



SANTOTO
— Seccional Tunja —

www.ustatunja.edu.co

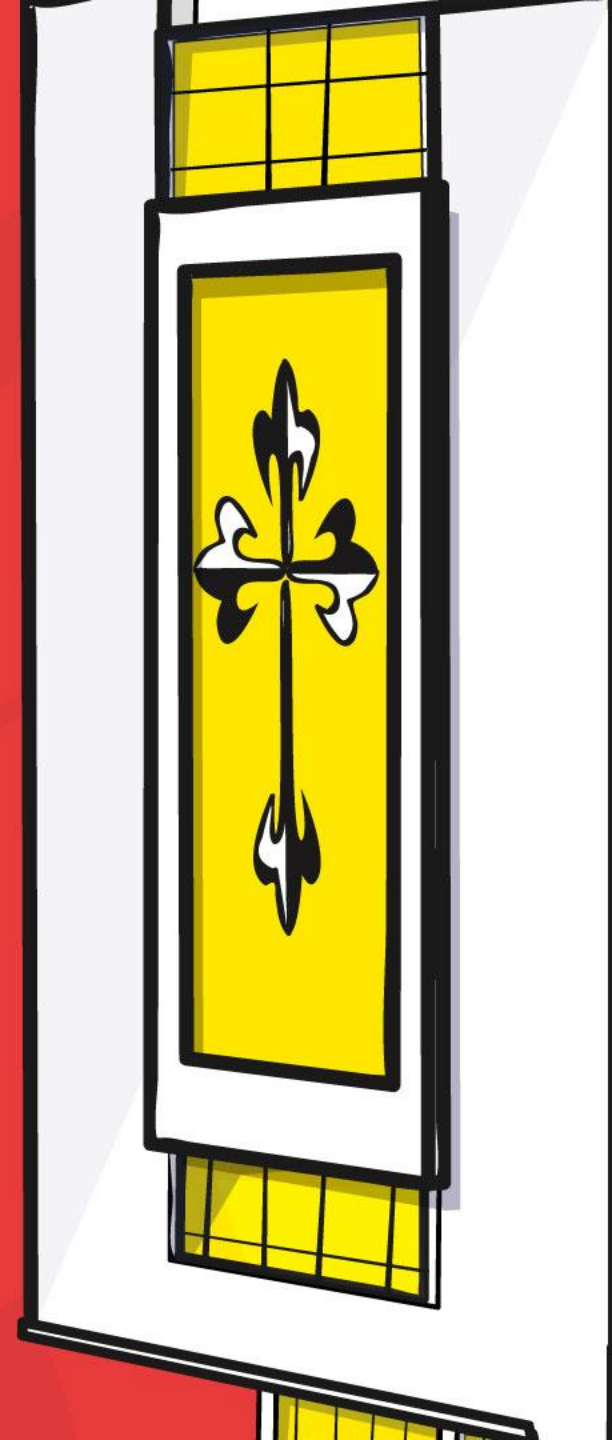


MÓDULO DE ADMINISTRACIÓN DE TRABAJOS DE GRADO MAESTRÍA EN CALIDAD Y GESTIÓN INTEGRAL

Ponentes: Juan Diego Gomez Gomez
Jose Alejandro Garcia Preciado
Director: Carlos Andres Guerrero Alarcon



@SANTOTOMASTUNJA



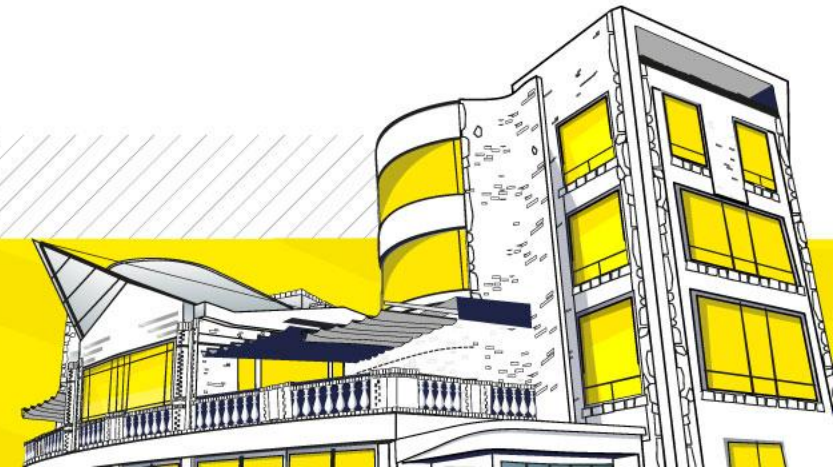
PRESENTACIÓN

Este software fue desarrollado como opción de grado “Desarrollo Tecnológico” para la carrera de ingeniería de sistemas y tiene como finalidad establecer un espacio para administrar los avances en los proyectos de grado de los usuarios pertenecientes a la Maestría en Calidad y Gestión integral de la universidad Santo Tomás.



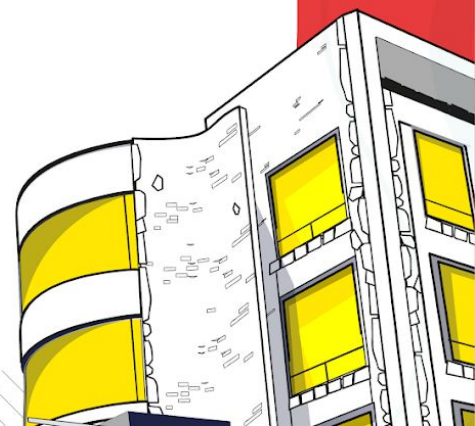
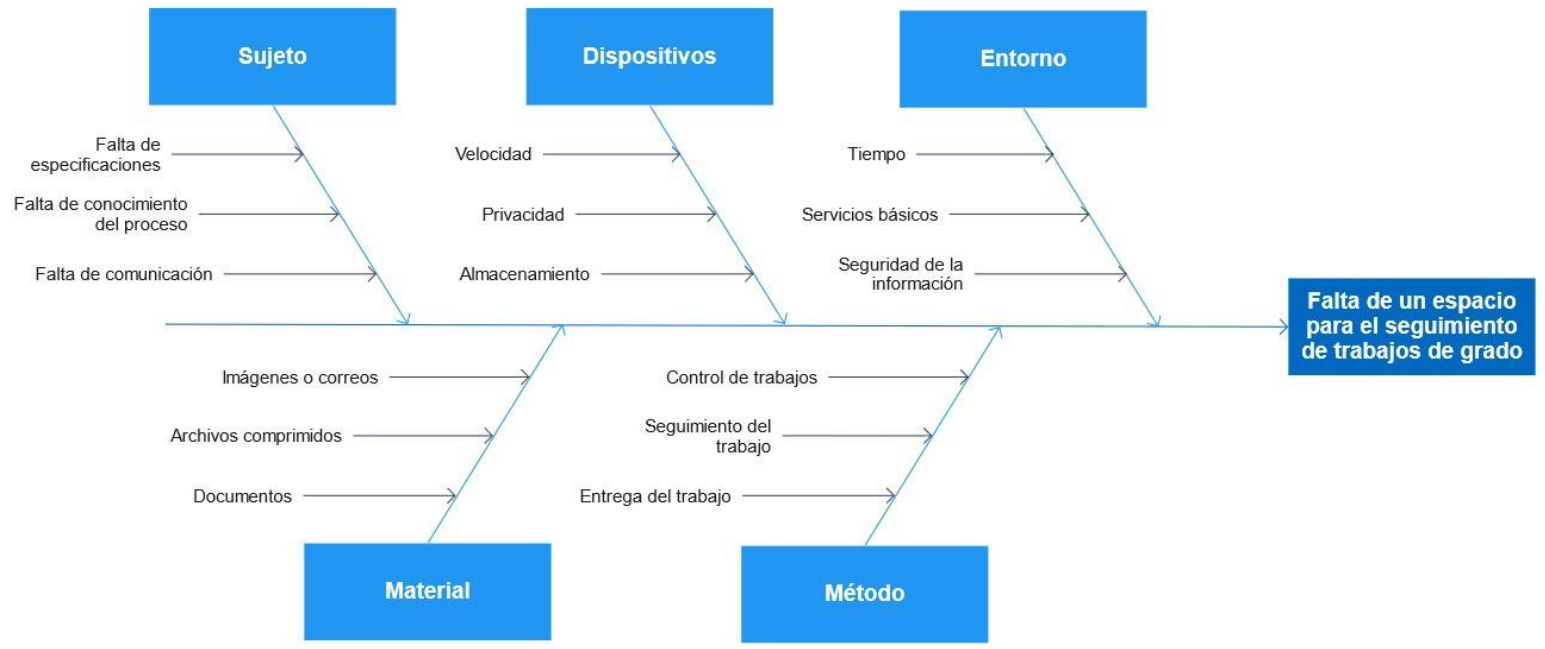
PLANTEAMIENTO DEL PROBLEMA

- El desarrollo de trabajos de grado a nivel de maestría tiene un alto nivel de complejidad, el cual requiere seguimiento y control para lograr el éxito del proyecto. Donde gran parte de trabajos de grado a nivel de maestría no cuentan con un seguimiento y control adecuados.
- Actualmente se realiza el seguimiento y control de los trabajos de maestría de manera manual, con el incremento de la cantidad de trabajos tanto en ejecución como los trabajos entregados, se ha generado un alto nivel de dificultad a la hora de realizar seguimiento y consultar los proyectos finalizados.
- Los directores utilizan sus propias herramientas para hacer el seguimiento de los proyectos, sin embargo, para la universidad es un inconveniente la ausencia de una herramienta para tener un control detallado del proceso de avance de sus trabajos de maestría.



PLANTEAMIENTO DEL PROBLEMA

Diagrama de Problemas



JUSTIFICACIÓN

Dada la complejidad de los trabajos de maestría, es de vital importancia llevar un control de los avances de cada proyecto de forma detallada. Se determinó las siguientes razones para construcción del software:

- Evitar la pérdida de información entre cada avance presentado por el estudiante.
- Cumplir y mantener registro de los tiempos de desarrollo de los proyectos de maestría.
- Evitar tareas repetitivas y consolidar toda la información de estos trabajos en un solo sitio.
- Accesibilidad para todos los involucrados.



OBJETIVO GENERAL

Construir una plataforma informática para la gestión y control de los avances de los proyectos de la Maestría en Calidad y Gestión Integral de la Universidad Santo Tomás, utilizando herramientas de código abierto para el desarrollo de la plataforma y una arquitectura orientada a servicios para la definición de las capas del proyecto.



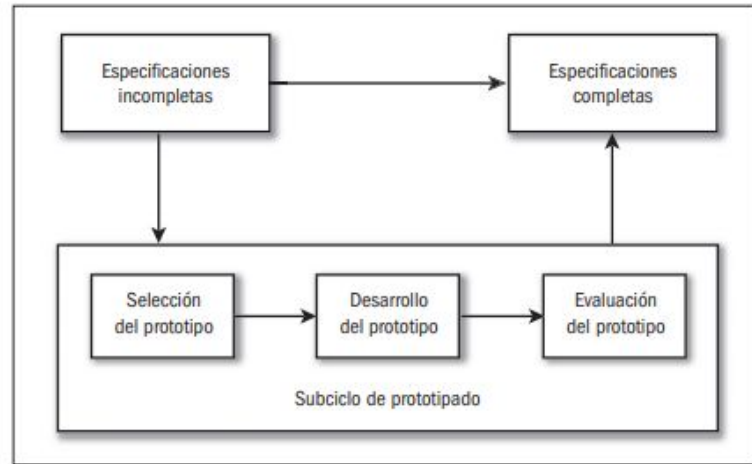
OBJETIVOS ESPECÍFICOS

1. Construir la especificación de requerimientos funcional del proyecto, utilizando la herramienta REM(Requirements management) propuesta por Amador Duran en su tesis doctoral.
2. Diseñar la arquitectura de la plataforma utilizando el modelo de vistas 4+1 propuesto por Philippe Kruchten, para definir los diseños requeridos en la fase de desarrollo.
3. Implementar una plataforma Web para el control y seguimiento de los trabajos de la maestría en Calidad y Gestión Integral de la Universidad Santo Tomás, incluyendo los siguientes módulos:
 - a. Autorización y autenticación.
 - b. Registro de trabajos de maestría.
 - c. Control de versiones de los trabajos entregados.
 - d. Asignación de recursos: Humano y otros.
 - e. Seguimiento de los avances del proyecto.
 - f. Consolidado de los trabajos de grado.
4. Asegurar la funcionalidad de la plataforma Web por medio de pruebas controladas donde se enfocarán en técnicas para pruebas unitarias y de integración de aplicaciones.



MODELO DE DESARROLLO

Modelo de desarrollo por Prototipos



Por medio de prototipos se desarrollan todas las funcionalidades previstas para el producto final.

Al determinar el modelo de desarrollo, se definió las fases de Software Development Life Cycle (SDLC):

- 1. Recopilación y análisis de requisitos.*
- 2. Estudio de viabilidad.*
- 3. Diseño del sistema.*
- 4. Desarrollo de la aplicación.*
- 5. Pruebas.*
- 6. Puesta en producción.*

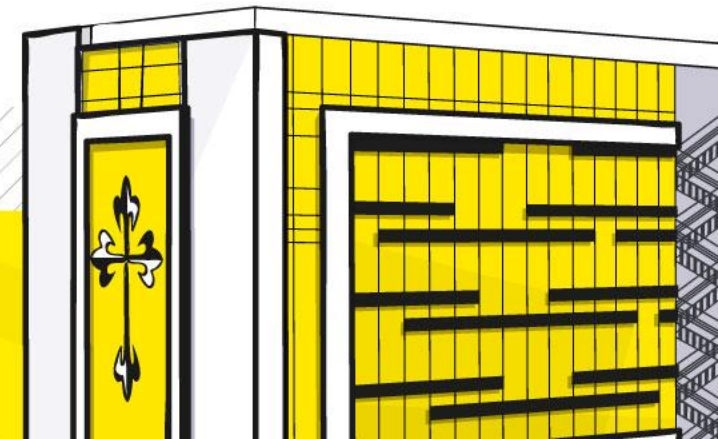
DESARROLLO DEL PROYECTO - Fase 1. Recopilación y análisis de requisitos

Construcción de Especificación de Requisitos de Software(ERS) - Anexo 1

NÚMERO	REQUERIMIENTOS FUNCIONALES
RF01	Registrar/Modificar Usuario.
RF02	Ingreso de Usuario.
RF03	Visualizar perfil de Usuario.
RF04	Crear Proyectos o Trabajo de grado.
RF05	Modificar/Actualizar Proyectos o Trabajos de grado.
RF06	Visualizar el Proyecto o Trabajo de grado.
RF07	Visualizar versiones anteriores.
RF08	Listar Proyectos o Trabajos de grado.
RF09	Asignar usuarios a Proyectos o Trabajos de grado.
RF10	Crear comentarios en cada Proyecto o Trabajo de grado.
RF11	Visualizar comentarios en cada Proyecto o Trabajo de grado.

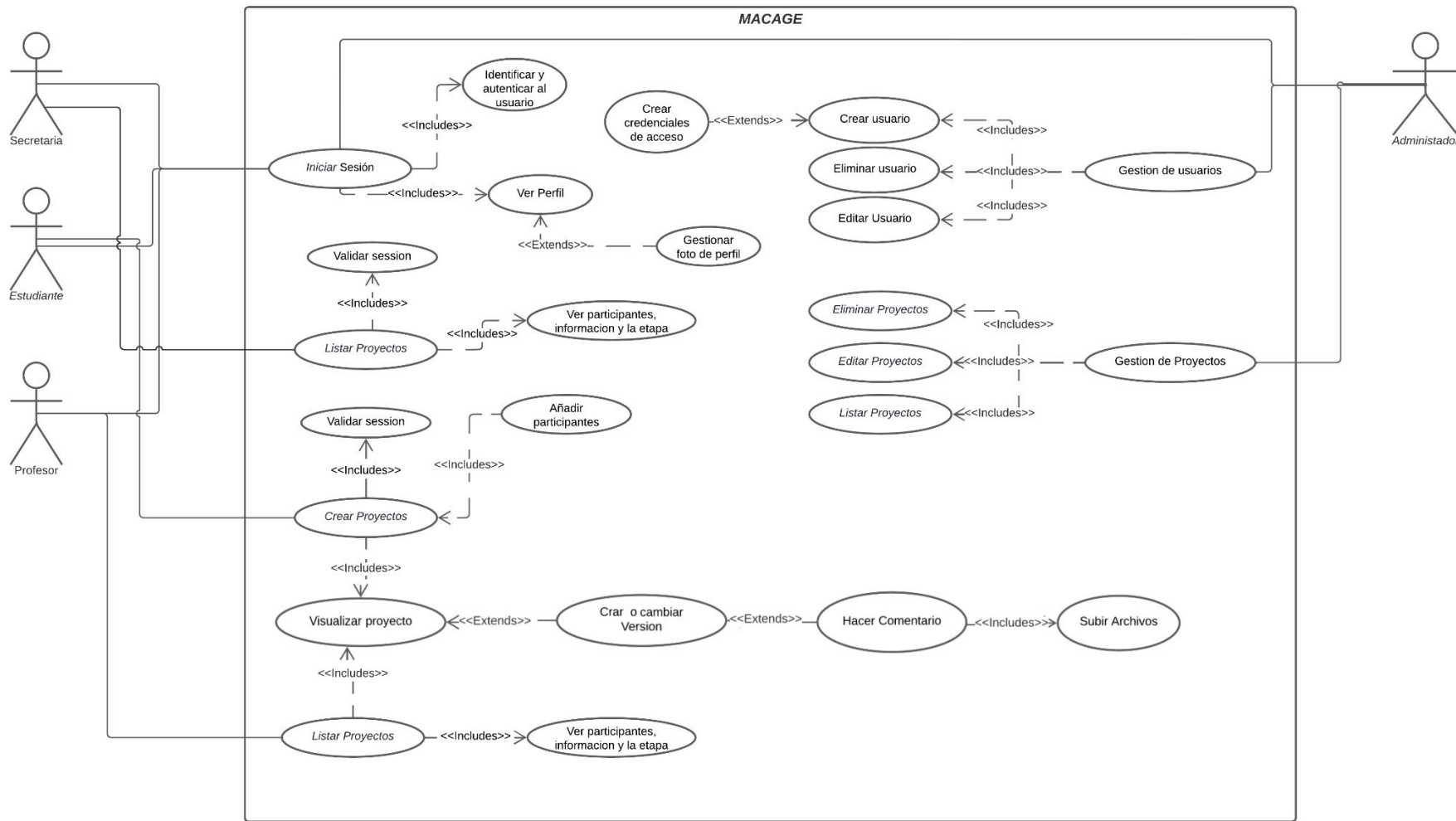
REQUERIMIENTOS NO FUNCIONALES			
Identificación de Requerimientos	NRF 01	Fecha	26/08/2021
Nombre de requerimiento	Disponibilidad.		
Descripción del requerimiento	El sitio deberá funcionar en cualquier hora y momento por el medio que el usuario desee usarlo.		
Criterio de aceptación	Los usuarios podrán ver el contenido de la página y los empleados podrán modificarla.		
Prioridad del requerimiento	Media		
Responsable de requerimientos	Juan Diego Gomez G. y Jose Alejandro Garcia Preciado		
Usuarios	Todos los usuarios.		

https://docs.google.com/document/d/1NYhI8uwe_FcF7wqhR7s2VL-n7IIwe5eIBX0k97I5kMs/edit?usp=sharing



DESARROLLO DEL PROYECTO - Fase 1. Recopilación y análisis de requisitos

Diagrama casos de uso



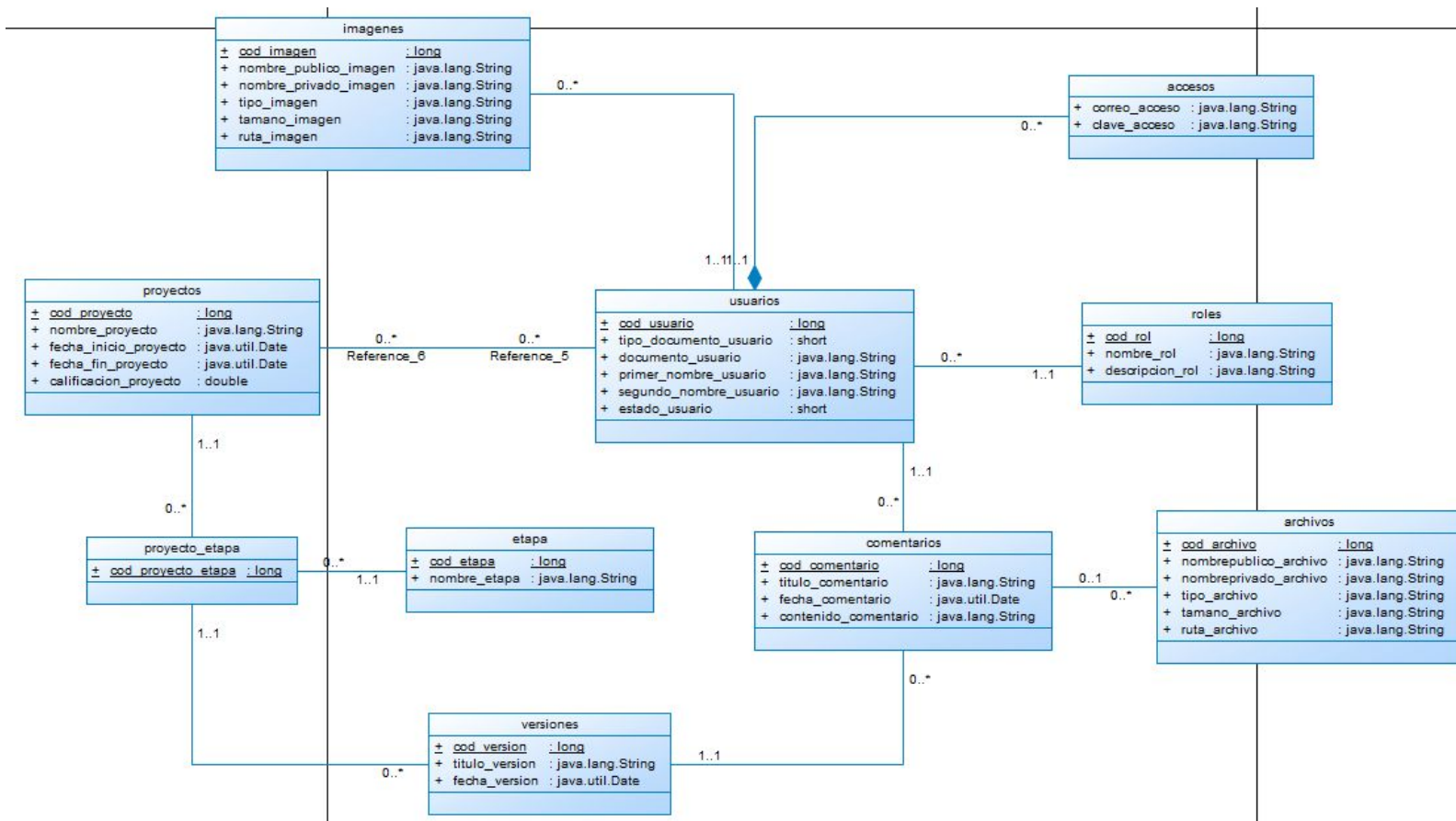
DESARROLLO DEL PROYECTO - Fase 2. Estudio de viabilidad.

Definición de Arquitectura por Modelo de Vistas 4 + 1 propuesto por P. Kruchten.



DESARROLLO DEL PROYECTO - Fase 2. Modelo de vistas 4 + 1

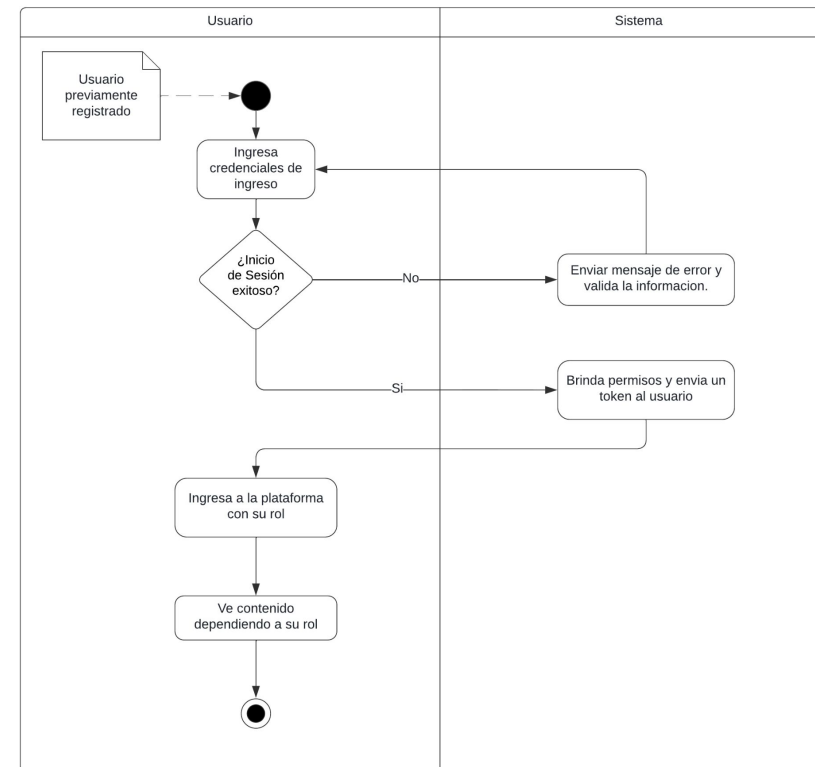
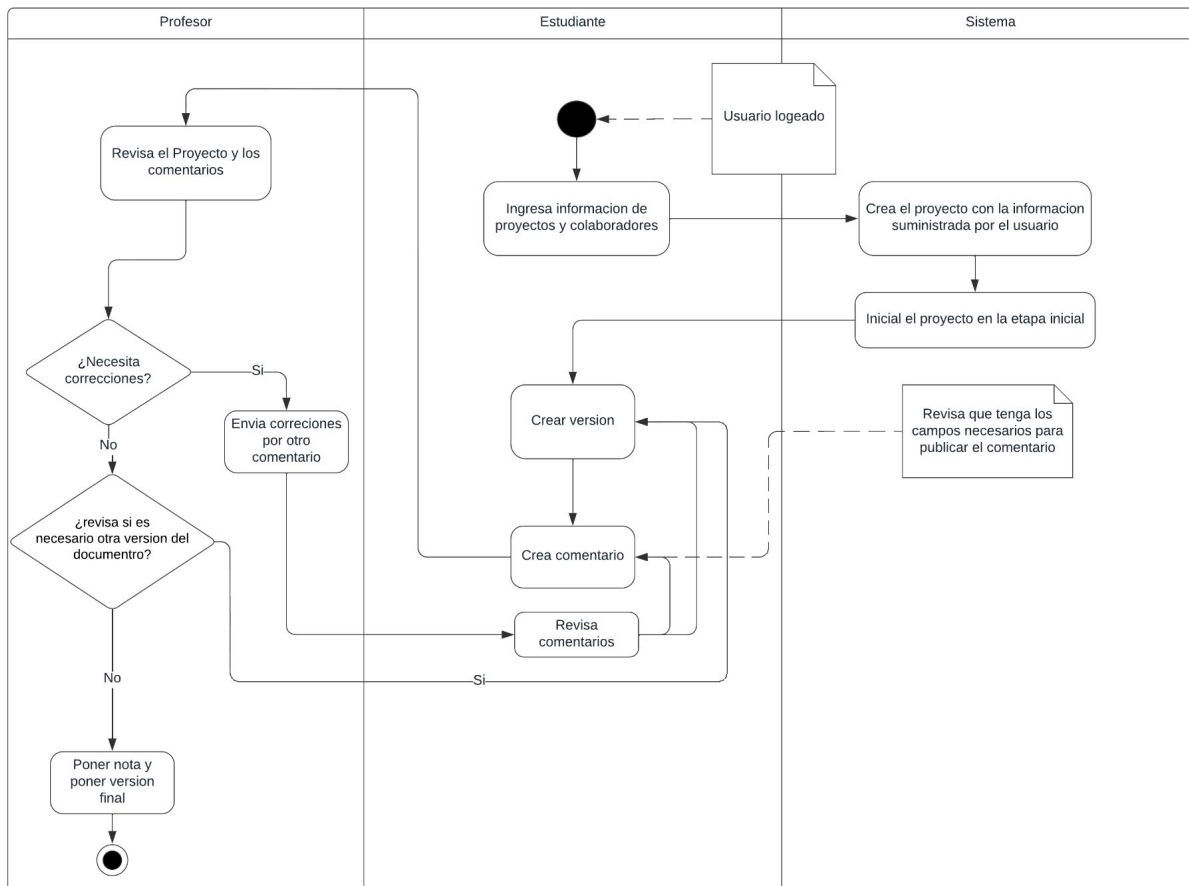
Vista Lógica - Diagrama de Clases



Representa la funcionalidad que el sistema proporcionará a los usuarios. Es decir, se ha de representar lo que el sistema debe hacer, y las funciones y servicios que ofrece.

DESARROLLO DEL PROYECTO - Fase 2. Modelo de vistas 4 + 1

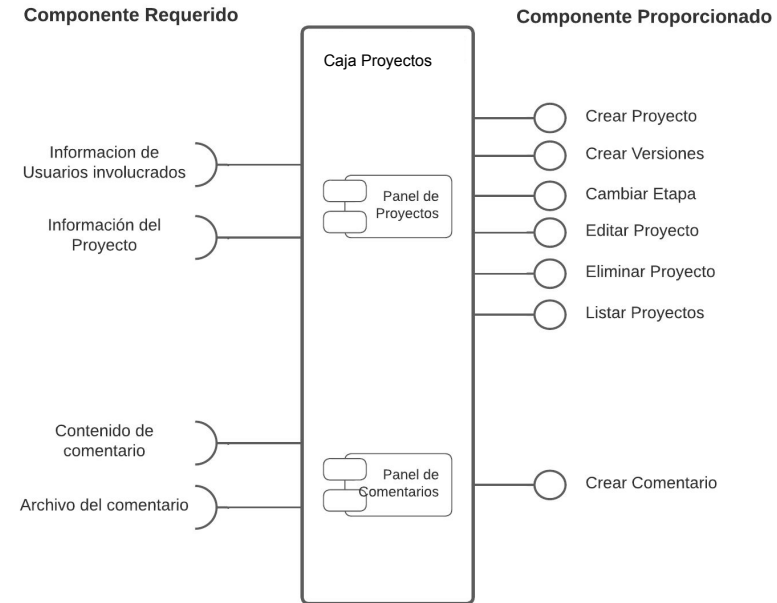
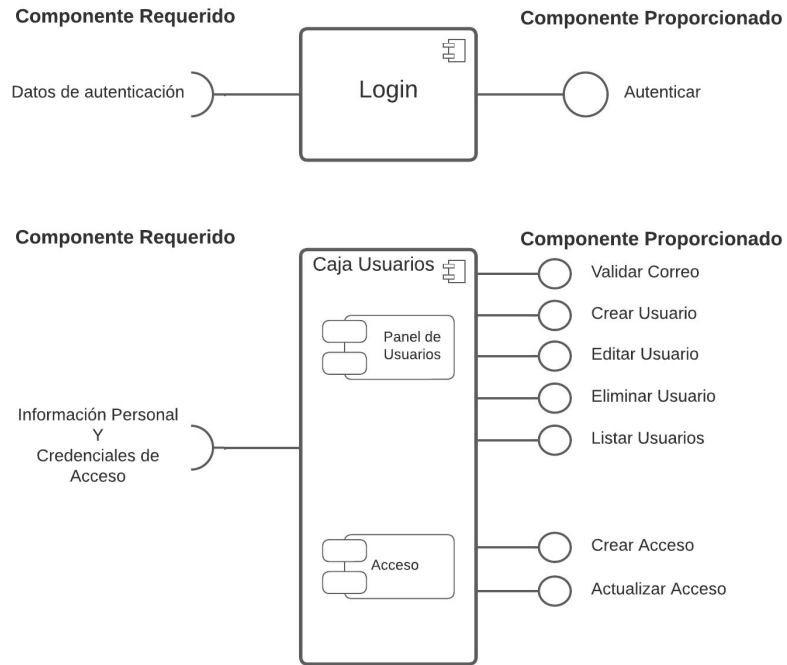
Vista de Procesos - Diagrama de Actividad



Permite ver los procesos que existen en el sistema y la forma en la que se comunican estos; describe el flujo de trabajo paso a paso de negocio y operacionales que conforman el sistema.

DESARROLLO DEL PROYECTO - Fase 2. Modelo de vistas 4 + 1

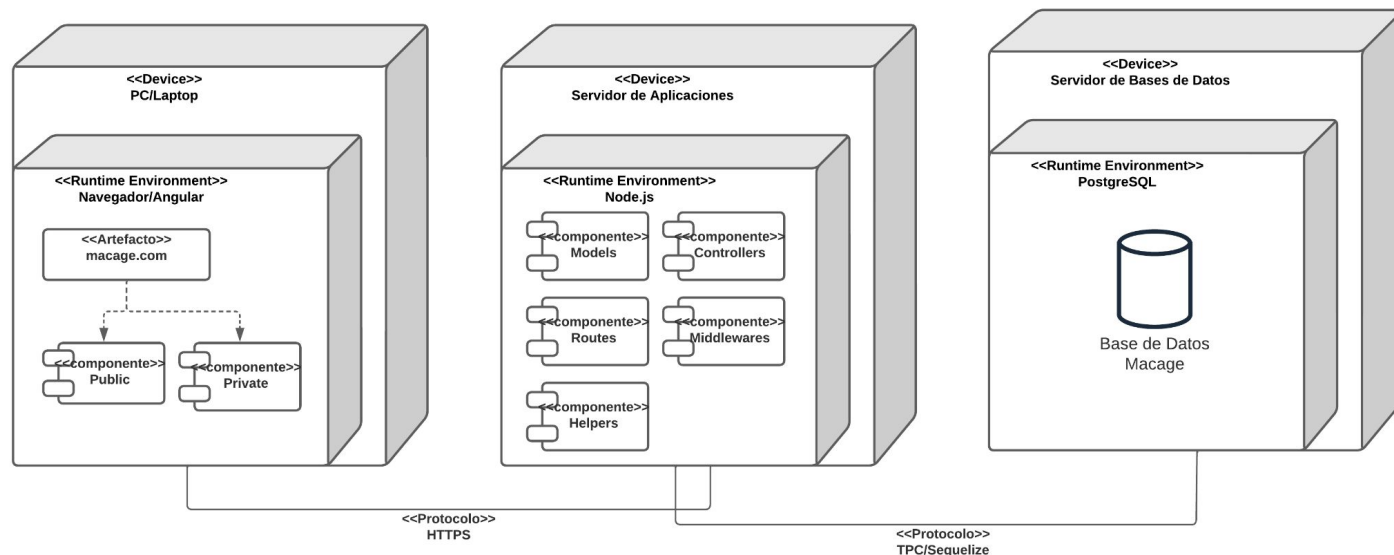
Vista de Despliegue - Diagrama de componentes



Perspectiva de un desarrollador división del sistema software en componentes y las dependencias que hay entre estos.

DESARROLLO DEL PROYECTO - Fase 2. Modelo de vistas 4 + 1

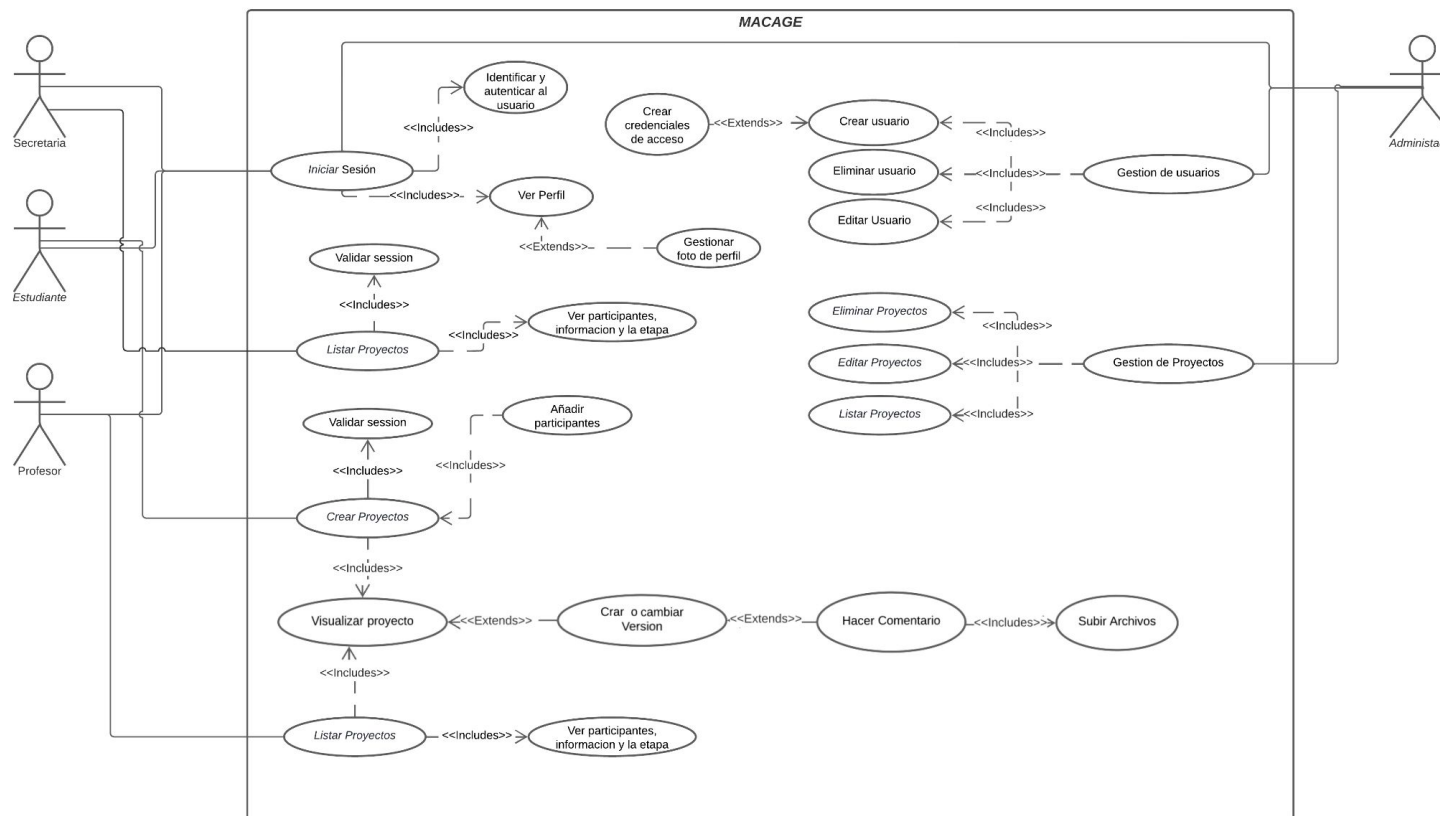
Vista Física - Diagrama de despliegue



Perspectiva de nodos y cómo se conforman todos los componentes del sistema, así como las conexiones entre estos componentes que conforman la solución.

DESARROLLO DEL PROYECTO - Fase 2. Modelo de vistas 4 + 1

+1 Vista de Escenarios - Diagrama de Casos de Uso

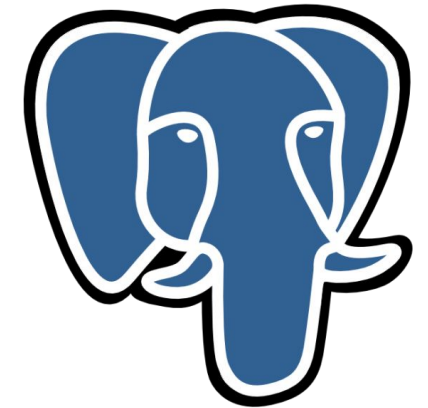


Luego de definir estas vistas tenemos claridad en la arquitectura del software; el paso a seguir para establecer las tecnologías a usar, el tiempo de desarrollo y la viabilidad de la implementación del proyecto.

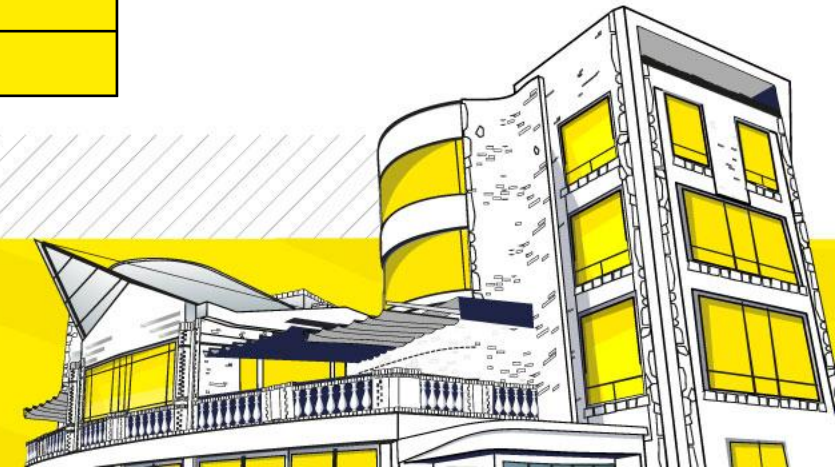
DESARROLLO DEL PROYECTO - Fase 3. Diseño del sistema.

Definición de Base de Datos.

TABLA	DESCRIPCIÓN
roles	Roles que existen dentro de la aplicación definidos como Administrador, Estudiante, Profesor y Secretaria.
usuarios	Información de usuarios.
accesos	Credenciales de acceso para cada usuario registrado.
imagenes	Imágenes de perfil de cada usuario y modificación de las mismas.
usuario_proyecto	Tabla de rompimiento para unir usuarios con proyectos.
proyectos	Información de los proyectos.
etapas	Estado de los proyectos.
proyecto_etapa	Tabla de rompimiento para definir estados a los proyectos.
versiones	Versión del proyecto para el manejo de comentarios.
comentarios	Observaciones a definir según cada proyecto.
archivos	Archivos vinculados a usuarios y comentarios dentro de las versiones de los proyectos.



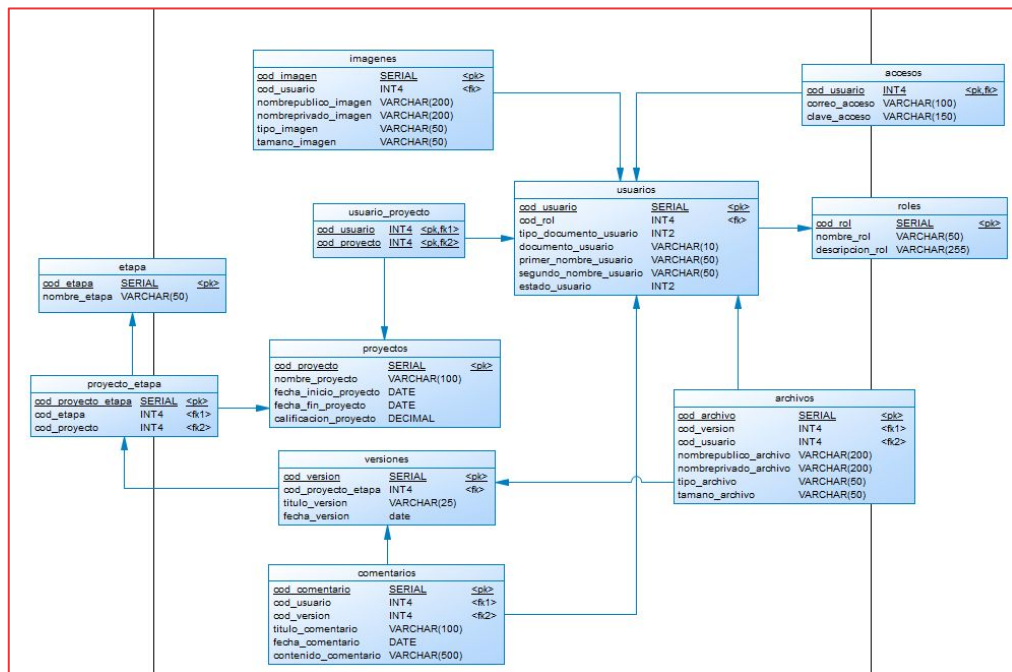
PostgreSQL



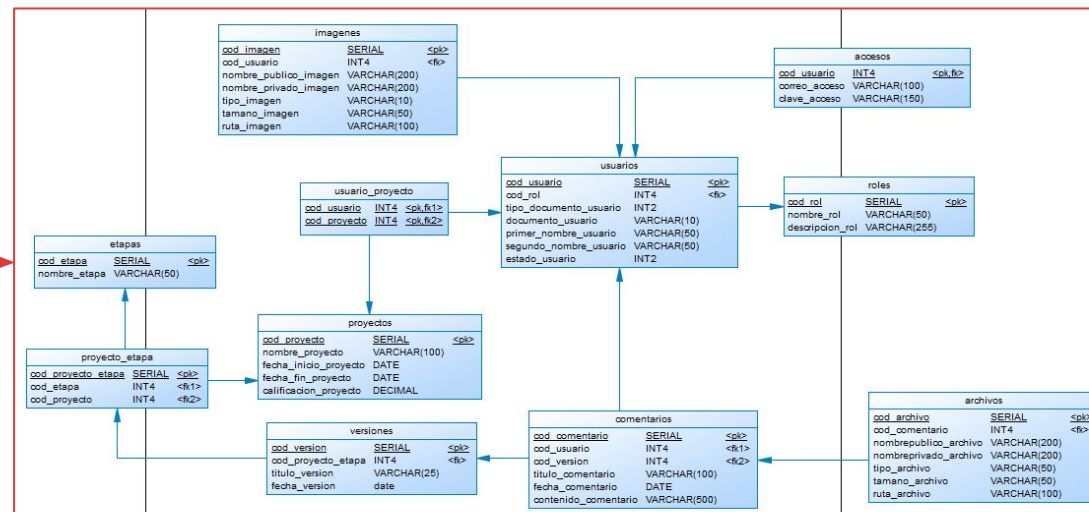
DESARROLLO DEL PROYECTO - Fase 3. Diseño del sistema.

Modelo Entidad – Relación MACAGE

Versión 1



Versión 2

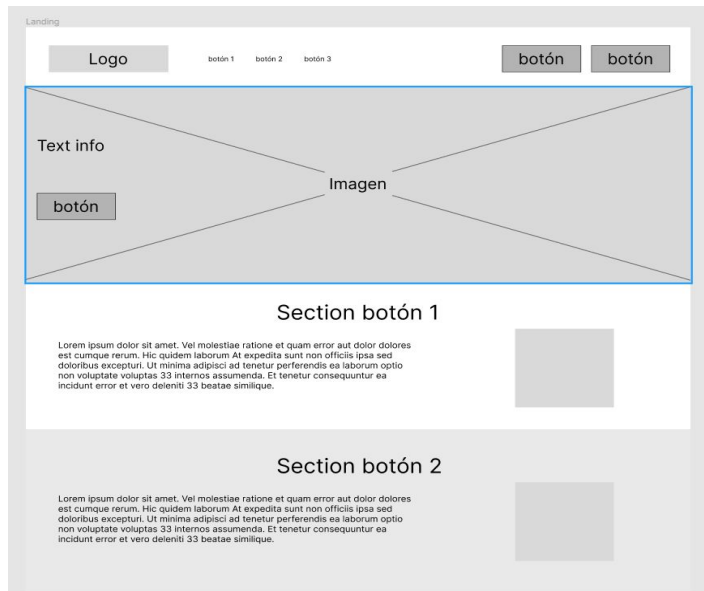


- Cambio para optimizar tiempo y entendimiento de plataforma (Reducir versiones para mejorar el uso de los comentarios).
- Adicionalmente se realizaron configuraciones y mejoras generales como las siguientes: Nombres de tablas y campos en español con una estructura de uso de minúsculas y guión bajo donde se use más de una palabra, Campos nulos y obligatorios y Constraints (Llaves primarias, Campos únicos y Llaves foráneas).

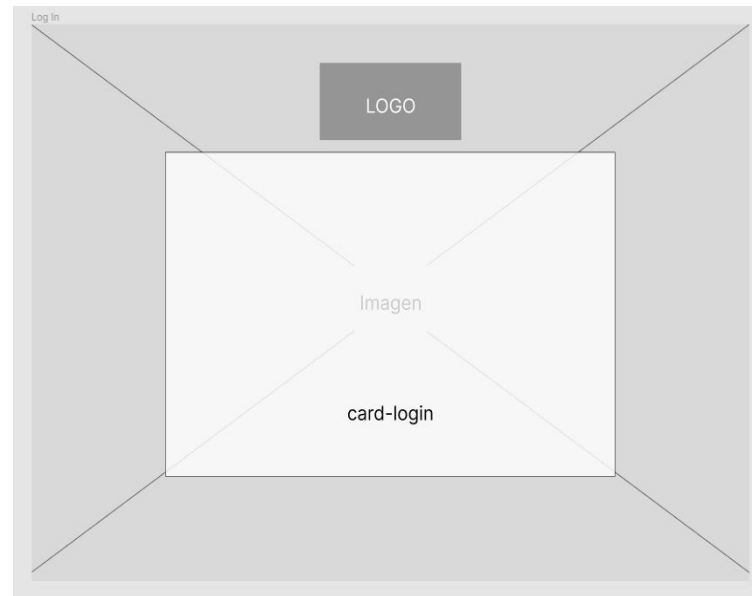
DESARROLLO DEL PROYECTO - Fase 3. Diseño del sistema.

Diseño de software y definición de colores.

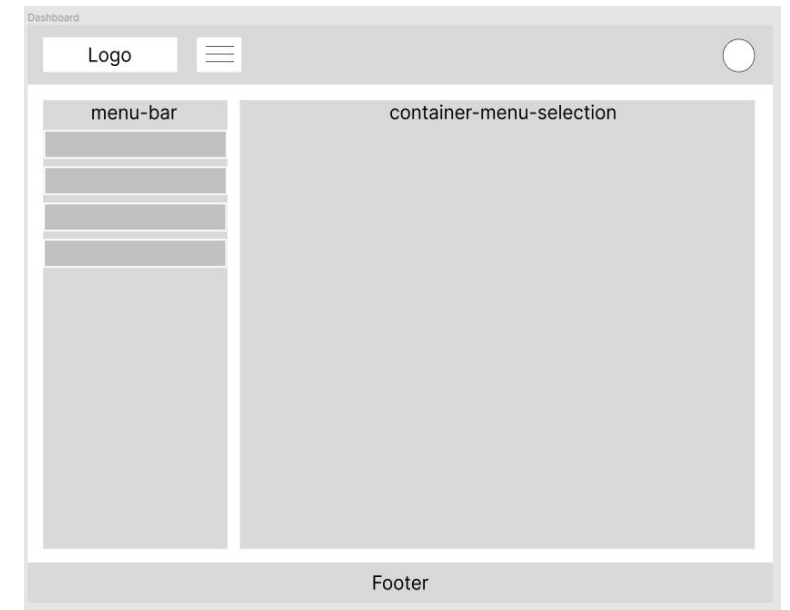
WIREFRAMES DEL SOFTWARE



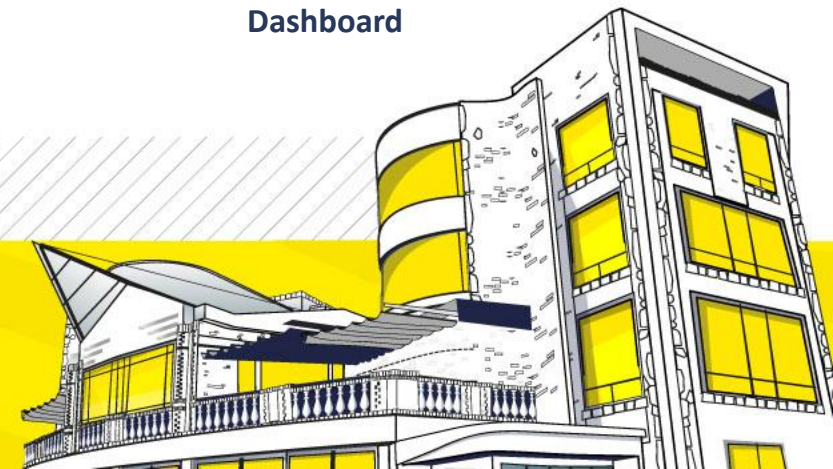
Landing Page



Login Page



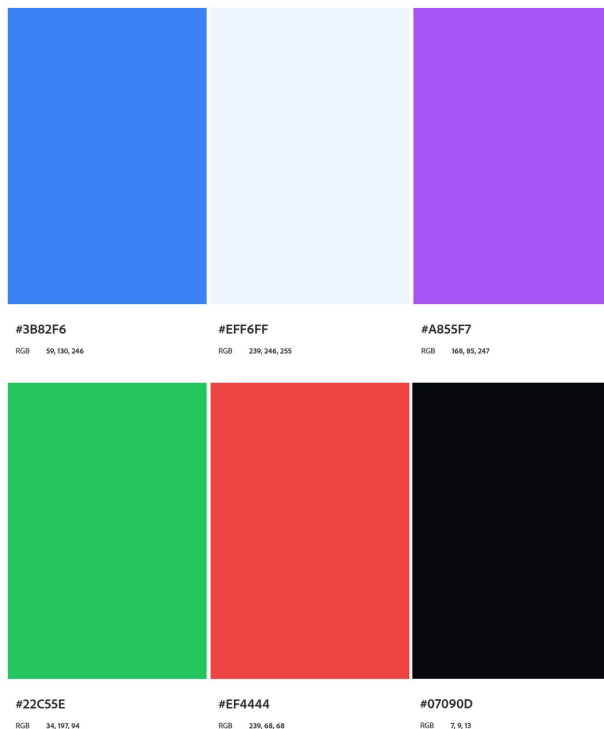
Dashboard



DESARROLLO DEL PROYECTO - Fase 3. Diseño del sistema.

Diseño de software y definición de colores.

Paleta de colores



Definición de colores

Color	Codigo Hex	Uso
Morado	#A855F7	Se empleó para algunos botones como el de editar.
Blanco	#EFF6FF	Se empleó para el fondo de la aplicación y para el texto de los botones.
Azul	#3B82F6	Se empleó para botones, representación de selecciones y títulos principales
Verde	#22C55E	Se empleó para botones de confirmación
Rojo	#EF4444	Se utilizó para representar errores y botones de eliminación
Negro	#07090D	Se empleó para el texto de la aplicación y algunos campos en los formularios



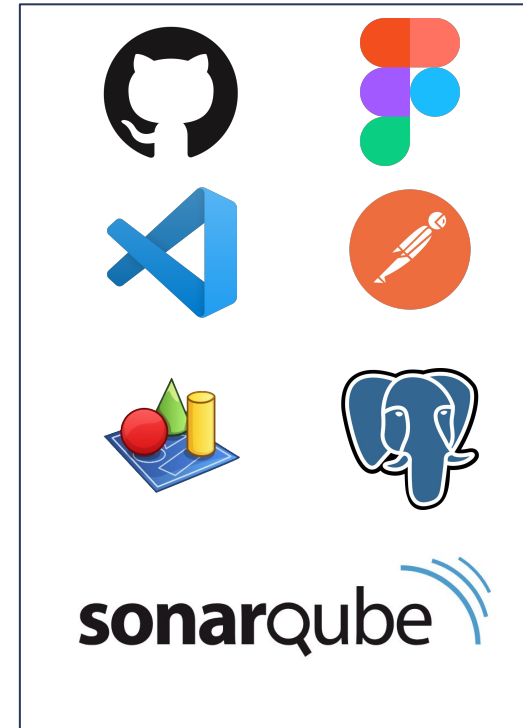
DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Tecnologías usadas en el proyecto.

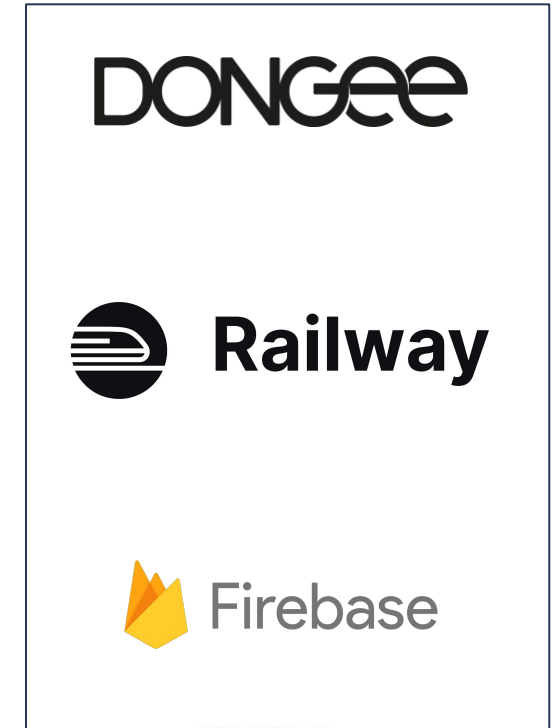
Desarrollo



Herramientas



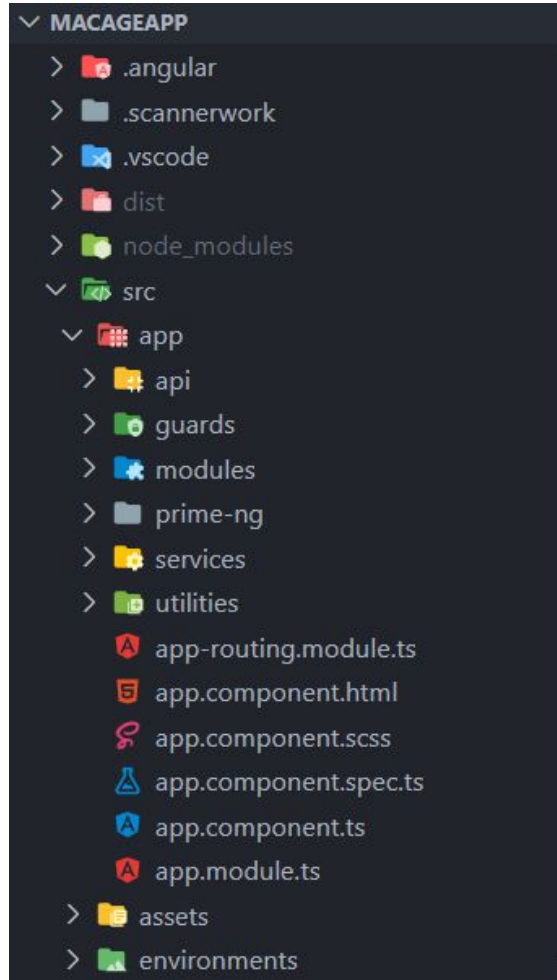
Despliegue



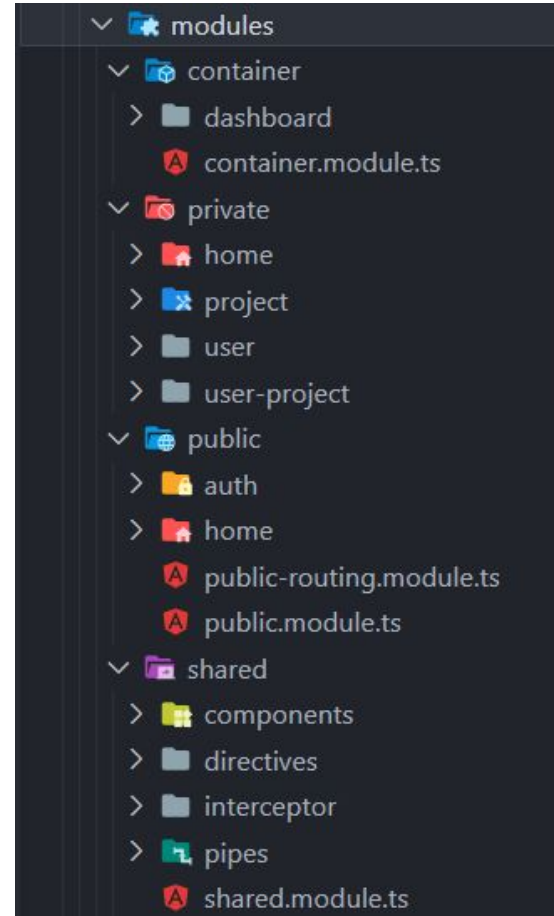
DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Estructura Macage App Frontend

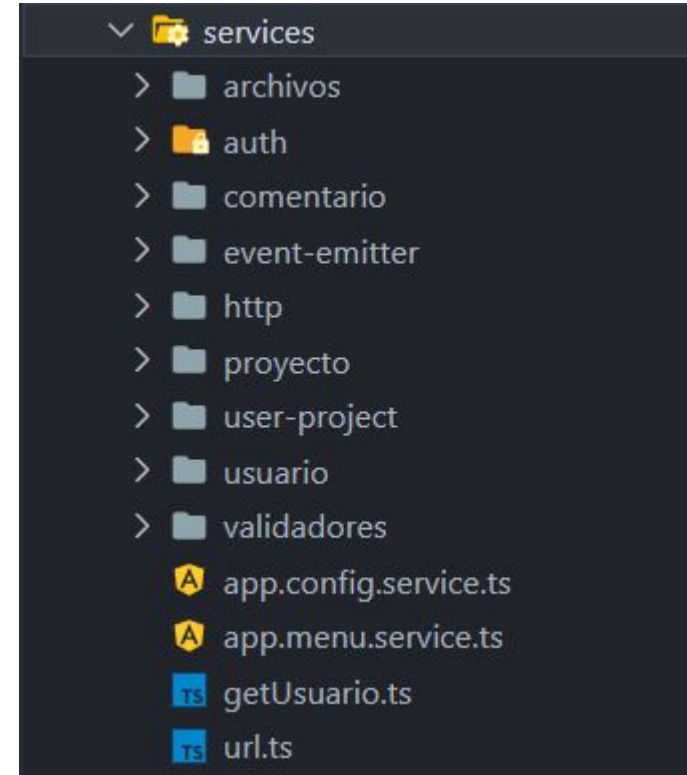
Estructura General



Módulos y Componentes



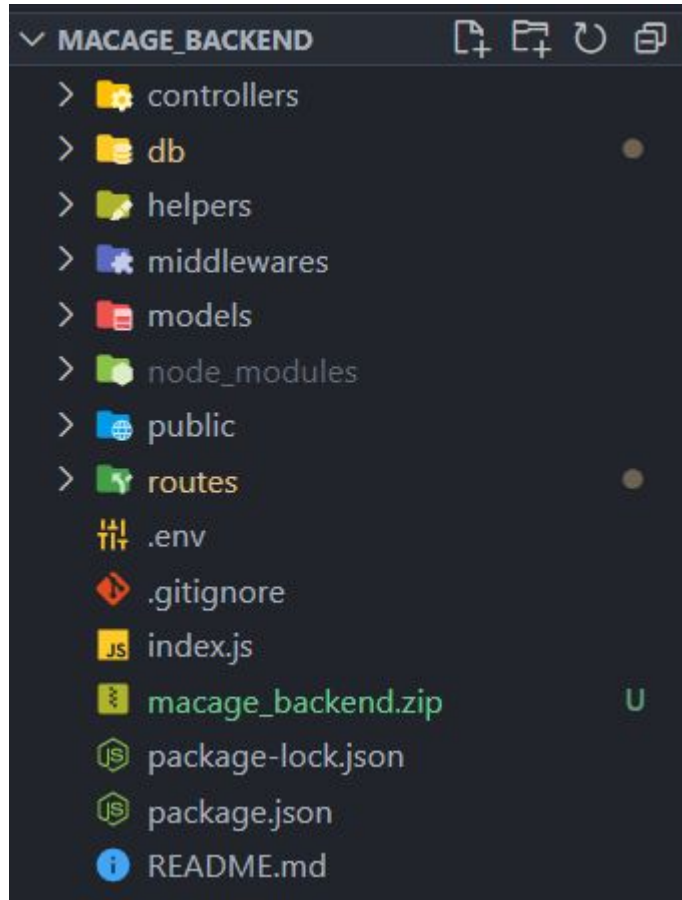
Servicios



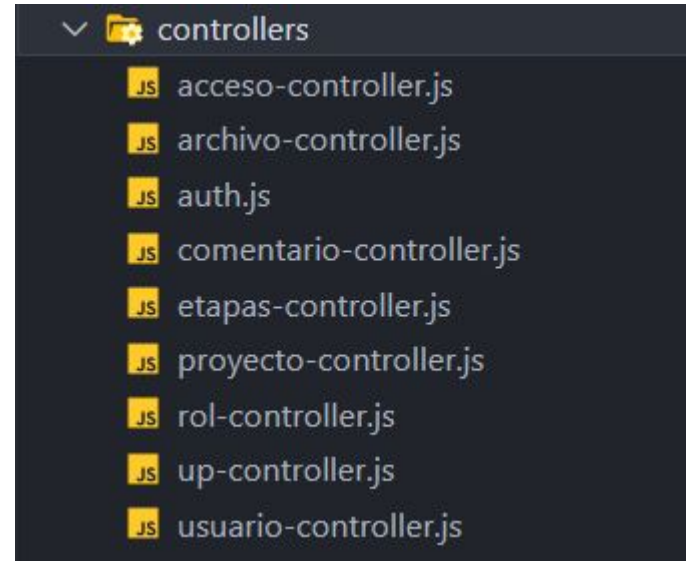
DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Estructura Macage App Backend

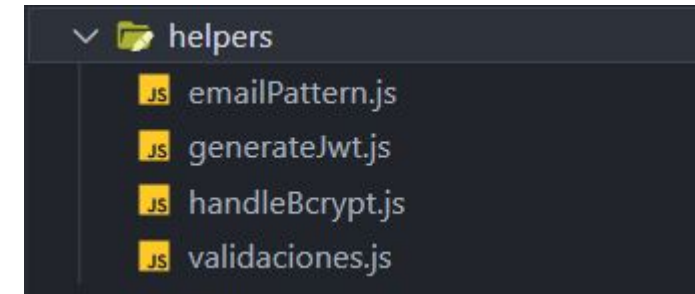
Estructura General



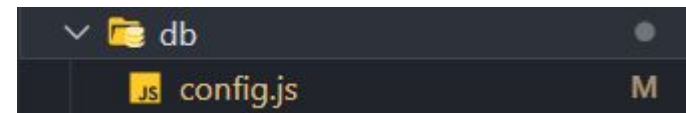
Controladores



Helpers



Conexión con la BD



DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Estructura Macage App Backend

Middleware

```
▼ middlewares
  JS auth.js
  JS authRole.js
  JS validar-campos.js
  JS validar-jwt.js
```

Modelos

```
▼ models
  JS acceso.js
  JS archivo.js
  JS comentario.js
  JS etapa.js
  JS imagenes.js
  JS proyecto_etapa.js
  JS proyecto.js
  JS rol.js
  JS usuario_proyecto.js
  JS usuario.js
  JS version.js
```

Data del servidor

```
▼ public
  > archives
  > images
  index.html
```

Rutas

```
▼ routes
  acceso.routes.js
  archivos.routes.js
  auth.routes.js
  comentario.routes.js
  etapas.routes.js
  proyecto.routes.js
  roles.routes.js
  up.routes.js
  usuarios.routes.js
  validate.routes.js
```



DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Resultados



Iniciar Sesión

Desarrollo Tecnológico Opción de Grado - USTA

"Módulo de administración de trabajos de grado Maestría en Calidad y Gestión Integral"



Características de la App Tecnologías utilizadas...

JS

Java Script

Lenguaje de programación interpretado, dialecto del estándar ECMAScript.

TS

Typescript

Lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft.



NodeJs

Entorno en tiempo de ejecución multiplataforma de código abierto, basado en el lenguaje de programación JavaScript.



Visual Studio Code

Editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web.



Angular

Angular es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google.

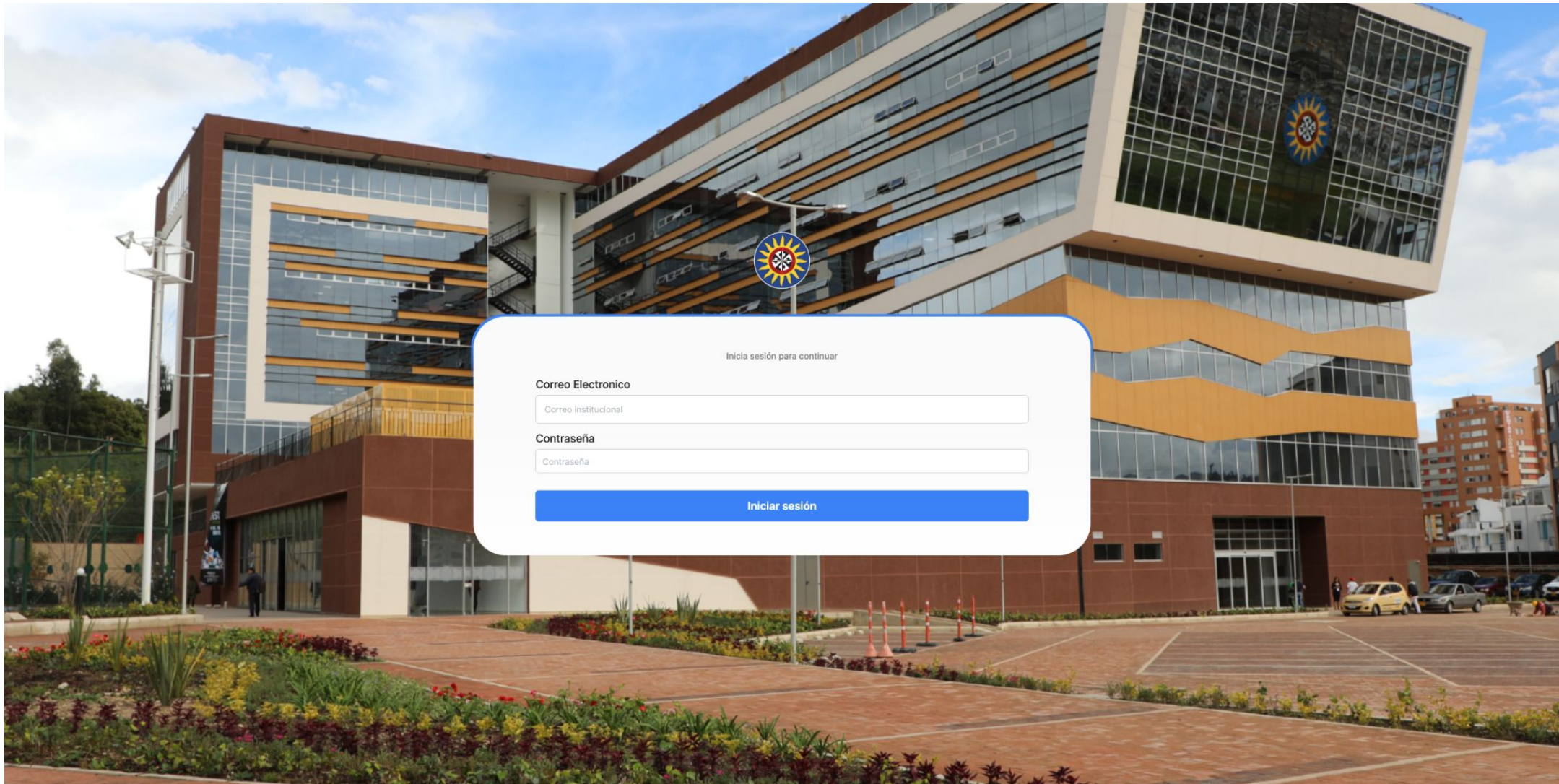


PrimeNg

PrimeNG es una colección de componentes de interfaz de usuario para Angular.

DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Resultados



DESARROLLO DEL PROYECTO - Fase 4. Desarrollo de la aplicación.

Resultados

The screenshot displays the MACAGE application interface. At the top left, the logo 'MACAGE' is visible. The main content area features a large banner image of the Universidad Santo Tomás Tunja building, overlaid with the university's circular logo and the text 'UNIVERSIDAD SANTO TOMÁS TUNJA'. Above the banner, the text reads 'Bienvenido a MACAGE!!' and 'Administrador Macage'. On the left side, there is a sidebar menu with the following items: 'INICIO' (containing 'Dashboard'), 'USUARIOS' (containing 'Crear Usuario' with an 'OK' button and 'Listar Usuarios'), and 'PROYECTOS' (containing 'Crear Proyecto', 'Listar Proyectos', and 'Listar Proyectos Finalizados'). In the top right corner, there are navigation icons for 'Ver Perfil' and a user profile icon.

DESARROLLO DEL PROYECTO - Fase 5. Pruebas.



Escaneo en aspectos de Tamaño, Mantenibilidad, Código innecesario, Fiabilidad y Seguridad.

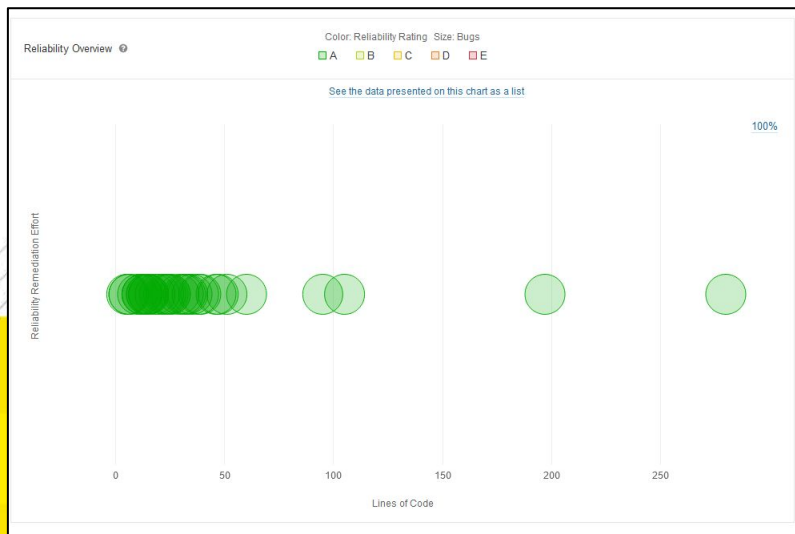
SonarQube Macage-Backend

Remove this commented out code. Sections of code should not be commented out. javascriptS125

Code Smell Major Open Not assigned 5min effort 0 comments

Where is the issue? Why is this an issue?

```
117 andr... }
118 };
119 andr... // try {
120 //   const { cod_version } = req.params;
121 // } catch (e) {
122 //   // const comentarios = await Comentario.findAll({
123 //     //   where: {
124 //     //     cod_version,
125 //     //   },
126 //     //   attributes: ['cod_comentario'],
127 //     //   include: [
128 //     //     {
129 //     //       model: Archivo,
130 //     //       attributes: ['nombre_publico_archivo', 'ruta_archivo'],
131 //     //     },
132 //     //   ],
133 //     // });
134 // }
135 // // const archivo = await Archivo.findOne({
136 // //   //   where: { cod_comentario },
137 // //   //   attributes: ['nombre_publico_archivo', 'ruta_archivo'],
138 // //   // });
139 // // }
140 // // if (archivo) {
141 // //   //   return res.status(200).json({
142 // //     //     ok: true,
```



SonarQube Macage-Frontend

Size	
Lines of Code	51,199
Lines	53,693
Statements	684
Functions	337
Classes	60
Files	125
Comment Lines	323
Comments (%)	0.6%

Maintainability	
Overview	
Overall	
Code Smells	150
Debt	1d 3h
Debt Ratio	0.0%
Rating	A

Maintainability	
Overview	
Overall	
Code Smells	2
Debt	10min
Debt Ratio	0.0%
Rating	A

Security	
Overview	
Overall	
Vulnerabilities	0
Rating	A
Remediation Effort	0

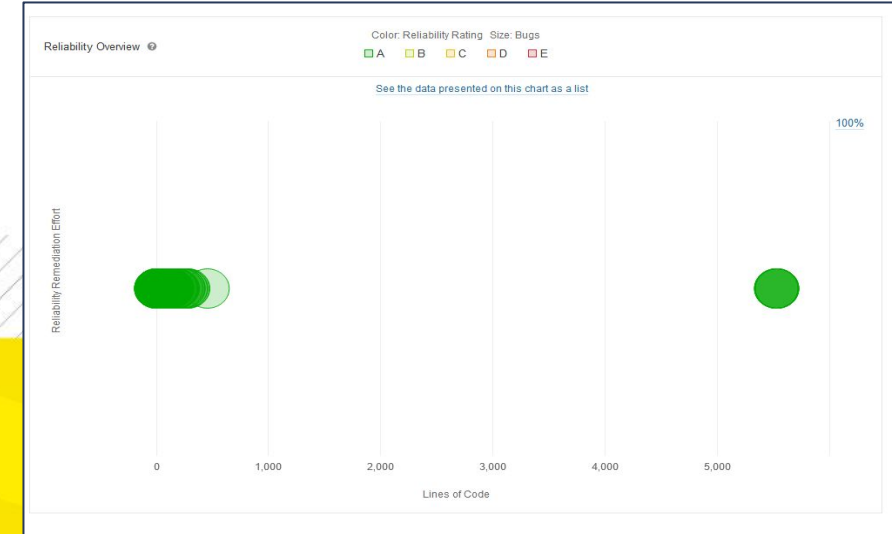
Security Review	
Overall	
Security Hotspots	0
Rating	A
Security Hotspots Reviewed	100%

Remove this commented out code. Sections of code should not be commented out. typescriptS125

Code Smell Major Open Not assigned 5min effort 0 comments

Where is the issue? Why is this an issue?

```
114 andr... }
115 }
116 }
117 getProyectosById() {
118   this.proyectoService.getProyectoById(this.cod_proyecto).subscribe((res) => {
119     // console.log('...' + JSON.stringify(res.proyecto));
120   });
121 }
122
123 this.proyecto = res.proyecto;
124 this.usuarios = this.proyecto['usuario_proyectos'];
125 //*****
126 this.numEstudiantes = 0;
127 console.log(this.numEstudiantes);
128 this.usuarios.forEach((object) => {
129   if (object.usuario['role'].nombreRol === 'Estudiante') {
130     this.numEstudiantes++;
131   }
132 });
```



DESARROLLO DEL PROYECTO - Fase 6. Puesta en producción.



MacageDB production

PostgreSQL

Data Query Connect Logs Variables Metrics Settings

Tables

- acessos
- archivos
- comentarios
- etapas
- imagenes
- proyecto_etapa
- proyectos
- roles
- usuario_proyecto
- usuarios
- versiones

Created 2022-09-14

Create table

PostgreSQL

Data Query Connect Logs Variables Metrics Settings

Variables are automatically provided to builds, deploys, and CLI "run" commands. Some are provided automatically. Docs

DATABASE_URL	*****
PGDATABASE	railway
PGHOST	containers-us-west-41.railway.app
PGPASSWORD	mswIMsNQqH53r9D36knq
PGPORT	6593
PGUSER	postgres

MacageApp

cPanel - Node.js

Node.js

WEB APPLICATIONS MACAGE.GRUPOATA.CO/API

Enter to the virtual environment. To enter to virtual environment, run the command `source /home/grupoata/nodeenv/mcage_grupoata.co/api/10bin/activate: 64 of /home/grupoata/mcage_grupoata.co/api`

DESTROY CANCEL SAVE

Node.js

STOP APP RESTART

Node.js version: 10.24.1

Application mode: Development

Application root: mcage_grupoata.co/api

Application URL: mcage_grupoata.co/api

Node.js

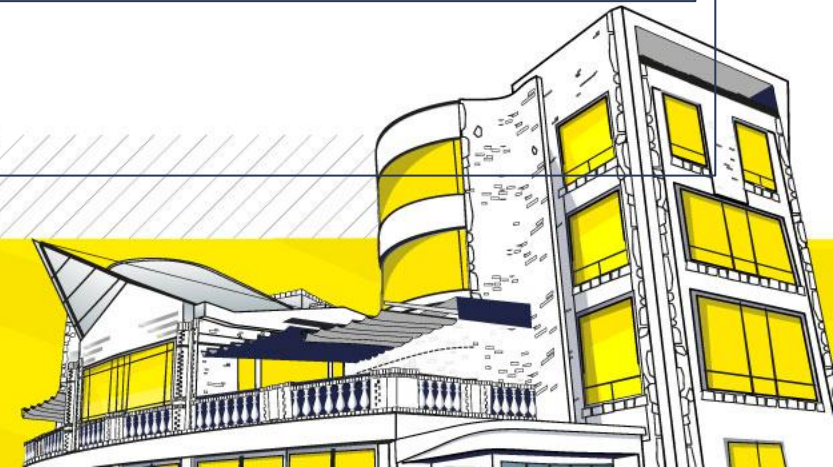
WEB APPLICATIONS

App URL	App Root Directory	Mode	Status	Actions
mcage_grupoata.co/api	/home/grupoata/mcage_grupoata.co/api	development	started (v10.24.1)	

CREATE APPLICATION



Name	Size	Last Modified	Type	Permissions
.scannerwork	46 bytes	Yesterday, 11:20 PM	httpd/unix-directory	0755
controllers	4 KB	Yesterday, 11:38 PM	httpd/unix-directory	0755
db	22 bytes	Yesterday, 11:20 PM	httpd/unix-directory	0755
helpers	93 bytes	Yesterday, 11:20 PM	httpd/unix-directory	0755
middlewares	83 bytes	Yesterday, 11:20 PM	httpd/unix-directory	0755
models	4 KB	Yesterday, 11:20 PM	httpd/unix-directory	0755
public	51 bytes	Yesterday, 11:39 PM	httpd/unix-directory	0755
routes	4 KB	Yesterday, 11:20 PM	httpd/unix-directory	0755
tmp	24 bytes	Yesterday, 11:28 PM	httpd/unix-directory	0755
.env	38 bytes	Yesterday, 11:20 PM	text/x-generic	0644
.gitignore	407 bytes	Yesterday, 11:20 PM	text/x-generic	0644
.htaccess	541 bytes	Yesterday, 11:28 PM	text/x-generic	0644
index.js	1.19 KB	Yesterday, 11:20 PM	text/x-generic	0644
node_modules	66 bytes	Yesterday, 11:28 PM	httpd/unix-directory	0777
package-lock.json	45.38 KB	Yesterday, 11:20 PM	text/x-generic	0644
package.json	618 bytes	Yesterday, 11:20 PM	text/x-generic	0644
README.md	264 bytes	Yesterday, 11:20 PM	text/x-generic	0644



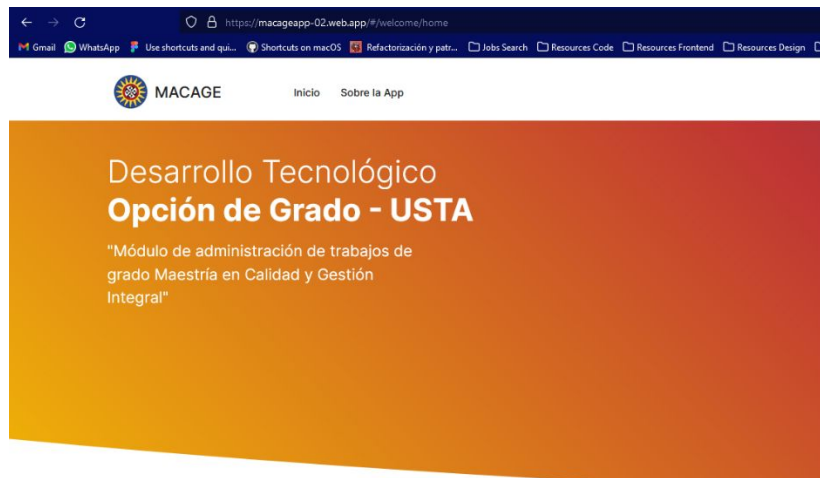
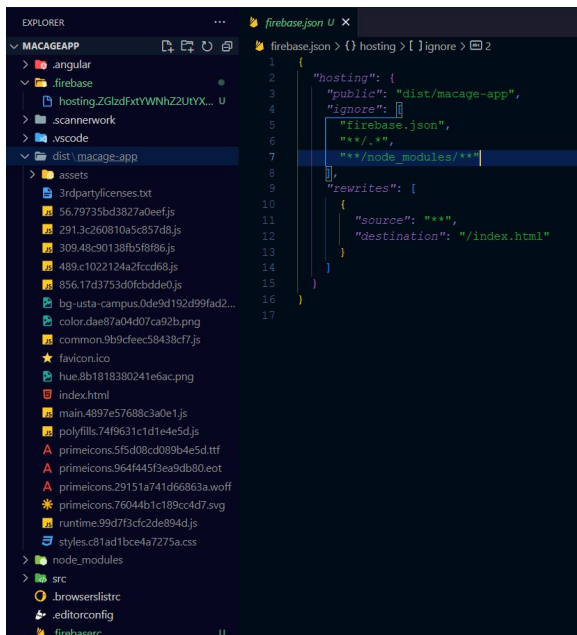
DESARROLLO DEL PROYECTO - Fase 6. Puesta en producción.



\$ npm install -g firebase-tools

```
PS C:\Users\juand\Documents\Proyecto de grado\Frontend\macageApp> npm run build
> macage-app@0.0.0 build
> ng build

✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.
```



<https://macageapp-02.web.app>

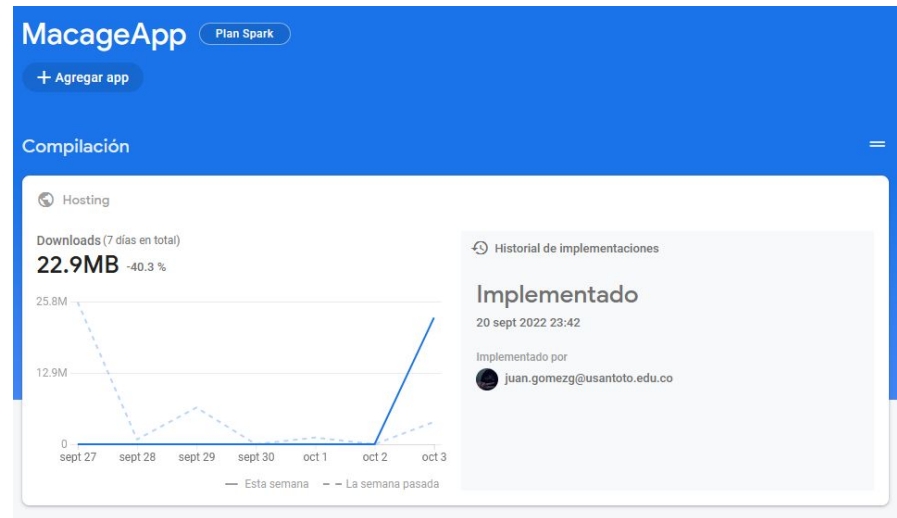
```
PS C:\Users\juand\Documents\Proyecto de grado\Frontend\macageApp> firebase deploy

== Deploying to 'macageapp-02'...

i deploying hosting
i hosting[macageapp-02]: beginning deploy...
i hosting[macageapp-02]: found 78 files in dist/macage-app
+ hosting[macageapp-02]: file upload complete
i hosting[macageapp-02]: finalizing version...
+ hosting[macageapp-02]: version finalized
i hosting[macageapp-02]: releasing new version...
+ hosting[macageapp-02]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/macageapp-02/overview
Hosting URL: https://macageapp-02.web.app
```



CONCLUSIÓN

- Se desarrolló con éxito la plataforma para la gestión y control de proyectos de la maestría en Calidad y Gestión Integral con uso de herramientas de código abierto como lo fue Angular, Node.js, Express y PostgreSQL. Lo que ayudó al desarrollar el proyecto fue el uso de la metodología de desarrollo en prototipos lo que permitió mantener mejoría constante en el desarrollo de la arquitectura del proyecto que se enfocó en construir microservicios y consumirlos.
- Se especificó un documento con los requerimientos funcionales y no funcionales del proyecto necesarios para la construcción, haciendo uso de las técnicas de Amador Duran para definir el alcance que tendría el software.
- Se implementó el uso del modelo de arquitectura de vistas 4+1 propuesto por P. Kruchten por medio de diagramas, necesarios para la definición de viabilidad del proyecto y sus fases de desarrollo.
- Tras el desarrollo y análisis de los módulos creados para el correcto funcionamiento de la plataforma web, se evidencia que cada uno de estos cumple a cabalidad su esperado funcionamiento.
- Se aseguró la funcionalidad de la plataforma con el escaneo al código fuente del proyecto proporcionado por SonarQube, el cual demostró la calidad del software.





SANTOTO

— Seccional Tunja —



SC4289-1

www.ustatunja.edu.co

♥ 🔍 📍 @SANTOTOMeSTunja 📌