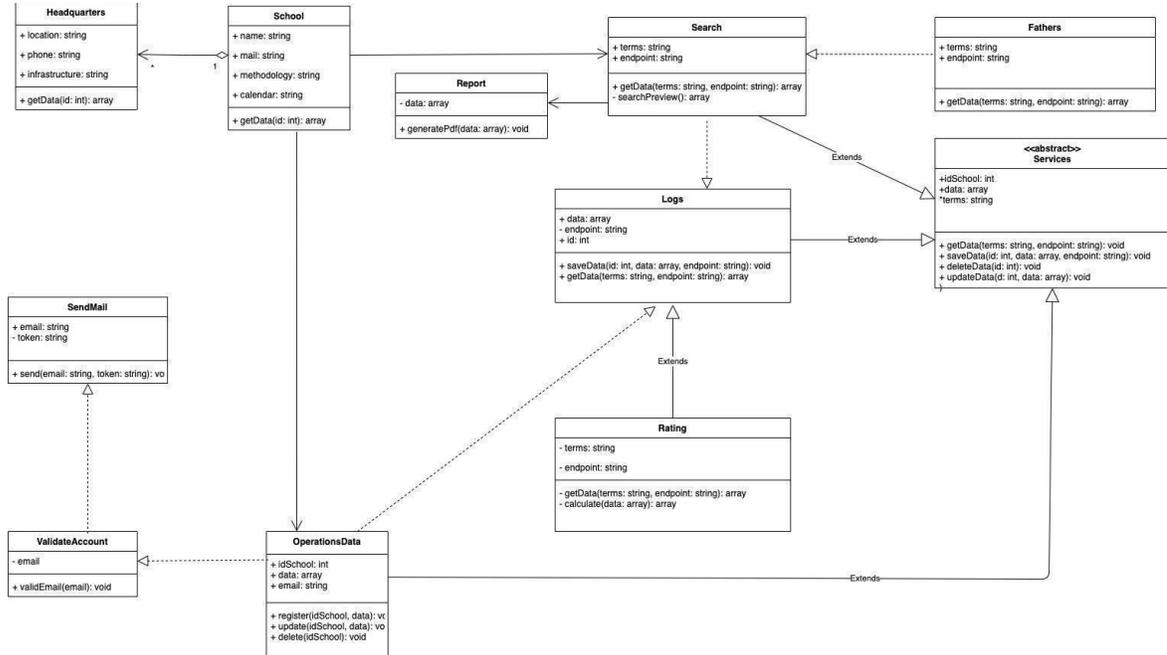
Ilustración 6: Diagrama de despliegue



Nota. Creación propia (2021) - En la figura se describe la estructura de componentes de la aplicación, donde se pueden ver los microservicios que tendrá la aplicación los servicios de encriptado y envío de correos .

Diagrama de clases

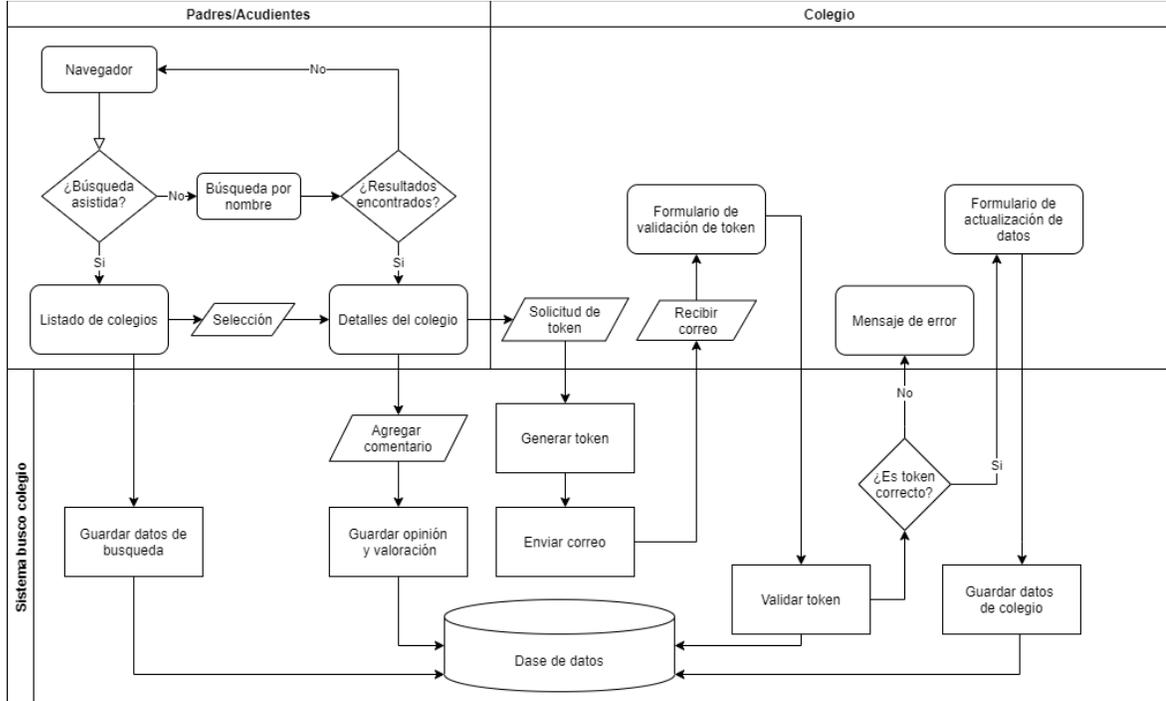
Ilustración 7: Diagrama de clases



Nota. Creación propia (2021) - En la figura se describe la estructura del diseño de clases, se puede ver con las características del sistema, las interacciones que van existir entre cada una de ellas.

Arquitectura de alto nivel

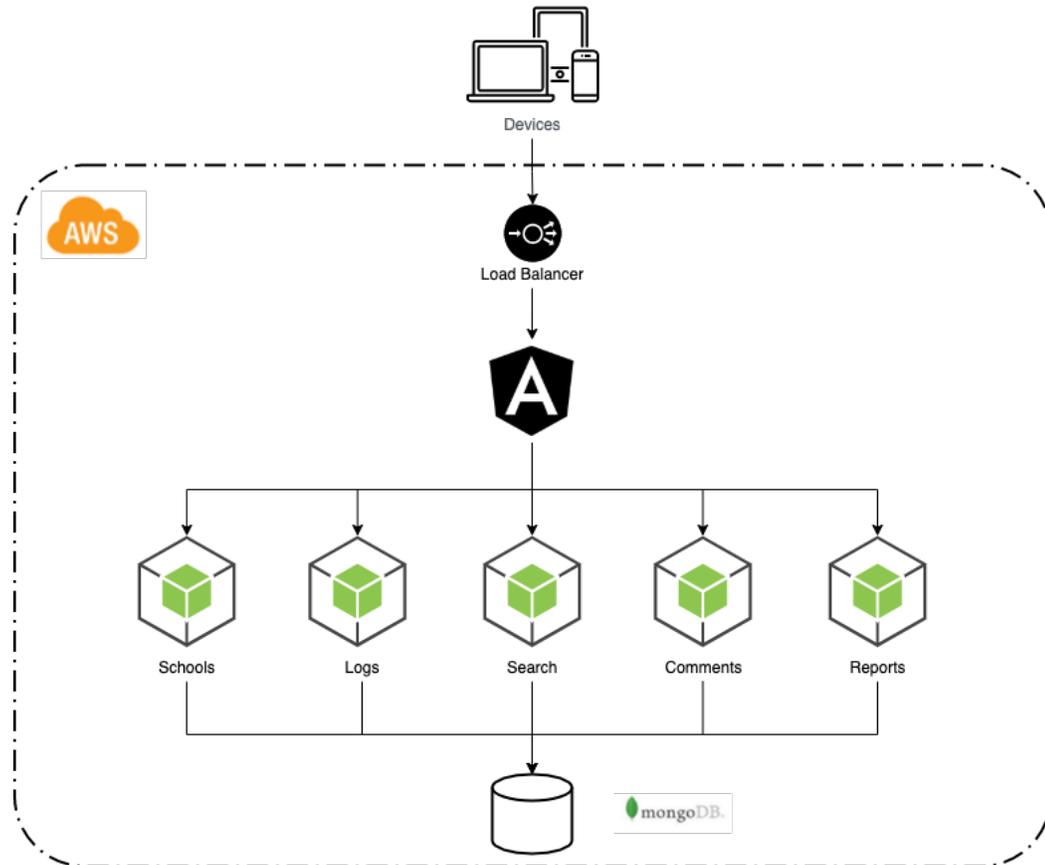
Ilustración 8: Diagrama de arquitectura de alto nivel



Nota. Creación propia (2021) - En este diagrama se puede observar cómo interactúa el sistema con los actores en todo el proceso, donde se destaca una única base de datos para persistir la información de todos los flujos e interacciones.

Instalación y Configuración

Ilustración 9: Diagrama de configuración y tecnologías



Nota. Creación propia (2021) - Diagrama de tecnologías y definición de objetos de servicios y aplicación.

Basamos la arquitectura de la aplicación en microservicios y conectados un cluster de base de datos, esto nos permite tener únicas responsabilidades para cada servicio como se puede evidenciar en la gráfica, todo va a ser desplegado en instancias de AWS, apoyado en Docker y Github para todo lo relacionado con el versionamiento de código y los release.

Se configuró un balanceador de carga de aplicaciones para el frontend, quien es el que recibirá todo el tráfico e ira llamando a cada uno de los microservicios según la necesidad, a su vez los microservicios irán usando la base de datos cuando se requiera. Tanto el frontend como los microservicios y la base de datos estarán disponibles por medio de contenedores Docker los cuales son desplegados por medio de docker-compose, de esta forma es requerido tener un archivo Dockerfile y un archivo docker-compose.yml que se encargarán de configurar lo necesario.

```
FROM node:18-alpine

RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app

WORKDIR /home/node/app

COPY package*.json ./

USER node

RUN npm install

COPY --chown=node:node . .

EXPOSE 3000

CMD [ "node", "app.js" ]
```

Nota. Creación propia (2021) - Archivo Dockerfile para aplicación NodeJs .

```
version: '3'
services:
  nodejs:
    build:
      context: .
      dockerfile: Dockerfile
    image: nodejs_search
    container_name: nodejs_search
    restart: unless-stopped
    env_file: .env
    environment:
      - MONGO_USERNAME=$MONGO_USERNAME
      - MONGO_PASSWORD=$MONGO_PASSWORD
      - MONGO_HOSTNAME=db
      - MONGO_PORT=$MONGO_PORT
      - MONGO_DB=$MONGO_DB
    ports:
      - "3000:3000"
    volumes:
      - ./home/node/app
      - node_modules:/home/node/app/node_modules
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

volumes:
  node_modules:
```

Nota. Creación propia (2021) - Archivo docker-compose.yml para aplicación NodeJs.

```

version: '3'
services:
  db:
    image: mongo:4.1.8-xenial
    container_name: db
    restart: unless-stopped
    env_file: .env
    environment:
      - MONGO_INITDB_ROOT_USERNAME=$MONGO_USERNAME
      - MONGO_INITDB_ROOT_PASSWORD=$MONGO_PASSWORD
    ports:
      - "27019:27017"
    volumes:
      - dbdata:/data/db
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

volumes:
  dbdata:

```

Nota. Creación propia (2021) - Archivo docker-compose.yml para la base de datos MongoDB.

Con los archivos configurados para cada aplicación desde el servidor se ejecuto un archivo bash que tiene como responsabilidad levantar todos los docker y así quedarán desplegados todos los componentes descritos.

```

#
echo "Init docker images!!"

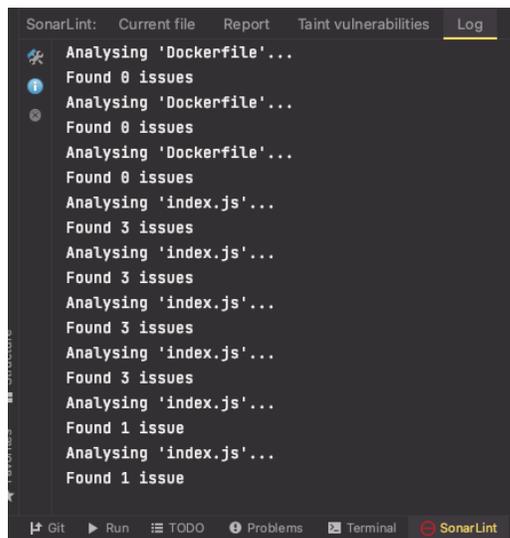
cd uan-mongo && docker-compose up -d
cd ../uan-logs && && docker-compose up -d
cd ../uan-comments && && docker-compose up -d
cd ../uan-search && && docker-compose up -d
cd ../uan-school && docker-compose up -d
cd ../uan-report && docker-compose up -d
cd ../uan-front && docker-compose up -d

```

Nota. Creación propia (2021) - Archivo bash para desplegar todos los contenedores Docker.

Pruebas y Calidad de Código

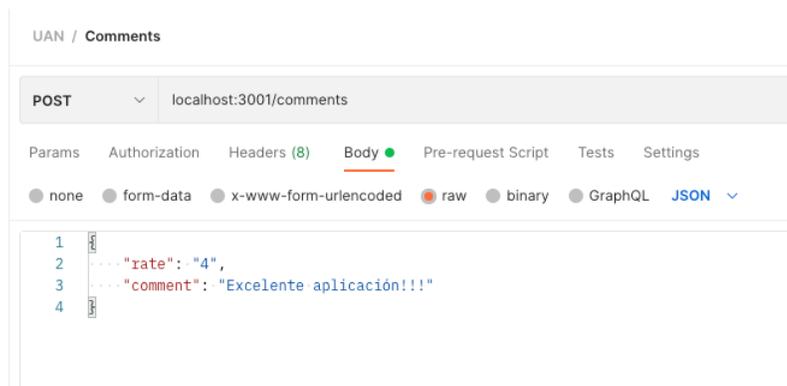
En lo que refiere a las pruebas y calidad de código, durante el proyecto se usaron varias herramientas y metodologías que ayudaron a llevar a buen puerto este apartado del proyecto. Como primera medida se usó SonarLint como apoyo para lo que implica el código limpio, evitar repetición de código nombramientos erróneos por ejemplo.



```
SonarLint: Current file Report Taint vulnerabilities Log
Analyzing 'Dockerfile'...
Found 0 issues
Analyzing 'Dockerfile'...
Found 0 issues
Analyzing 'Dockerfile'...
Found 0 issues
Analyzing 'index.js'...
Found 3 issues
Analyzing 'index.js'...
Found 1 issue
Analyzing 'index.js'...
Found 1 issue
```

Nota. Creación propia (2021) - Log que iba generando el plugin de SonarLint.

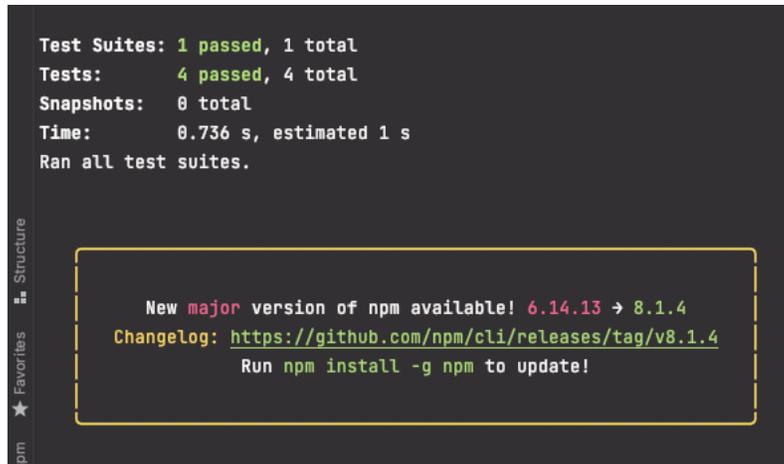
Para las pruebas de caja negra se utilizó a Postman para validar las funcionalidades de los servicios.



```
UAN / Comments
POST localhost:3001/comments
Body
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1
2   ... "rate": "4",
3   ... "comment": "Excelente aplicación!!!"
4
```

Nota. Creación propia (2021) - Ejemplo del uso de la herramienta Postman

Una metodología que fue usada y generó grandes resultados fue realizar pruebas cruzadas, al no contar con equipo un equipo de QA se realizaron las pruebas intercambiando los desarrolladores para que estas no estuvieran segadas o viciadas por cada uno de los involucrados, permitiendo tener un mejor panorama al momento de probar, todo esto apoyado y soportado por pruebas unitarias.



```
Test Suites: 1 passed, 1 total
Tests:      4 passed, 4 total
Snapshots:  0 total
Time:       0.736 s, estimated 1 s
Ran all test suites.
```

npm Structure
★ Favorites

```
New major version of npm available! 6.14.13 → 8.1.4
Changelog: https://github.com/npm/cli/releases/tag/v8.1.4
Run npm install -g npm to update!
```

Nota. Creación propia (2021) - Ejemplo de uno de los resultados de las pruebas ejecutadas

Conclusiones

Todo el trabajo realizado y toda la investigación hecha durante estos últimos meses deja varias conclusiones importantes, como la falta de herramientas que se enfoquen en las necesidades que pueden tener los padres de familia y/o acudientes a la hora de encontrar un colegio para sus hijos, que todo lo sucedido con la situación de la pandemia demostró que no estábamos preparados para darle manejo a diferentes necesidades de nuestro día a día y con el tema de la educación quizás se fue donde más cosas se tuvieron que modificar y se descubren necesidades de herramientas como la descrita en este documento.

Con el desarrollo de esta aplicación se quiere explotar de una forma diferente la búsqueda de colegios, enfocado en las necesidades y habilidades que tienen los niños y jóvenes hoy en día, ofreciendo a los padres de familia y/o acudientes una herramienta que les ayude a encontrar un colegio que permita explorar y explotar esos talentos, que se acomoden mejor al estilo y creencias de la familia; se busca ayudar a ver los colegios desde otras perspectivas ya que la educación ha evolucionado y la manera tradicional como muchos estudiamos ya no es bien vista y con esta herramienta se espera ayudar en gran parte a ver esos cambios.

Referencias Bibliográficas

Datos abiertos Colombia (2019-2021). Bogotá: www.datos.gov.co.
<https://datos.gov.co/browse?sortBy=newest&utf8=%E2%9C%93>

DANE. (2020-2021). Boletín Técnico. Bogotá: www.dane.gov.co
https://www.dane.gov.co/files/investigaciones/boletines/educacion/bol_EDUC_20.pdf

Ministerio de Educación. Impactos del COVID- 19 (2020-2021). Bogotá:
www.mineducacion.gov.co.
https://www.dane.gov.co/files/investigaciones/boletines/educacion/bol_EDUC_19.pdf

Ministerio del interior (1991). Bogotá: <https://www.mininterior.gov.co>
https://www.mininterior.gov.co/sites/default/files/constitucion-politica-colombia_2.pdf

Reportes públicos de la Secretaría de Educación (2021). Bogotá:
www.educacionbogota.edu.co
<https://dueb.educacionbogota.edu.co/Dueb/varIrGenerarReportesPublicos.sed>

Datos abiertos de Bogotá - Matriculas (2017- 2021). <https://datosabiertos.bogota.gov.co>
https://www.educacionbogota.edu.co/portal_institucional/conoce-nuestras-cifras

Informe de Análisis Estadístico LEE (2020). Laboratorio de Economía de la Educación
<https://secureservercdn.net/198.71.233.138/evb.eef.myftpupload.com/wp-content/uploads/2020/08/INFORME-12-LEE-PUJ-TRANSPORTE-PU%CC%81BLICO-Y-ESTUDIANTES-5-Y-9.pdf>