



**UNIVERSIDAD SANTO TOMÁS**  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
T U N J A

**TÍTULO PROPUESTA**

GENERADOR DE BACKENDS A PARTIR DE MAPEADORES EN BASE DE DATOS  
POSTGRESQL CON LA LIBRERÍA TYPEORM

**PROPONENTE(S)**

Luis Felipe Chaparro Hurtado

1057609409

2273178

Juan Pablo Robles Arias

1002397519

2272055

**DIRECTOR**

Diego Alejandro Vela Beltrán



Tunja, Boyacá  
2024

## CONTENIDO

<b>1. FICHA TÉCNICA DEL PROYECTO.....</b>	<b>4</b>
<b>2. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>5</b>
<b>3. JUSTIFICACIÓN.....</b>	<b>8</b>
<b>4. OBJETIVOS.....</b>	<b>9</b>
4.1. OBJETIVO GENERAL .....	9
4.2. OBJETIVOS ESPECÍFICOS.....	9
<b>5. ESTADO DEL ARTE .....</b>	<b>10</b>
<b>6. MODELO DE DESARROLLO.....</b>	<b>14</b>
<b>7. DESARROLLO DEL PROYECTO .....</b>	<b>16</b>
7.1. FASE 1: ANÁLISIS .....	17
7.2. FASE 2: DISEÑO .....	18
7.3. FASE 3: IMPLEMENTACIÓN.....	19
7.4. FASE 4: VERIFICACIÓN.....	20
7.5. FASE 5: MANTENIMIENTO .....	21
<b>8. CONCLUSIONES Y ANÁLISIS DE RESULTADOS .....</b>	<b>23</b>
<b>9. REFERENCIAS.....</b>	<b>25</b>
<b>10. ANEXOS.....</b>	<b>27</b>
10.1. ANEXO: ANÁLISIS DE ENTREVISTA (DIRECTOR DEL SEMILLERO DE DESARROLLO DE SOFTWARE).....	27
10.2. ANEXO: ANÁLISIS DE ENCUESTAS (ESTUDIANTES DEL SEMILLERO DE DESARROLLO DE SOFTWARE).....	32
10.3. ANEXO: INVESTIGACIÓN INICIAL .....	39
10.4. ANEXO: CASOS DE USO .....	43

<b>10.5.</b>	<b>ANEXO: DIAGRAMA DE PROCESOS .....</b>	<b>44</b>
<b>10.6.</b>	<b>ANEXO: DIAGRAMAS DE SECUENCIA .....</b>	<b>47</b>
<b>10.7.</b>	<b>ANEXO: REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES .....</b>	<b>57</b>
10.7.1.	REQUERIMIENTOS FUNCIONALES.....	57
10.7.2.	REQUERIMIENTOS NO FUNCIONALES.....	70
<b>10.8.</b>	<b>ANEXO: SELECCIÓN DE TECNOLOGÍA .....</b>	<b>76</b>
<b>10.9.</b>	<b>ANEXO: ARQUITECTURA DEL SISTEMA .....</b>	<b>80</b>
<b>10.10.</b>	<b>ANEXO: VISUALIZACIÓN DE LA INTERFAZ DE USUARIO (MOCKUPS).....</b>	<b>82</b>
<b>10.11.</b>	<b>ANEXO: DIAGRAMA DE BASE DE DATOS.....</b>	<b>89</b>
<b>10.12.</b>	<b>ANEXO: DESARROLLO BACKEND.....</b>	<b>90</b>
<b>10.13.</b>	<b>ANEXO: DESARROLLO FRONTEND .....</b>	<b>93</b>
<b>10.14.</b>	<b>ANEXO: PRUEBAS UNITARIAS BACK-END.....</b>	<b>97</b>
<b>10.15.</b>	<b>ANEXO: PRUEBAS DE ACCESIBILIDAD Y MEJORES PRÁCTICAS DEL FRONTEND .....</b>	<b>105</b>
<b>10.16.</b>	<b>ANEXO: PRUEBAS DE EXPERIENCIA DE USUARIO.....</b>	<b>118</b>
<b>10.17.</b>	<b>ANEXO: PLAN DE PRUEBAS BASE DE DATOS (“COLEGIO”) .....</b>	<b>124</b>
<b>10.18.</b>	<b>ANEXO: PLAN DE PRUEBAS BASE DE DATOS (“EMPRESA”) .....</b>	<b>139</b>
<b>10.19.</b>	<b>ANEXO: PLAN DE PRUEBAS BASE DE DATOS (“TIENDA”).....</b>	<b>152</b>
<b>10.20.</b>	<b>ANEXO: MANUAL DE USUARIO .....</b>	<b>169</b>
<b>10.21.</b>	<b>ANEXO: MANTENIMIENTO Y CORRECCIONES DE LA META HERRAMIENTA.....</b>	<b>181</b>

## 1. FICHA TÉCNICA DEL PROYECTO

Título	Generador de backends a partir de mapeadores en base de datos PostgreSQL con la librería typeORM
Nombre Estudiante	Luis Felipe Chaparro Hurtado Juan Pablo Robles Arias
Documento estudiante	CC 1057609409 CC 1002397519
Correo electrónico estudiante	<a href="mailto:luis.chaparroh@usantoto.edu.co">luis.chaparroh@usantoto.edu.co</a> <a href="mailto:juan.roblesa@usantoto.edu.co">juan.roblesa@usantoto.edu.co</a>
Director	Diego Alejandro Vela Beltrán
Lugar de ejecución del proyecto	Tunja, Boyacá
Duración	4 meses
Palabras claves	CRUD, TypeORM, Metadatos, Backend
<p>Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente (incluir nombres y firmas de estudiantes y director).</p> <p></p> <p>Luis Felipe Chaparro Hurtado</p> <p></p> <p>Juan Pablo Robles Arias</p> <p>Firma del director Diego Alejandro Vela Beltrán</p>	

## 2. PLANTEAMIENTO DEL PROBLEMA

En el desarrollo de aplicaciones web, la interacción con bases de datos es esencial. Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) son fundamentales para la gestión de la información y constituyen la base de numerosas aplicaciones modernas. Sin embargo, la creación manual de código TypeScript para estas operaciones puede ser tediosa y propensa a errores. Cada vez que se incorpora una nueva entidad o tabla a la base de datos, los desarrolladores deben invertir tiempo en escribir y mantener el código correspondiente, consumiendo valiosos recursos.

En respuesta a esta problemática, surge la propuesta de proyecto, la cual consiste en utilizar mapeadores en bases de datos PostgreSQL para representar de manera lógica las estructuras relacionales dentro del entorno de un lenguaje de programación. Esta iniciativa ha generado la necesidad de diseñar y desarrollar un Generador de código eficiente para operaciones CRUD utilizando la librería TypeORM (Es un ORM, es decir una librería que se ejecuta en la plataforma Node.js y ofrece características adicionales que facilitan el desarrollo de diversos tipos de aplicaciones que emplean bases de datos) [1]. Este sistema analizará la estructura de la base de datos, identificará relaciones, y creará consultas y funciones. El resultado será una simplificación notable en el desarrollo web, al disminuir la carga de trabajo manual y minimizar los errores asociados a las operaciones del desarrollo backend.

No obstante, en la Figura 1 se evidencia una elevada inversión de tiempo en la generación de operaciones CRUD. Este problema puede derivar de diversas causas, tales como:

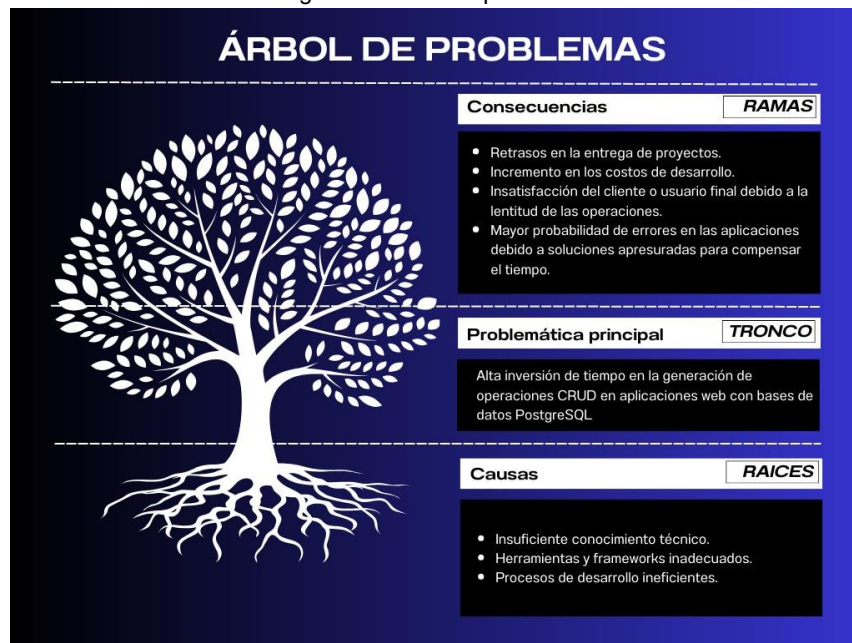
- Insuficiente conocimiento técnico: La carencia de conocimientos técnicos adecuados en el equipo de desarrollo puede resultar en la creación de códigos ineficientes y consultas poco mejoradas al realizar operaciones CRUD con PostgreSQL.
- Herramientas y frameworks inadecuados: La elección inapropiada de herramientas y frameworks para el desarrollo de la aplicación y la interacción con la base de datos puede ocasionar demoras en las operaciones CRUD.

- Procesos de desarrollo ineficientes: Un desarrollo desorganizado o la falta de una metodología efectiva para generar código CRUD pueden llevar a retrasos en la implementación.[2]

Estas causas pueden desencadenar una serie de consecuencias, tales como:

- Retrasos en la entrega de proyectos: Los retrasos en la implementación de operaciones CRUD pueden afectar los plazos acordados con los clientes y retrasar la entrega de proyectos web.
- Aumento de los costos de desarrollo: Los retrasos y la necesidad de abordar problemas ocasionados por operaciones CRUD ineficientes pueden incrementar los costos de desarrollo.
- Insatisfacción del cliente o usuario final: La lentitud en las operaciones CRUD puede generar insatisfacción entre los clientes o usuarios finales, afectando negativamente la percepción de la calidad del producto.
- Mayor probabilidad de errores en las aplicaciones: La implementación apresurada para compensar el tiempo perdido puede propiciar la aparición de errores en las aplicaciones [3].

Figura 1. Árbol de problemas

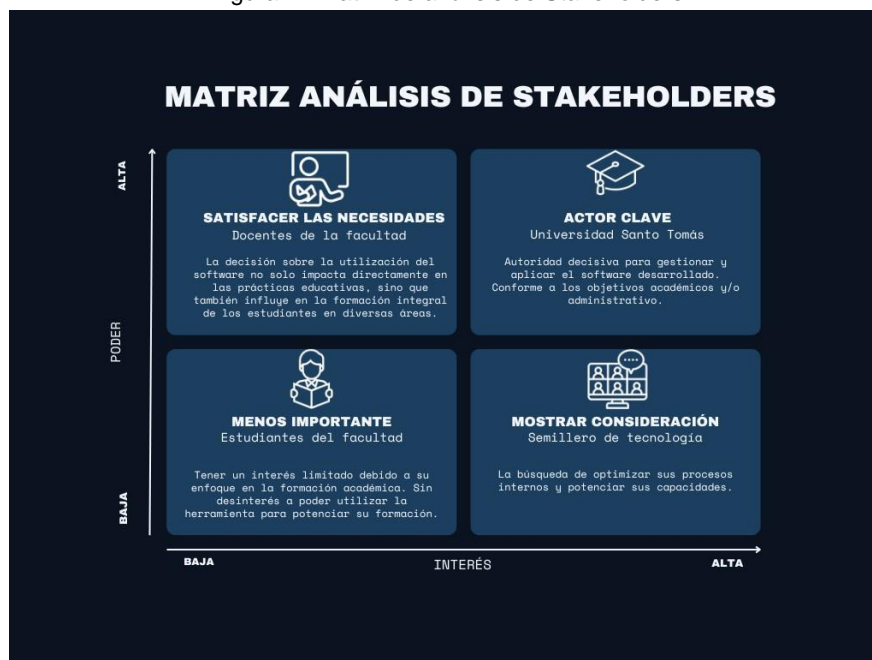


Fuente: Autores

Este proyecto no solo tiene implicaciones significativas para el desarrollo web en general, sino que también está directamente dirigido a la Universidad Santo Tomás y sus semilleros de desarrollo de software. Los principales stakeholders incluyen a los profesores y estudiantes involucrados en los semilleros, quienes se beneficiarán directamente de la herramienta propuesta. Además, la universidad como entidad educativa y promotora de la investigación y desarrollo tecnológico, puede utilizar esta herramienta como parte de su infraestructura tecnológica, contribuyendo así a la mejora continua de los procesos de desarrollo de software dentro de la institución.

La retroalimentación y colaboración activa de la universidad y los semilleros fueron esenciales para el éxito y la validación de este proyecto. En la Figura 2, se evidencia la matriz de análisis de Stakeholders, destacando la importancia estratégica de la participación y el respaldo de estos actores claves en el desarrollo y la implementación exitosa de la herramienta propuesta. La interacción continua con los stakeholders proporcionaron valiosos aportes y aseguraron que la herramienta cubriera las necesidades específicas de la comunidad académica y tecnológica de la Universidad Santo Tomás.

Figura 2. Matriz de análisis de Stakeholders



Fuente: Autores

### 3. JUSTIFICACIÓN

La implementación de una herramienta para generar código TypeScript en plataformas Backend, mediante el mapeo de metadatos de tablas en bases de datos relacionales, se presenta como una estrategia clave para abordar los desafíos del desarrollo de aplicaciones web. La construcción de componentes CRUD (Crear, Leer, Actualizar, Eliminar) es esencial pero laboriosa, propensa a errores y consume tiempo. La concepción de esta herramienta responde a la necesidad de automatizar y acelerar estos procesos, ofreciendo una solución centrada en la generación automatizada de código TypeScript.

La necesidad de simplificar y acelerar el proceso de desarrollo de aplicaciones web justifica el desarrollo de este Generador de Backends utilizando TypeORM. Este proyecto tiene el potencial de mejorar de manera significativa tanto la eficiencia como la calidad del desarrollo de software en el ámbito de las aplicaciones web, al automatizar la implementación de las operaciones CRUD y al proporcionar una estructura de código ágil, fiable y eficiente de mantener.

## 4. OBJETIVOS

### 4.1. Objetivo General

Implementar una meta herramienta para la generación de código TypeScript en plataformas Backend, mediante el mapeo de los metadatos de tipo tabla en bases de datos relacionales, para reducir tiempos en procesos de desarrollo de software que impliquen la construcción de componentes CRUD.

### 4.2. Objetivos específicos

Tabla 1. Objetivos específicos

Nro.	Objetivo específico
1	Definir los requerimientos funcionales de la meta herramienta, analizando tecnologías que permitan implementar desarrollos bajo el paradigma orientado a objetos y utilicen el patrón modelo vista controlador en los desarrollos Backend.
2	Implementar el módulo de autenticación multiplataforma para acceder a cualquier modelo relacional diseñado en PostgreSQL, con el fin de extraer los metadatos de las tablas y relaciones del motor seleccionado.
3	Generar el código Backend en TypeScript mediante el mapeo de la base de datos y el uso de marcos de trabajo para el acceso a los datos, incluyendo componentes para añadir, eliminar, actualizar y entregar información por bloques.
4	Diseñar el plan de pruebas requerido para implementar aplicaciones en tres contextos de bases de datos diferentes, obtenido el código generado para su análisis y revisión

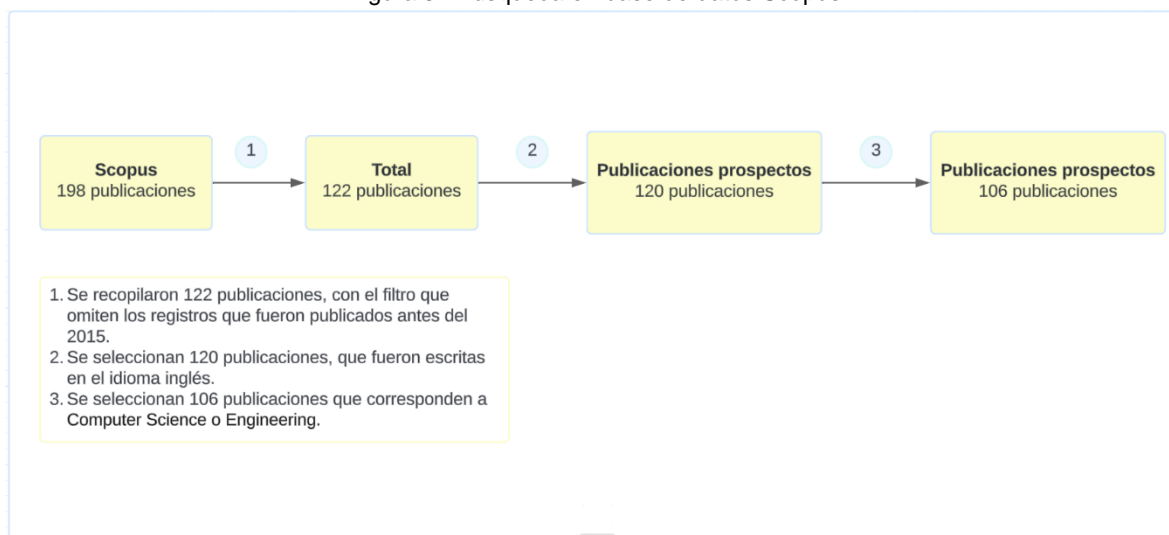
Fuente: Autores

## 5. ESTADO DEL ARTE

Se realizó una búsqueda exhaustiva en la base de datos Scopus para explorar las últimas tendencias y desarrollos tecnológicos relevantes para este proyecto. Esta búsqueda se centró específicamente en la formulación de la siguiente cadena de búsqueda: "**Backend**" + "**Code AND Generation**". Se realizaron los siguientes criterios de búsqueda:

- ✓ Las publicaciones a partir del año 2015 a 2023.
- ✓ Las publicaciones deben estar escritas en inglés.
- ✓ Las publicaciones deben estar enfocados en Computer Science o Engineering.

Figura 3. Búsqueda en base de datos Scopus



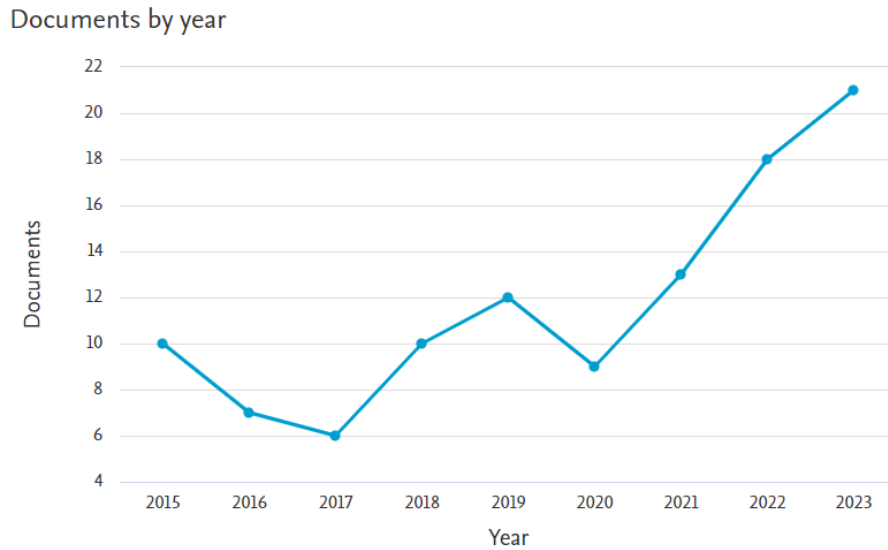
Fuente: Autores

Se evidencia que, para la base de datos Scopus, se recopilaron un total de 106 publicaciones, como se aprecia en la Figura 3, que se destacan por la identificación de técnicas para el desarrollo de backend y la generación de código.

Con base en la Figura 4, se afirma que en la base de datos Scopus se encontraron publicaciones delimitadas dentro del periodo de tiempo comprendido entre 2015 y 2023. Se destaca que la mayoría de las publicaciones se concentran en el periodo que va desde 2019 hasta 2023. En la Figura 5, se observa que los países con mayores publicaciones son

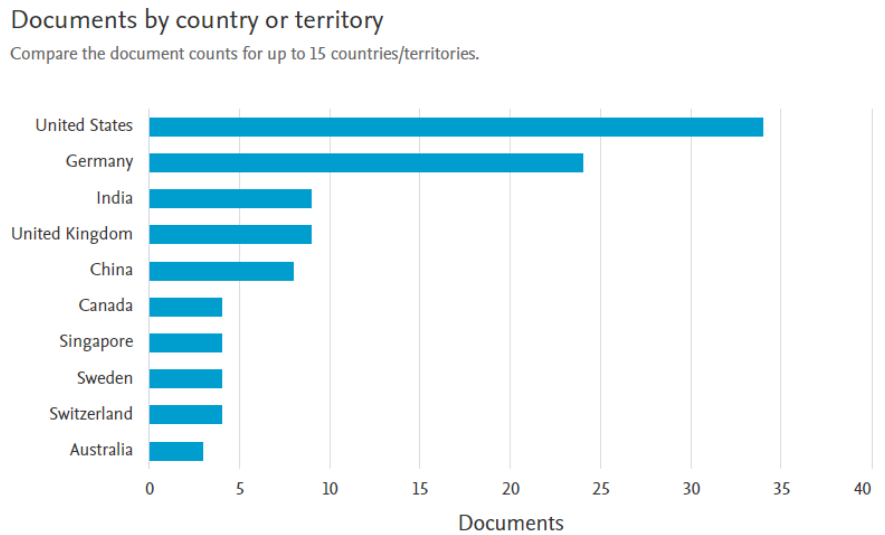
Estados Unidos y Alemania. Esto evidencia que en estos países existe un mayor interés en el tema.

Figura 4. Años de publicación en base de datos Scopus



Fuente: Scopus

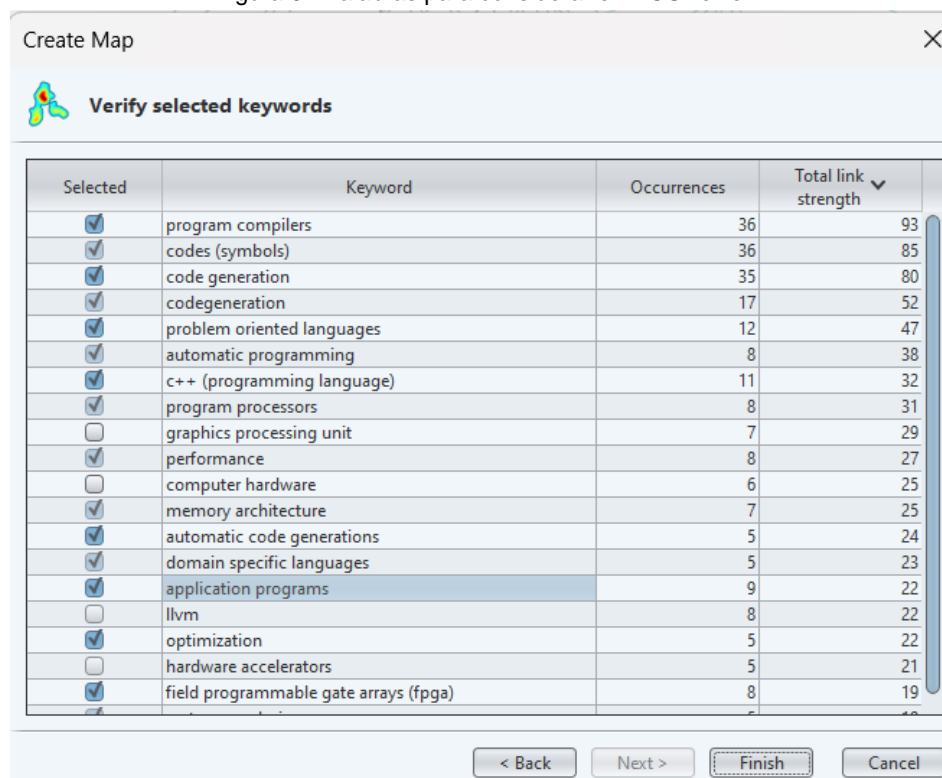
Figura 5. Publicación en países en base de datos Scopus



Fuente: Scopus

Para la recopilación y análisis de las publicaciones recolectadas, se empleó la herramienta de software VOSviewer. VOSviewer se destaca por su diseño especialmente concebido para la visualización y análisis de redes de datos, siendo particularmente útil en los campos de la bibliometría y la cienciometría. Esta herramienta permite a los usuarios adentrarse en la exploración y visualización de las relaciones entre elementos, tales como la co-ocurrencia de términos en documentos científicos, la colaboración entre autores, o la co-citación de referencias [4]. En la construcción de las gráficas, es crucial seleccionar las palabras clave (Figura 6). Estas palabras clave son destacadas ya que permiten la eliminación de documentos que carecen de relevancia para el proyecto.

Figura 6. Palabras para considerar en VOSviewer



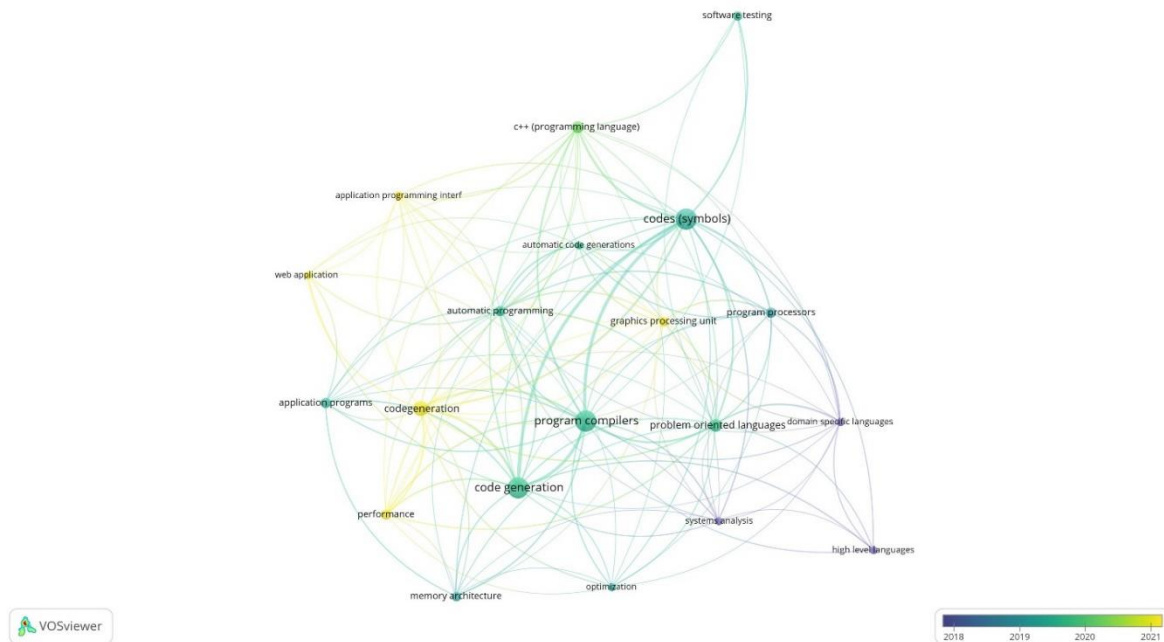
Selected	Keyword	Occurrences	Total link strength
<input checked="" type="checkbox"/>	program compilers	36	93
<input checked="" type="checkbox"/>	codes (symbols)	36	85
<input checked="" type="checkbox"/>	code generation	35	80
<input checked="" type="checkbox"/>	codegeneration	17	52
<input checked="" type="checkbox"/>	problem oriented languages	12	47
<input checked="" type="checkbox"/>	automatic programming	8	38
<input checked="" type="checkbox"/>	c++ (programming language)	11	32
<input checked="" type="checkbox"/>	program processors	8	31
<input type="checkbox"/>	graphics processing unit	7	29
<input checked="" type="checkbox"/>	performance	8	27
<input type="checkbox"/>	computer hardware	6	25
<input checked="" type="checkbox"/>	memory architecture	7	25
<input checked="" type="checkbox"/>	automatic code generations	5	24
<input checked="" type="checkbox"/>	domain specific languages	5	23
<input checked="" type="checkbox"/>	application programs	9	22
<input type="checkbox"/>	llvm	8	22
<input checked="" type="checkbox"/>	optimization	5	22
<input type="checkbox"/>	hardware accelerators	5	21
<input checked="" type="checkbox"/>	field programmable gate arrays (fpga)	8	19

Fuente: VOSviewer

En la Figura 7 se presenta la visualización de la red, donde se identificaron diferentes clústers. En el clúster de aplicaciones web se desprenden ramas de programación automática, generación de código automático, compiladores de programas, interfaz de programación de aplicaciones y generación de código. En el caso del clúster de lenguajes

de alto nivel, se observan ramas como dominio específico de lenguaje, análisis de sistemas, lenguajes orientados a problemas y compiladores de programas. Por último, se identifica el clúster de optimización que incluye ramas importantes como rendimiento, generación de código, dominio de lenguaje específico y programación automática.

Figura 7. Overlay Visualition VOSviewer



Fuente: VOSviewer

## 6. MODELO DE DESARROLLO

El modelo de cascada, también conocido como modelo waterfall, fue la metodología elegida para este proyecto. Este enfoque de gestión de proyectos se caracteriza por su división en fases secuenciales, donde cada etapa comienza una vez finalizada la anterior.

La decisión de adoptar este modelo se fundamentó en una serie de ventajas que lo hacen apropiado para las necesidades específicas del proyecto:

- ✓ **Estructura Secuencial:** Proporciona una organización clara en etapas sucesivas, lo que facilita la planificación y ejecución del proyecto.
- ✓ **Planificación Detallada:** Requiere una planificación exhaustiva desde el inicio, lo que resulta idóneo para proyectos con requisitos estables.
- ✓ **Control de Cambios y Riesgos:** Permite un manejo efectivo de cambios y una gestión proactiva de riesgos a lo largo del proyecto.
- ✓ **Revisión Rigurosa:** Cada fase incorpora actividades de revisión y validación, lo que ayuda a minimizar riesgos y costos asociados a errores.
- ✓ **Adecuado para Requisitos Bien Definidos:** Es especialmente adecuado para proyectos con requisitos claros y estables desde el principio.

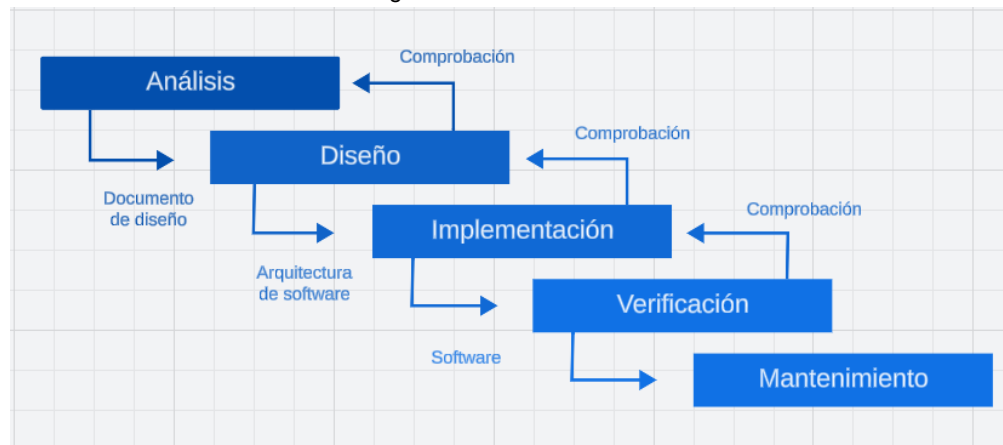
El modelo en cascada de cinco niveles, basado en las propuestas de Winston W. Royce, divide los procesos de desarrollo en las siguientes fases de proyecto: análisis, diseño, implementación, verificación y mantenimiento. La Figura 8, incluye una de las ampliaciones del modelo planteadas por Royce: la verificación de los resultados de cada una de las fases tomando en consideración las exigencias y especificaciones formuladas en el paso anterior.

El proceso de desarrollo de software sigue una serie de etapas clave:

- ✓ **Análisis:** Comienza con un estudio de viabilidad y la definición de requisitos. Se evalúan costos, rentabilidad y factibilidad del proyecto. Luego, se realiza una definición detallada de los requisitos, generando una descripción detallada y un plan de prueba.

- ✓ **Diseño:** Se formula una solución específica basada en los requisitos definidos anteriormente. Los desarrolladores diseñan la arquitectura de software y un plan detallado, centrándose en componentes específicos. Se genera un borrador preliminar con el plan de diseño y los planes de prueba.
- ✓ **Implementación:** La arquitectura de software diseñada se implementa, incluyendo la programación, la depuración y las pruebas unitarias. Los componentes se desarrollan por separado, se prueban y se integran gradualmente en el producto final.
- ✓ **Prueba:** Se integra el software en el entorno seleccionado y se realizan pruebas de aceptación para verificar si cumple con los requisitos definidos. Los productos que superan estas pruebas están listos para su lanzamiento.
- ✓ **Servicio:** Se autoriza la aplicación productiva del software y se pasa a la entrega, mantenimiento y mejora continua del mismo.

Figura 8. Modelo Waterfall



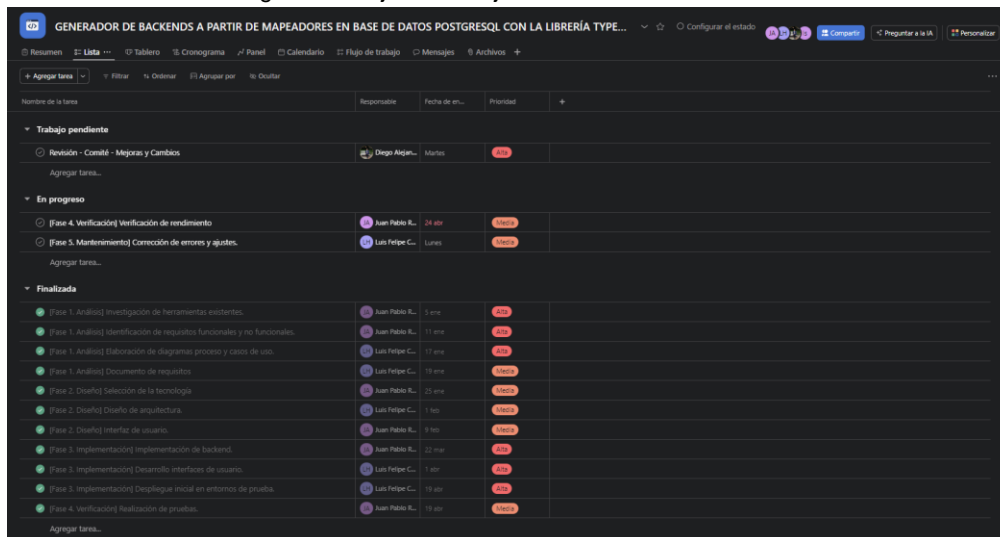
Fuente: Autores

## 7. DESARROLLO DEL PROYECTO

Durante este proceso de desarrollo, cada miembro del equipo colaboró estrechamente utilizando la herramienta Asana y su workflow integrado para automatizar las tareas clave del proyecto. Este enfoque permitió una distribución equitativa de responsabilidades y una asignación eficiente de recursos, garantizando que cada etapa del proyecto avanzara de manera fluida y sin contratiempos. Además, Asana facilitó una coordinación efectiva al permitir la asignación clara de tareas, el establecimiento de fechas límite y la comunicación en tiempo real entre los miembros del equipo.

La Figura 9 proporciona una representación visual de cómo se organizaban y ejecutaban las actividades a lo largo del tiempo, destacando la estructura ordenada y progresiva del flujo de trabajo.

Figura 9. Flujo de trabajo en la herramienta Asana



Nombre de la tarea	Responsable	Fecha de en...	Prioridad
<b>Trabajo pendiente</b>			
Revisión - Comité - Mejoras y Cambios	Diego Rojas...	Martes	Alta
<b>En progreso</b>			
[Fase 4. Verificación] Verificación de rendimiento	Juan Pablo R...	24 Jun	Alta
[Fase 5. Mantenimiento] Corrección de errores y ajustes.	Luis Felipe C...	Lunes	Alta
<b>Finalizada</b>			
[Fase 1. Análisis] Investigación de herramientas existentes.	Juan Pablo R...	5 Jun	Alta
[Fase 1. Análisis] Identificación de requisitos funcionales y no funcionales.	Juan Pablo R...	11 Jun	Alta
[Fase 1. Análisis] Elaboración de diagramas proceso y casos de uso.	Luis Felipe C...	17 Jun	Alta
[Fase 1. Análisis] Documento de requisitos.	Luis Felipe C...	19 Jun	Alta
[Fase 2. Diseño] Selección de la tecnología.	Juan Pablo R...	23 Jun	Alta
[Fase 2. Diseño] Diseño de arquitectura.	Luis Felipe C...	1 Jul	Alta
[Fase 2. Diseño] Interfaz de usuario.	Juan Pablo R...	4 Jul	Alta
[Fase 3. Implementación] Implementación de backend.	Juan Pablo R...	22 Jun	Alta
[Fase 3. Implementación] Desarrollo interfaces de usuario.	Luis Felipe C...	1 Jul	Alta
[Fase 3. Implementación] Despliegue inicial en entornos de pruebas.	Luis Felipe C...	18 Jun	Alta
[Fase 4. Verificación] Realización de pruebas.	Juan Pablo R...	18 Jun	Alta

Fuente: Autores

El desarrollo del proyecto se llevó a cabo siguiendo un enfoque estructurado que se basó en las fases establecidas en el cronograma de actividades propuesto. Esta metodología proporcionó un marco sólido para guiar todas las etapas del proceso, desde la recopilación de datos a través de entrevistas y encuestas hasta la puesta en marcha de la meta

herramienta en un entorno de prueba. Cada fase fue cuidadosamente planificada y ejecutada para garantizar la calidad y efectividad del proyecto en su totalidad.

### 7.1. Fase 1: Análisis

- ✓ Recopilación de datos a través de entrevistas a los directores de los semilleros y encuestas a los estudiantes.
  - Realizar entrevistas a los directores para obtener información sobre las necesidades y expectativas de los semilleros de desarrollo de software.
  - Realizar encuestas a los estudiantes de los semilleros de desarrollo de software para entender sus experiencias, necesidades y preferencias.
  - Analizar las respuestas obtenidas de las entrevistas y encuestas para identificar tendencias, necesidades y desafíos clave.
  
- ✓ Revisión literaria de herramientas para generación de código TypeScript:
  - Realizar un análisis de las herramientas disponibles en el mercado para la generación de código TypeScript en plataformas backend.
  - Evaluar las características, ventajas y desventajas de cada herramienta.
  - Seleccionar las herramientas más adecuadas para tu proyecto en función de los requisitos y objetivos establecidos.
  
- ✓ Identificación y priorización de requisitos:
  - Construcción de requisitos funcionales y no funcionales según su importancia y su impacto en el éxito del proyecto.
  - Documentar los requisitos de manera clara y detallada en un documento que servirá como guía para el desarrollo.
  
- ✓ Elaboración de casos de uso, diagramas de procesos y diagramas de secuencias:
  - Utilizar técnicas como diagramas de procesos y casos de uso para visualizar los procesos clave y las interacciones entre los usuarios y el sistema.
  - Identificar los casos de uso principales y secundarios que reflejen las funcionalidades esperadas de la meta herramienta.

- Integrar los diagramas de procesos y casos de uso en el documento de requisitos para una comprensión completa del sistema.

Tabla 2. Fase 1: Análisis

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> <li>• Bibliotecas virtuales</li> <li>• Documentación de herramientas</li> <li>• Sitios Web</li> <li>• Tutoriales</li> </ul>
Instrumentos y/o herramientas	<ul style="list-style-type: none"> <li>• Espacio de trabajo remoto</li> <li>• Conexión a internet</li> <li>• Equipo de computo                             <ul style="list-style-type: none"> <li>✓ Navegador Web</li> <li>✓ StarUML</li> <li>✓ Formularios Google</li> <li>✓ Teams</li> <li>✓ Bizagi Modeler</li> </ul> </li> </ul>
Salidas	<ul style="list-style-type: none"> <li>• Anexo: Investigación inicial</li> <li>• Anexo: Análisis de entrevista</li> <li>• Anexo: Análisis de encuestas (Estudiantes del semillero de desarrollo de software)</li> <li>• Anexo: Casos de Uso</li> <li>• Anexo: Diagrama de Procesos</li> <li>• Anexo: Diagramas de secuencia</li> <li>• Anexo: Requerimientos</li> </ul>
Tiempo	Semanas/meses

Fuente: Autor

## 7.2. Fase 2: Diseño

- ✓ Selección de tecnología:
  - Evaluar diferentes tecnologías y herramientas disponibles para el desarrollo de la meta herramienta, teniendo en cuenta los requisitos y objetivos del proyecto.
  - Seleccionar la tecnología más adecuada que permita implementar los requerimientos de manera eficiente y efectiva.
  
- ✓ Diseño arquitectónico:
  - Definir la arquitectura del sistema, incluyendo la estructura de la base de datos.
  - Establecer la relación entre los diferentes componentes del sistema y definir cómo interactuarán entre sí.

- Documentar el diseño arquitectónico para facilitar la comprensión y la colaboración entre los miembros del equipo.
- ✓ Definición de la interfaz de usuario y experiencia de usuario (UI/UX):
  - Diseñar la interfaz de usuario para garantizar una experiencia de usuario intuitiva y atractiva.
  - Definir la navegación y la disposición de los elementos en la interfaz de usuario.
  - Establecer la lógica de negocio necesaria para la generación de código TypeScript y su integración con la interfaz de usuario.

Tabla 3. Fase 2: Diseño

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> <li>• Documento detallado de requisitos</li> <li>• Diagramas de procesos</li> <li>• Herramientas de selección de tecnología</li> </ul>
Instrumentos y/o herramientas	<ul style="list-style-type: none"> <li>• Espacio de trabajo remoto</li> <li>• Conexión a internet</li> <li>• Computador                             <ul style="list-style-type: none"> <li>✓ Navegador Web</li> <li>✓ Draw.io</li> <li>✓ Figma</li> <li>✓ Balsamiq</li> <li>✓ Lucidchart</li> </ul> </li> </ul>
Salidas	<ul style="list-style-type: none"> <li>• Anexo: Selección de tecnología</li> <li>• Anexo: Arquitectura del sistema</li> <li>• Anexo: Visualización de la interfaz de Usuario (Mockups)</li> </ul>
Tiempo	Semanas/meses

Fuente: Autor

### 7.3. Fase 3: Implementación

- ✓ Implementación del backend:
  - Desarrollar el backend de la meta herramienta, centrándose en el mapeo de metadatos de tablas en bases de datos relacionales.
  - Implementar las funcionalidades adicionales para la generación de código TypeScript y la automatización de procesos CRUD.
- ✓ Desarrollo de interfaces de usuario:

- Desarrollar las interfaces de usuario y la interacción con el sistema backend para la generación y visualización de código TypeScript.
  - Garantizar que la interfaz de usuario sea coherente con el diseño establecido en la fase de diseño.
- ✓ Despliegue inicial en entornos de prueba:
- Desplegar inicialmente la meta herramienta en entornos de prueba para verificar su funcionamiento y asegurarse de que todos los sistemas estén funcionando correctamente.

Tabla 4. Fase 3: Implementación

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> <li>• Diseño arquitectónico de la fase de diseño</li> <li>• Documento detallado de requisitos</li> <li>• Desarrollo del backend y frontend</li> </ul>
Instrumentos y/o herramientas	<ul style="list-style-type: none"> <li>• Espacio de trabajo remoto</li> <li>• Conexión a internet</li> <li>• Computador                             <ul style="list-style-type: none"> <li>✓ Navegador Web</li> <li>✓ Visual Studio Code</li> <li>✓ Postman</li> <li>✓ Insonmnia</li> <li>✓ Git</li> <li>✓ Github</li> <li>✓ PgAdmin</li> </ul> </li> </ul>
Salidas	<ul style="list-style-type: none"> <li>• Anexo: Diagrama de base de datos</li> <li>• Anexo: Desarrollo Backend</li> <li>• Anexo: Desarrollo Frontend</li> </ul>
Tiempo	Semanas/meses

Fuente: Autor

#### 7.4. Fase 4: Verificación

- ✓ Realización de pruebas:
- Realizar pruebas exhaustivas, incluyendo pruebas unitarias y pruebas de usuario para garantizar la calidad y el funcionamiento adecuado del sistema.
  - Verificar la eficiencia y el rendimiento del código en entornos de prueba para identificar posibles cuellos de botella o problemas de rendimiento.
- ✓ Creación de manual de usuario:

- Desarrollar un anexo que contenga el manual del usuario, proporcionando instrucciones detalladas, ejemplos y capturas de pantalla para guiar a los usuarios en el uso de la meta herramienta.

Tabla 5. Fase 4: Verificación

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> <li>• Implementación de la fase de implementación</li> <li>• Documento de requisitos</li> <li>• Plan de pruebas</li> </ul>
Instrumentos y/o herramientas	<ul style="list-style-type: none"> <li>• Espacio de trabajo remoto</li> <li>• Conexión a internet</li> <li>• Computador                             <ul style="list-style-type: none"> <li>✓ Navegador Web</li> <li>✓ Jest</li> <li>✓ PostgreSQL</li> <li>✓ Lighthouse</li> <li>✓ SEOquake Analytics</li> </ul> </li> </ul>
Salidas	<ul style="list-style-type: none"> <li>• Anexo: Pruebas Unitarias Back-end</li> <li>• Anexo: Pruebas de accesibilidad y mejores prácticas del Frontend</li> <li>• Anexo: Pruebas de experiencia de usuario</li> <li>• Anexo: Plan de pruebas Base de Datos (“Colegio”)</li> <li>• Anexo: Plan de pruebas Base de Datos (“Empresa”)</li> <li>• Anexo: Plan de pruebas Base de Datos (“Tienda”)</li> <li>• Anexo: Manual de usuario</li> </ul>
Tiempo	Semanas/meses

Fuente: Autor

## 7.5. Fase 5: Mantenimiento

- ✓ Corrección de errores y ajustes:
  - Corregir errores y realizar ajustes basados en los resultados de las pruebas realizadas durante la fase de verificación.
  - Atender cualquier problema o incidencia que surja en el sistema y garantizar su correcto funcionamiento

Tabla 6. Fase 5: Mantenimiento

Tipo	Elementos
Entradas	<ul style="list-style-type: none"><li>• Resultados de pruebas de la fase de verificación</li><li>• Incidencias y problemas identificados</li><li>• Retroalimentación de los usuarios</li></ul>
Instrumentos y/o herramientas	<ul style="list-style-type: none"><li>• Espacio de trabajo remoto</li><li>• Conexión a internet</li><li>• Computador<ul style="list-style-type: none"><li>✓ Github</li><li>✓ GitLens</li></ul></li></ul>
Salidas	<ul style="list-style-type: none"><li>• Anexo: Mantenimiento y correcciones de la meta herramienta</li></ul>
Tiempo	Semanas/meses

Fuente: Autor

## 8. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

- La meta herramienta automatiza la generación de código TypeScript para operaciones CRUD con TypeORM simplificando y acelerando el proceso manual. Esto mejora significativamente el flujo de trabajo de los estudiantes en el desarrollo de aplicaciones web, permitiéndoles concentrarse en aspectos más críticos de sus proyectos.
- El uso de tecnologías como TypeORM y TypeScript en el proyecto fomenta la adopción de metodologías y herramientas en desarrollo, preparando a los estudiantes para el mercado laboral y futuro.
- La retroalimentación y colaboración activa de los docentes y estudiantes de los semilleros fueron cruciales para asegurar que la herramienta fuera funcional a las necesidades específicas de la comunidad académica y tecnológica de la Universidad Santo Tomás.
- El plan de pruebas permitió evaluar el código generado en diferentes contextos de bases de datos, asegurando su correcto funcionamiento y compatibilidad con diversas estructuras de datos. Esta estrategia robusta y exhaustiva ha fortalecido la calidad del código generado y su adaptabilidad a distintas configuraciones, lo que confirma la efectividad de la herramienta en escenarios variados.
- GitHub permitió la colaboración efectiva entre los miembros del equipo, facilitando el acceso compartido al código fuente y a la documentación relacionada. El uso de ramas (branches) para gestionar diferentes tareas y contextos de pruebas promovió un desarrollo estructurado y paralelo, acelerando el progreso de manera eficiente.
- La implementación de la metodología en cascada en el proyecto proporcionó un enfoque claro y secuencial para el desarrollo, permitiendo una planificación detallada y una ejecución organizada de cada fase.
- La implementación del módulo de autenticación multiplataforma permitió un acceso seguro y eficiente a la meta herramienta para conectarse a los modelos relacionales en PostgreSQL.
- Los resultados de la encuesta revelan que la mayoría de los desarrolladores dedican entre 1 y 3 horas a la construcción de componentes CRUD en proyectos de software, lo que indica una práctica común pero que puede ser laboriosa. El desarrollo de este proyecto, demuestra completarla en solo 4 segundos durante las pruebas, destacando

el potencial de la herramienta para mejorar la eficiencia y la productividad en el desarrollo de software.

## 9. REFERENCIAS

- [1] P. Malo-Perisé and J. Merseguer, "The 'Socialized Architecture': A Software Engineering Approach for a New Cloud," *Sustainability*, vol. 14, no. 4, p. 2020, Feb. 2022, doi: 10.3390/su14042020.
- [2] E. Maida and J. Pacienza, "Metodologías de desarrollo de software," *Bibl. Digit. la Univ. Católica Argentina*, p. 117, 2018, [Online]. Available: <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>
- [3] A. Marin Diaz, Y. Trujillo Casañola, and D. Buedo Hidalgo, "Apuntes para gestionar actividades de calidad en proyectos de desarrollo de software para disminuir los costos de corrección de defectos," *Ingeniare. Rev. Chil. Ing.*, vol. 27, no. 2, pp. 319–327, 2019, doi: 10.4067/s0718-33052019000200319.
- [4] N. J. van Eck and L. Waltman, "Software survey: VOSviewer, a computer program for bibliometric mapping," *Scientometrics*, vol. 84, no. 2, pp. 523–538, 2010, doi: 10.1007/s11192-009-0146-3.
- [5] S. Miguel, "P u c p e p," 2020.
- [6] TreeView Webgen, "Softonic," *TreeView Webgen*, 2001. <https://treeview-webgen.softonic.com/>
- [7] "Ruby on Rails." <https://rubyonrails.org/>
- [8] "Django." <https://www.djangoproject.com/>
- [9] "LaRavel - the PHP framework for web artisans." <https://laravel.com/docs/11.x>
- [10] "Spring boot." <https://spring.io/projects/spring-boot>
- [11] "Express - Node.js web application framework." <https://expressjs.com/>
- [12] "Sequelize." <https://sequelize.org/>
- [13] "Información general de ASP.NET Core," 2023. <https://learn.microsoft.com/es-es/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0>
- [14] "Documentation | NestJS - A progressive Node.js framework." <https://docs.nestjs.com/>
- [15] "Backend Generator by Duomly - Build backend in minutes without coding," 2020. <https://www.backendgenerator.com/>
- [16] "Auto backend." <https://www.autobackend.dev>
- [17] "Turn your team into a customer-obsessed, NRR driving machine | BackEngine: The

- Product for Customer-Obsessed Teams.” <https://www.backengine.com/>
- [18] “Dynaboard – Collaborative low-code IDE for developers.” <https://dynaboard.com>
- [19] E. A. Meyer, “Casos de uso,” p. 2014, 2004.
- [20] Sitio Bizagi, “Sitio de Bizagi,” 2022. <https://www.bizagi.com/es/plataforma/pruebe-modeler>
- [21] “Diagrama de procesos,” no. 16, pp. 4–6, 2024.
- [22] “Diagrama de secuencia (Administrador)”.
- [23] “Diagrama de secuencia (Usuario)”.
- [24] “Introduction to node.js,” 2024. <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- [25] “Node.js with TypeScript,” 2024. <https://nodejs.org/en/learn/getting-started/nodejs-with-typescript>
- [26] “About,” 2024. <https://www.postgresql.org/about/>
- [27] “Angular,” 2024. <https://angular.io/guide/what-is-angular>
- [28] “Jest.” <https://jestjs.io>
- [29] “MixPanel Marketing Analytics - MixPanel,” 2024. [https://mixpanel.com/m/mixpanel-marketing-analytics?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=LATAM-Non-Brand-Search-Marketing-Analytics-EN-Phrase-Desktop&utm\\_content=Platform-Phrase&utm\\_ad=676745289274&utm\\_term=marketing\\_analytics\\_platform&matchtype=](https://mixpanel.com/m/mixpanel-marketing-analytics?utm_source=google&utm_medium=cpc&utm_campaign=LATAM-Non-Brand-Search-Marketing-Analytics-EN-Phrase-Desktop&utm_content=Platform-Phrase&utm_ad=676745289274&utm_term=marketing_analytics_platform&matchtype=)
- [30] “Google Lighthouse: Qué es y Cómo Utilizarlo,” 2023. [https://es.semrush.com/blog/como-utilizar-google-lighthouse/?kw=&cmp=LM\\_SRCH\\_DSA\\_Blog\\_ES&label=dsa\\_pagefeed&Network=g&Device=c&utm\\_content=678247163034&kwid=dsa-2232567167021&cmpid=19249322807&agpid=152775994777&BU=Core&extid=109486340859&adpos=&gad\\_sourc](https://es.semrush.com/blog/como-utilizar-google-lighthouse/?kw=&cmp=LM_SRCH_DSA_Blog_ES&label=dsa_pagefeed&Network=g&Device=c&utm_content=678247163034&kwid=dsa-2232567167021&cmpid=19249322807&agpid=152775994777&BU=Core&extid=109486340859&adpos=&gad_sourc)
- [31] Connor (C.) Lahey, “Conceptos básicos de SEO: cómo hacer SEO para principiantes.” [https://es.semrush.com/blog/seo-basico/?kw=&cmp=LM\\_SRCH\\_DSA\\_Blog\\_ES&label=dsa\\_pagefeed&Network=g&Device=c&utm\\_content=678247163691&kwid=dsa-2232567167701&cmpid=19249322807&agpid=152775995697&BU=Core&extid=123532550104&adpos=&gad\\_source=1&gclid=CjwKCAjwrli](https://es.semrush.com/blog/seo-basico/?kw=&cmp=LM_SRCH_DSA_Blog_ES&label=dsa_pagefeed&Network=g&Device=c&utm_content=678247163691&kwid=dsa-2232567167701&cmpid=19249322807&agpid=152775995697&BU=Core&extid=123532550104&adpos=&gad_source=1&gclid=CjwKCAjwrli)

## 10. ANEXOS

### 10.1. Anexo: Análisis de entrevista (Director del semillero de desarrollo de software)

A continuación, se presenta la transcripción de la entrevista con el ingeniero Sergio Arley Puerto, director del semillero de investigación de desarrollo de software de la Universidad Santo Tomás. Actualmente, la universidad cuenta con dos semilleros: uno en desarrollo de software “Veritate Software” y otro en videojuegos “Videojuegos y metaverso”. Sin embargo, para el desarrollo de la encuesta y la definición de los requerimientos funcionales y no funcionales, solo se considerará el semillero de desarrollo de software “Veritate Software”. Por esta razón, el semillero de videojuegos no fue incluido.

#### Transcripción

##### ED Equipo de desarrollo

Ingeniero Buenos días.

El grupo de trabajo está conformado por Luis Felipe Chaparro Hurtado y quien le habla Equipo de desarrollo. Esta entrevista es con el propósito, pues, de dar a conocer nuestro trabajo de grado, que tiene como título de proyecto “Generador de Back-end a partir de mapeadores de base de datos PostgreSQL con la librería TypeORM” Entonces, pues deseamos realizarle unas preguntas para conocer su concepto sobre el tema.

##### S Sergio Arley Puerto Moreno

Bueno, listo.

##### ED Equipo de desarrollo

Listo sí señor. Entonces la primera sería en el ámbito del desarrollo de aplicaciones web. ¿Considera usted importante contar con una herramienta que genere código backend para operaciones de persistencia CRUD?

**S Sergio Arley Puerto Moreno**

Sí, claro.

Sería un valor agregado que nos puede funcionar o nos puede facilitar la tarea. Ahora hay que pensar si, cuál va a ser el tipo de archivo de persistencia, pero si claro es necesario y sumaría al proceso del desarrollo backend.

**ED Equipo de desarrollo**

La segunda sería, ¿cómo cree que una herramienta de generación de código backend para CRUD podría impactar en la productividad y la eficiencia en los desarrolladores web?

**S Sergio Arley Puerto Moreno**

Bueno, puede impactar tomándolo como una buena práctica de desarrollo. Sí, que sería pues, complementario al trabajo precisamente del desarrollador full ¿Pero porque Full-stack?

Sí, porque a veces el desarrollo Full-stack enfoca su interés en uno de los dos entornos. El Full-stack no es una panacea, es aquella personita pues que le tocó hacerse cargo de las de las dos partes, pero puede tener su fuerte más denotado tal vez en el Front-end sí puede llegar a ser un escenario. Entonces de esa manera, puede llegar a considerarse como una buena práctica el tener un generador de backend, por supuesto, cuando esté reúne esfuerzos más tal vez en el Front-end permite, pues, que una herramienta le sea complementaria para trabajar sobre el backend.

**ED Equipo de desarrollo**

Bueno, Ingeniero la siguiente, usted, pues como docente, ¿considera usted que una herramienta de generación de Backend sería beneficiosa en el desarrollo de aplicaciones web para permitir que los estudiantes se concentren más en el diseño y desarrollo del Front-end?

**S Sergio Arley Puerto Moreno**

Pero, por supuesto, les va a resultar de suficiente impacto. Cuando en este caso, pues se les está indicando cuál va a ser el camino al que se va a orientar si usted va a decir bueno, lo vamos a trabajar con auto, vamos a trabajar con typescript o pudiese llegar en un momento determinado escalable a otros lenguajes donde ustedes digan, no venga, llevémoslo a Python un ejemplo.

También podríamos optar por trabajar con Spring. Aunque esté enfocado en un solo lenguaje, especialmente en el caso del Back-end, resultará de gran interés, complementario y orientador para los estudiantes, ayudándoles a trazar su ruta de aprendizaje.

**ED Equipo de desarrollo**

Excelente, la siguiente pregunta es ¿Cree usted que sería beneficioso implementar un módulo de autenticación multiplataforma en la aplicación para acceder al generador de códigos y su conexión a base de datos?

Pues para acceder a este generador de código.

**S Sergio Arley Puerto Moreno**

Nos resulta bastante útil y sí, pues que ustedes tengan ese control de acceso, por supuesto. Es totalmente integrable tanto como en seguridad y en alcance. Si es lo que de pronto ustedes le pueden apuntar para que tenga una buena experiencia de usuario a quienes vayan a hacer parte de este generador de Back-end estupendo.

**ED Equipo de desarrollo**

Listo ingeniero.

Por último, pues sería, ¿Qué funcionalidad o característica considera útil para una aplicación de generador de código Backend?

### **S** Sergio Arley Puerto Moreno

Listo esa pregunta, claro, permite cerrar bien el interrogante. Yo le agregaría una funcionalidad que sea como un paso a paso, cuando yo ya ingresé a este generador de Back-end, como hace un paso a paso y listo.

Pues el ideal es que me permita tener como una orientación, no sé si han fijado ustedes en diferentes aplicaciones, que es lo que hacen es generar un evento sobre la aplicación y ese evento le va diciendo que quiere ver el asistente, cómo va a funcionar el aplicativo de este generador de Back-end de acá. En su primera experiencia en cuanto a usabilidad, pues sea totalmente acertada y tenga una buena acogida por los usuarios, porque va a generar bastante expectativa cuando esto ya se sea un hecho.

Cuando ya esté, pues no digamos en producción, esté acá al menos un prototipo o en piloto le va a resultar interesante.

### **ED** Equipo de desarrollo

¿Listo?

Pues, por último, no sé si desea agregar algún comentario sobre, pues la idea y pues la aplicación que pensamos desarrollar.

### **S** Sergio Arley Puerto Moreno

¿Ah, listo, perfecto y no, pues a qué iba?

No, no, no. Con todo el cariño y con todo el respeto que cuando ya sea un hecho y esté, pues en un alcance tangible, si me quiere invitar para revisarla, bienvenido sea. Sí, eso es lo único que yo le sumaría, porque pues sé que le están metiendo ustedes acá

corazón al tema con todo su conocimiento, entonces si gustan más adelante me invitan para ir viendo cómo va, cómo va quedando la misma.

**ED** Equipo de desarrollo

Claro que si ingeniero. Su opinión sería gratificante.

Y, por último, pues agradecerle por su tiempo y su participación en esta entrevista.

**S** Sergio Arley Puerto Moreno

Tranquilos con el mayor gusto, como les dije y le reitero, digamos que es un tema. El tema del desarrollo, pues a mí me apasiona bastante, sí, y pues es interesante que un proyecto marque hito, que estoy diciendo yo a todos los estudiantes de que acá es dejar una huella, porque ustedes más adelante, después de esto, van a estudiar un posgrado, van a estudiar una maestría, se les va a pedir que tengan una experiencia en investigación, que este proyecto no solamente quede en el CRAI, ni que vaya a quedar en un repositorio local donde nadie lo va a visibilizar.

**ED** Equipo de desarrollo

Si señor correcto.

Muchísimas gracias ingeniero, por su tiempo y sus comentarios.

**S** Sergio Arley Puerto Moreno

Con todo gusto. Hasta luego buen día.

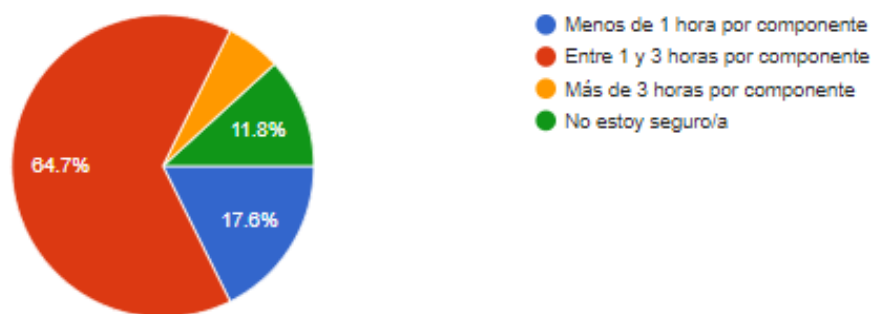
## 10.2. Anexo: Análisis de encuestas (Estudiantes del semillero de desarrollo de software)

Este anexo presenta un análisis detallado de las encuestas realizadas a los estudiantes del semillero de desarrollo de software de la Universidad Santo Tomás. El propósito de estas encuestas fue recopilar información sobre las experiencias y opiniones de los estudiantes en relación con la creación del proyecto "Generador de Backend a partir de Mapeadores de Bases de Datos PostgreSQL con la Biblioteca TypeORM".

1. ¿Cuánto tiempo sueles dedicar a la construcción de componentes CRUD en tus proyectos de desarrollo de software?

La Figura 10 muestra que la mayoría de las personas (64.7%) dedica entre 1 y 3 horas a la construcción de componentes CRUD en proyectos de desarrollo de software, siendo esta la práctica más común. Un 17.6% dedica menos de 1 hora, lo que sugiere simplicidad o experiencia previa. Un 11.8% no está seguro del tiempo dedicado, indicando falta de experiencia en la actividad. Y el 6% invierte más de 3 horas, indicando la posible complejidad de la actividad.

Figura 10. ¿Cuánto tiempo sueles dedicar a la construcción de componentes CRUD en tus proyectos de desarrollo de software?

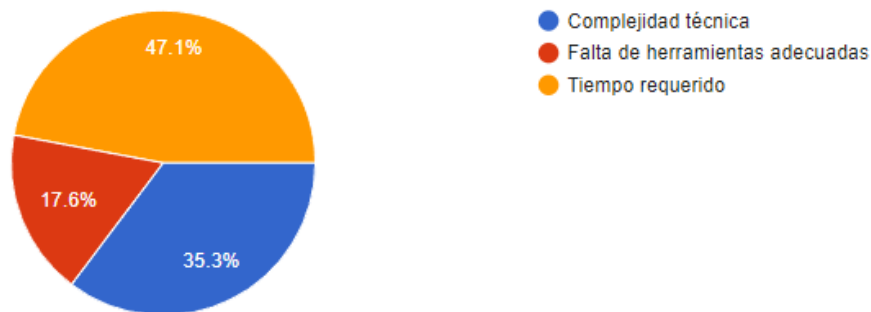


Fuente: Autores

2. ¿Qué dificultades has experimentado al trabajar con la creación del backend para bases de datos relacionales en proyectos anteriores?

La Figura 11 muestra que la principal dificultad en la creación de un backend para bases de datos relacionales es el tiempo requerido (47.1%), lo que sugiere la necesidad de una planificación y gestión cuidadosa. La segunda dificultad más común es la complejidad técnica (35.3%), lo que indica la necesidad de una comprensión profunda de conceptos de bases de datos y desarrollo de software. La tercera dificultad es la falta de herramientas adecuadas (17.6%).

Figura 11. ¿Qué dificultades has experimentado al trabajar con la creación del backend para bases de datos relacionales en proyectos anteriores?

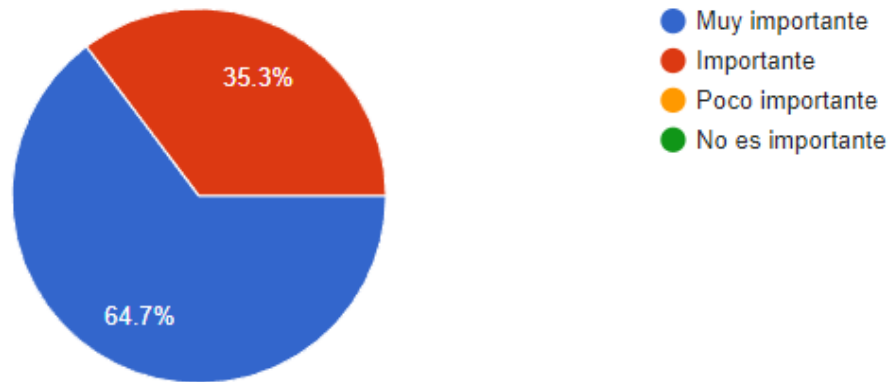


Fuente: Autores

3. ¿Qué importancia le das al uso del patrón Modelo-Vista-Controlador (MVC) en los desarrollos Backend?

La Figura 12 muestra que casi todas las personas (99,9%) consideran importante o muy importante el uso del patrón Modelo-Vista-Controlador (MVC) en desarrollos Back-end. Esto indica que el patrón MVC es ampliamente utilizado y valorado como una práctica beneficiosa en el desarrollo de software.

Figura 12. ¿Qué importancia le das al uso del patrón Modelo-Vista-Controlador (MVC) en los desarrollos Backend?

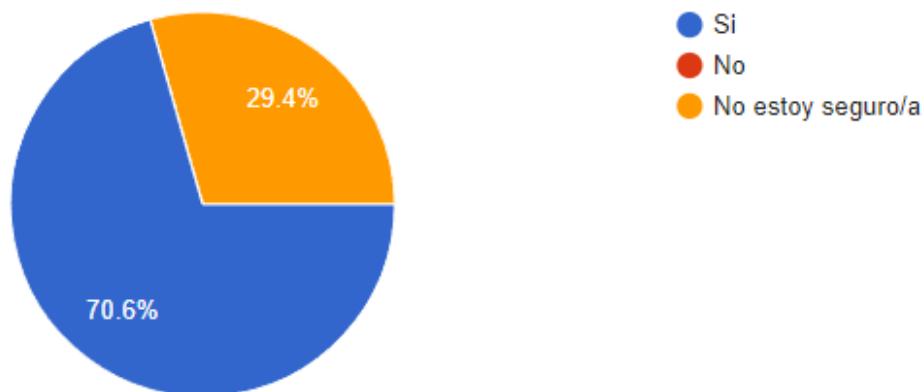


Fuente: Autores

4. ¿Consideras que la generación automática de código TypeScript para operaciones CRUD en el backend podría beneficiar la eficiencia de tu trabajo?

La Figura 13 muestra que la mayoría de las personas (70,6%) si creen que la generación de código TypeScript para operaciones CRUD en el Back-end podría beneficiar la eficiencia de su trabajo, mientras que solo el (29,4%) cree que no estaria seguro si este seria beneficioso.

Figura 13. ¿Consideras que la generación automática de código TypeScript para operaciones CRUD en el backend podría beneficiar la eficiencia de tu trabajo?

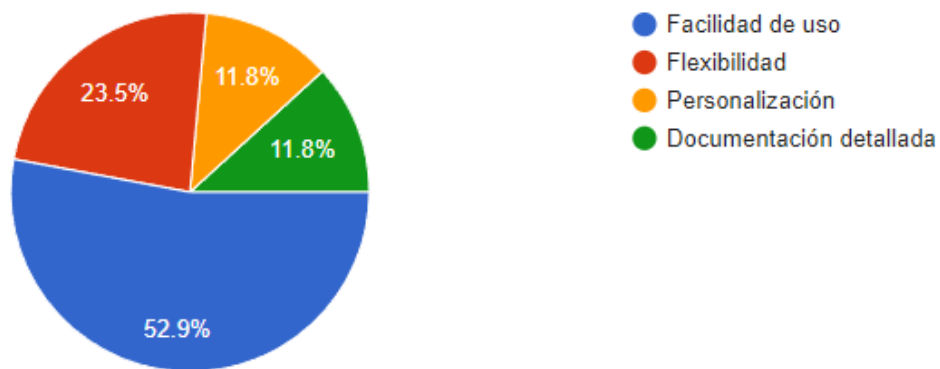


Fuente: Autores

5. ¿Qué características consideras esenciales en una herramienta para la generación de código TypeScript en plataformas Backend?

La Figura 14 muestra que la característica más importante que los estudiantes consideran esencial en una herramienta para la generación de código en plataformas backend es la facilidad de uso (52.9%), ya que valoran herramientas fáciles de aprender y usar para ser productivos rápidamente. Además, consideran importantes la flexibilidad (23.5%) para satisfacer diversas necesidades, la personalización (11.8%) para adaptarse a su flujo de trabajo, y la documentación detallada (11.8%) para usar la herramienta de manera efectiva.

Figura 14. ¿Qué características consideras esenciales en una herramienta para la generación de código TypeScript en plataformas Backend?

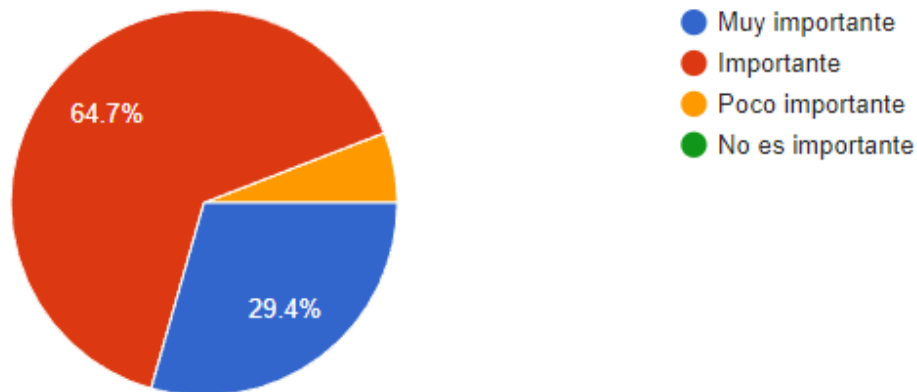


Fuente: Autores

6. ¿Qué tan importante crees que es la generación automática de código para reducir los tiempos de desarrollo en proyectos de software?

La Figura 15 muestra la mayoría de los estudiantes (94,1%) creen que la generación automática de código es muy importante (64,7%) o importante (29,4%) para reducir los tiempos de desarrollo en proyectos de software. Esto sugiere que esta práctica es ampliamente utilizada y considerada como una forma efectiva de mejorar la eficiencia del desarrollo.

Figura 15. ¿Qué tan importante crees que es la generación automática de código para reducir los tiempos de desarrollo en proyectos de software?



Fuente: Autores

7. ¿Te gustaría tener acceso a una herramienta que genere automáticamente el código backend en TypeScript, incluyendo operaciones CRUD, a partir del diseño de la base de datos?

La Figura 16 muestra que el 100% de los estudiantes está a favor de una herramienta que genere automáticamente el código backend en TypeScript, incluyendo operaciones CRUD, a partir del diseño de la base de datos.

Figura 16. ¿Te gustaría tener acceso a una herramienta que genere automáticamente el código backend en TypeScript, incluyendo operaciones CRUD, a partir del diseño de la base de datos?

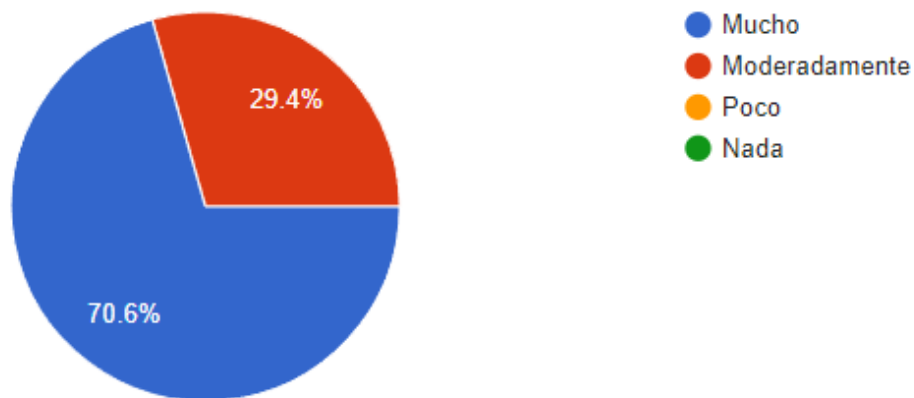


Fuente: Autores

8. ¿Cuánto valorarías una herramienta que automatice el proceso de generación de código TypeScript en el backend, reduciendo la probabilidad de errores y acelerando el desarrollo?

La Figura 17 muestra la mayoría de los estudiantes (70,6%) valorarían mucho la meta herramienta, ya que esta reduciría la probabilidad de errores y aceleraría el desarrollo, mientras que (29,4%) lo valorarían moderadamente.

Figura 17. ¿Cuánto valorarías una herramienta que automatice el proceso de generación de código TypeScript en el backend, reduciendo la probabilidad de errores y acelerando el desarrollo?

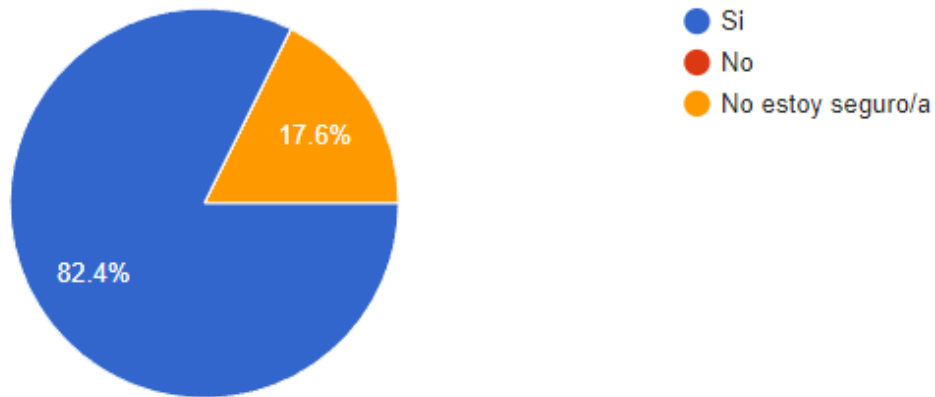


Fuente: Autores

9. ¿Estarías dispuesto/a a probar y proporcionar retroalimentación sobre una herramienta de generación de código TypeScript en el backend que cumpla con los objetivos mencionados?

La Figura 18 muestra que la mayoría de los estudiantes (82,4%) están dispuestos a probar y proporcionar retroalimentación sobre la meta herramienta. Esto es una señal de que existe interés en este tipo de herramienta.

Figura 18. ¿Estarías dispuesto/a a probar y proporcionar retroalimentación sobre una herramienta de generación de código TypeScript en el backend que cumpla con los objetivos mencionados?

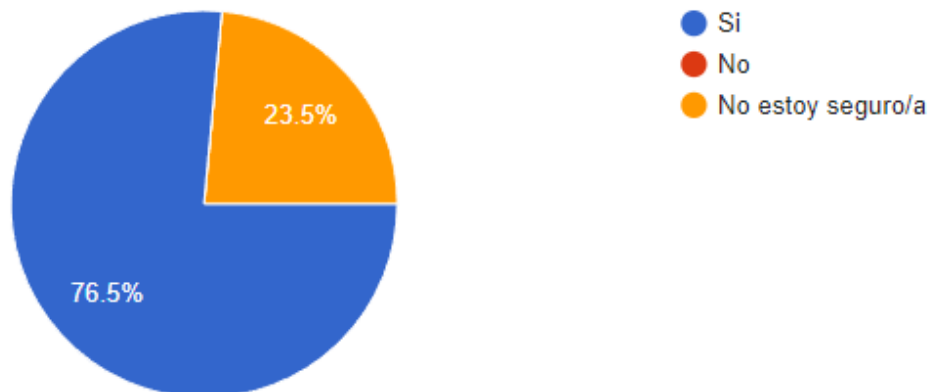


Fuente: Autores

10. ¿Te gustaría que el código del backend se pueda descargar para simplemente ponerlo en funcionamiento en un entorno de prueba y evaluar su desempeño?

La Figura 19 muestra (76,5%) de los estudiantes están de acuerdo con que la meta herramienta permita descargar e implementar el código backend para ejecutarlo en un entorno de prueba, lo que indica un fuerte interes de los estudiantes.

Figura 19. ¿Te gustaría que el código del backend se pueda descargar para simplemente ponerlo en funcionamiento en un entorno de prueba y evaluar su desempeño?



Fuente: Autores

### 10.3. Anexo: Investigación inicial

El Anexo de Investigación Inicial constituye una recopilación exhaustiva de herramientas, frameworks y plataformas destinadas al desarrollo de aplicaciones web y móviles, así como generadores de código. Estas herramientas están meticulosamente diseñadas para agilizar y simplificar el proceso de desarrollo de software, ofreciendo funcionalidades esenciales como la generación automática de código fuente, la creación de operaciones CRUD, la implementación de arquitecturas específicas y la integración fluida con diversas tecnologías.

Entre las herramientas destacadas se encuentran:

- ✓ **Travel Guide Application:** Genera código para la implementación de aplicaciones web portátiles utiliza patrón de Arquitectura MVVM a partir del diseño de un diagrama de transición de estado.[5]
- ✓ **WebGen:** Genera menús tipo árbol automáticamente desde el front-end hasta el back-end. Esta herramienta genera menús en formato de árbol en código HTML, incluyendo enlaces necesarios y organizando el contenido web en carpetas de manera eficiente. [6]
- ✓ **Titan Architect Mobile:** Genera código para el desarrollo de aplicaciones web y móviles. Sigue el patrón REST.
- ✓ **Visual Paradigm:** Generador de código fuente para diferentes lenguajes de programación. Se integra con IDE como NetBeans, Eclipse, Laika.
- ✓ **JustBusiness:** Desarrollo de aplicaciones Android utilizando patrones de arquitectura Naked Objects, Navegación y operaciones CRUD.
- ✓ **OpenCLGen:** Es un servicio web, es legible y puede integrarse en una aplicación existente o ejecutarse de forma independiente.

- ✓ **CodaGen:** Permite a las empresas generar automáticamente código fuente a partir de modelos UML, basado en MDA y patrón multicapa.
- ✓ **Lyo code generator:** Utiliza un estándar OASIS OSLC (Open Services for Lifecycle Collaboration) para la interoperabilidad de las herramientas de software. Sigue el patrón arquitectónico REST.
- ✓ **JEEWiz:** Genera sistemas de aplicaciones web casi completos a partir de modelos XML. Utiliza MDA y patrón multicapa.
- ✓ **CreaCod:** Generador de Código Fuente para controlar base de datos MySQL, SQL Server y Access en lenguajes JAVA, PHP y ASP. [5]
- ✓ **Ruby on Rails:** es un framework de desarrollo web que facilita la creación de aplicaciones web mediante el paradigma de convención sobre configuración. Rails incluye generadores de código que pueden crear fácilmente operaciones CRUD para modelos. [7]
- ✓ **Django:** Django es un framework de desarrollo web en Python que sigue el principio DRY (Don't Repeat Yourself). Django proporciona un ORM (Object-Relational Mapping) que simplifica el acceso a la base de datos y facilita la creación de operaciones CRUD.[8]
- ✓ **Laravel:** Laravel es un framework de desarrollo web en PHP que ofrece una sintaxis elegante y expresiva. Laravel incluye un generador de código llamado Artisan que puede ser utilizado para crear rápidamente controladores, modelos y migraciones para operaciones CRUD.[9]
- ✓ **Spring Boot:** Spring Boot es un framework de desarrollo de aplicaciones Java que facilita la creación de aplicaciones robustas y escalables. Spring Boot incluye Spring Data JPA, que proporciona herramientas para crear fácilmente operaciones CRUD sobre bases de datos relacionales.[10]

- ✓ **Express.js:** Es un framework de desarrollo web para Node.js que es minimalista y flexible. Aunque no incluye generadores de código integrados, existen herramientas como Sequelize para ORM y otros paquetes que pueden facilitar la creación de operaciones CRUD.[11] [12]
  
- ✓ **ASP.NET Core:** ASP.NET Core es un framework de desarrollo web de código abierto desarrollado por Microsoft. Ofrece características como el scaffolding, que permite generar automáticamente código para operaciones CRUD basado en modelos de datos.[13]
  
- ✓ **NestJS:** NestJS es un framework de Node.js que se basa en TypeScript y está inspirado en Angular. NestJS tiene integración directa con TypeORM y proporciona herramientas para generar automáticamente controladores y servicios CRUD utilizando los decoradores proporcionados por NestJS.[14]
  
- ✓ **Duomly:** Es una herramienta que permite a los usuarios crear un backend y una solución API completa para aplicaciones web o móviles en pocos pasos y sin necesidad de codificar. Genera código funcional de NodeJS y TypeScript con puntos finales de API REST personalizables y vinculados a bases de datos.[15]
  
- ✓ **Autobackend:** Es una herramienta impulsada por IA que permite crear backends para aplicaciones web y móviles sin necesidad de escribir código. Simplemente se describe lo que necesitas en un par de frases y Autobackend proporcionará la estructura del backend.[16]
  
- ✓ **Backengine:** Es una plataforma de desarrollo sin código que permite crear backends para aplicaciones web y móviles sin necesidad de escribir código. Permite crear APIs, bases de datos, lógica de negocio y flujos de trabajo.[17]
  
- ✓ **Dynaboard:** Es una plataforma de desarrollo web sin código que permite crear aplicaciones web completas sin necesidad de escribir código. Par crear interfaces de usuario, definir la lógica de negocio.[18]

Estas herramientas, entre otras, cubren una amplia gama de necesidades en el desarrollo de software, desde la creación de interfaces de usuario hasta la implementación de complejas operaciones de backend, lo que las convierte en recursos valiosos para cualquier equipo de desarrollo.

Asimismo, las mejores prácticas y tendencias para el desarrollo de software incluyen:

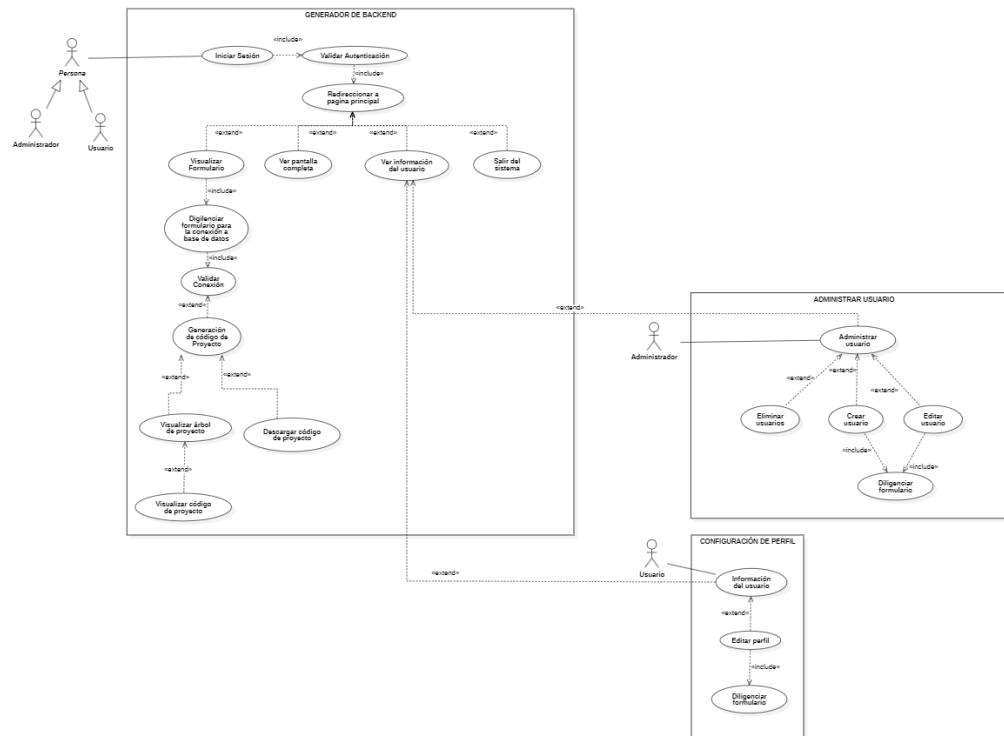
- ✓ **Desarrollo Ágil:** El enfoque ágil se ha convertido en una práctica estándar en muchas empresas. Se caracteriza por ciclos de desarrollo cortos y entregas frecuentes, lo que permite una mayor flexibilidad y capacidad de respuesta a los cambios.
- ✓ **DevOps:** La integración de desarrollo y operaciones (DevOps) es una tendencia importante que busca mejorar la colaboración entre equipos de desarrollo y operaciones para acelerar el tiempo de lanzamiento al mercado, mejorar la calidad del software y aumentar la eficiencia operativa.
- ✓ **Microservicios:** La arquitectura de microservicios se centra en construir aplicaciones como conjuntos de servicios pequeños e independientes, lo que facilita la escalabilidad, la implementación y la mantenibilidad del software.
- ✓ **Computación en la nube:** El uso de servicios en la nube, como plataformas de infraestructura como servicio (IaaS) y plataformas como servicio (PaaS), está en aumento debido a su flexibilidad, escalabilidad y reducción de costos.
- ✓ **Seguridad en el desarrollo:** La seguridad es una preocupación cada vez mayor en el desarrollo de software. Las prácticas de seguridad en el desarrollo, como la revisión de código, las pruebas de penetración y la integración de herramientas de seguridad en el ciclo

## 10.4. Anexo: Casos de Uso

Este anexo describe varios casos de uso del generador de backend desarrollado. En la Figura 20 se describen los casos de uso de las diversas interacciones que los usuarios y administradores pueden tener con el generador de backend, desde la configuración inicial hasta la generación y despliegue de código.

Cada caso de uso adjuntado proporciona un escenario específico, los pasos involucrados y los resultados esperados. Esta documentación es invaluable para comprender cómo utilizar eficazmente el generador de backend y aprovechar al máximo sus capacidades para acelerar el desarrollo de aplicaciones web.

Figura 20. Caso de uso [19]



Fuente: Autores

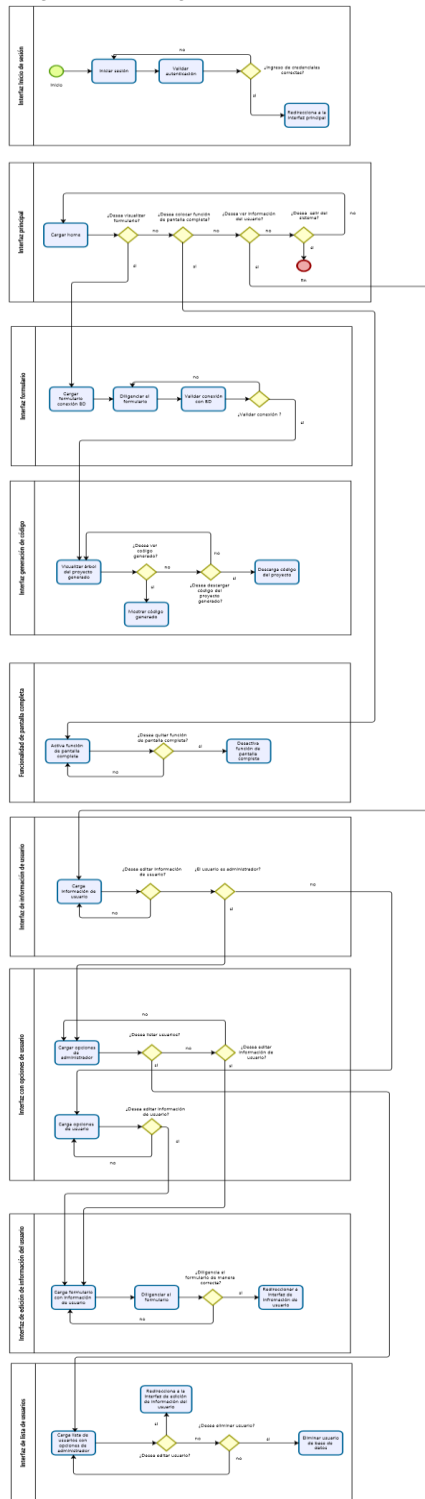
## 10.5. Anexo: Diagrama de Procesos

Este anexo incluye la Figura 21, que muestra el diagrama de procesos de la aplicación. En él se destacan las operaciones clave y su interacción para ofrecer una experiencia de usuario completa y fluida [20].

- Interfaz de inicio de sesión
  - ✓ Proceso que permite a los usuarios autenticarse en la aplicación con su nombre de usuario y contraseña.
  - ✓ Una vez autenticados, los usuarios son dirigidos a la interfaz principal.
  
- Interfaz principal
  - ✓ Pantalla de inicio después del inicio de sesión.
  - ✓ Muestra un resumen de las funcionalidades clave disponibles en la aplicación.
  
- Interfaz de formulario de conexión
  - ✓ Permite a los usuarios hacer la conexión a la base de datos relacional de PostgreSQL.
  
- Interfaz de generación de código
  - ✓ Herramienta que permite a los usuarios generar código establecida una conexión de base de datos relacional.
  
- Funcionalidad pantalla completa
  - ✓ Proporciona a los usuarios la opción de visualizar la aplicación en modo de pantalla completa.
  - ✓ Mejora la experiencia del usuario al eliminar distracciones.
  
- Interfaz de información de usuario
  - ✓ Muestra información personal y detalles de la cuenta del usuario.
  - ✓ Incluye datos como nombre, dirección de correo electrónico y fecha de creación del usuario.

- Interfaz con opciones de usuario
  - ✓ Proceso que permite a los usuarios ajustar configuraciones personales.
  
- Interfaz de edición de información del usuario
  - ✓ Permite a los usuarios editar y actualizar su información personal.
  - ✓ Ofrece un formulario de edición sencillo e intuitivo.
  
- Interfaz de lista de usuarios
  - ✓ Muestra una lista de todos los usuarios en la aplicación.
  - ✓ Puede incluir funciones de eliminar, editar o crear nuevos usuarios.

Figura 21. Diagrama de procesos [21]



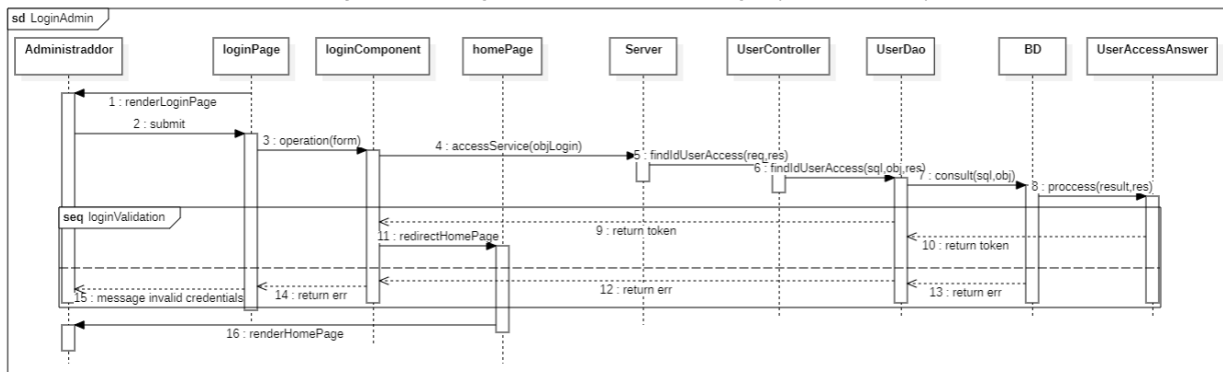
Fuente: Autores

## 10.6. Anexo: Diagramas de secuencia

El presente anexo proporciona una visión detallada del flujo de interacción entre los distintos componentes y procesos del generador de backend desarrollado. Este diagrama de secuencias ofrece una representación gráfica de las acciones y eventos que tienen lugar durante la ejecución del generador, desde la solicitud inicial hasta la respuesta final.

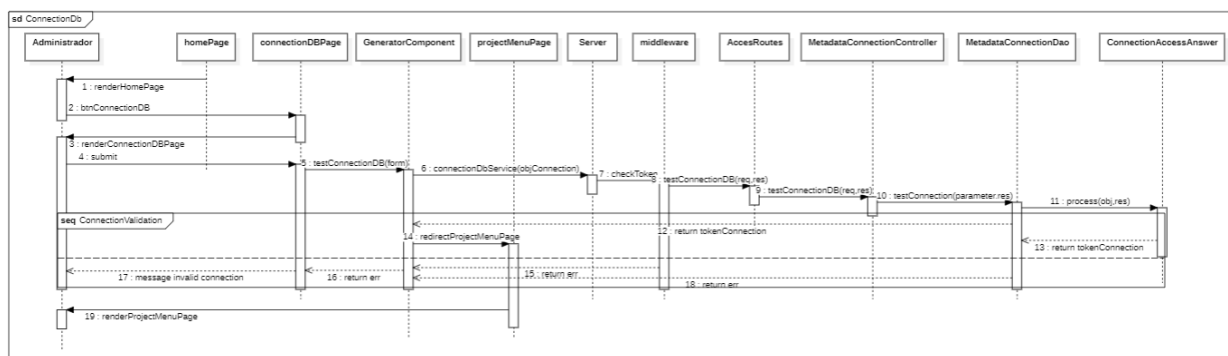
En la Figura 22, Figura 23, Figura 24, Figura 25, Figura 26, Figura 27, Figura 28, Figura 29, Figura 30, Figura 31, Figura 32, Figura 33, Figura 34, Figura 35, Figura 36 y en la Figura 37 se presenta los diagramas de secuencias correspondientes al administrador, quien cuenta con privilegios adicionales dentro del sistema. Estos diagramas detallan las interacciones clave entre el administrador y los distintos componentes del sistema, incluyendo la autenticación y la administración de usuarios. [22]

Figura 22. Diagrama de secuencia "Login (Administrator)"



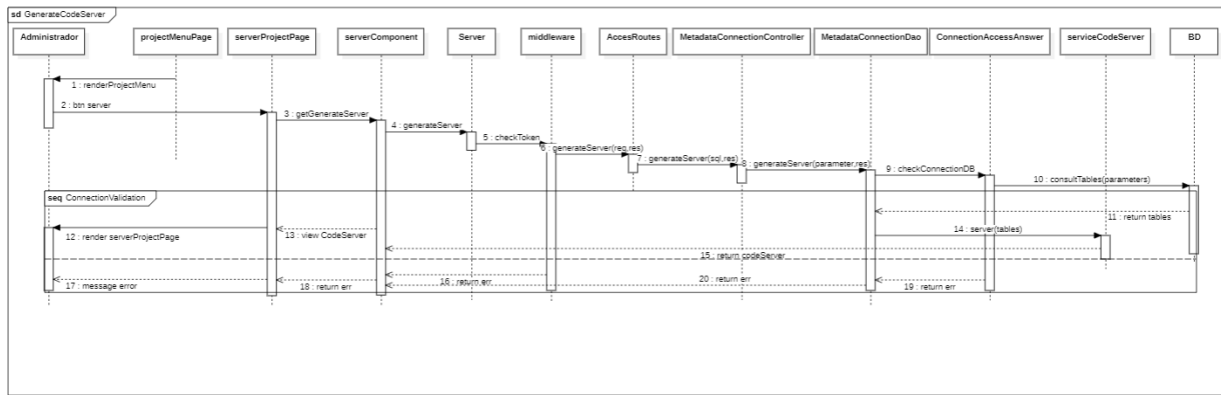
Fuente: Autores

Figura 23. Diagrama de secuencia "Connection database (Administrator)"



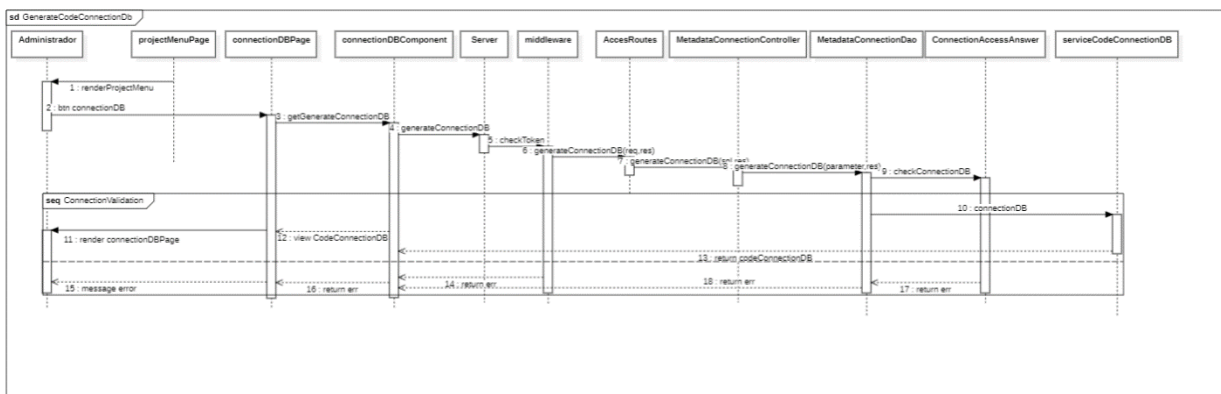
Fuente: Autores

Figura 24. Diagrama de secuencia "Generate code server (Administrator)"



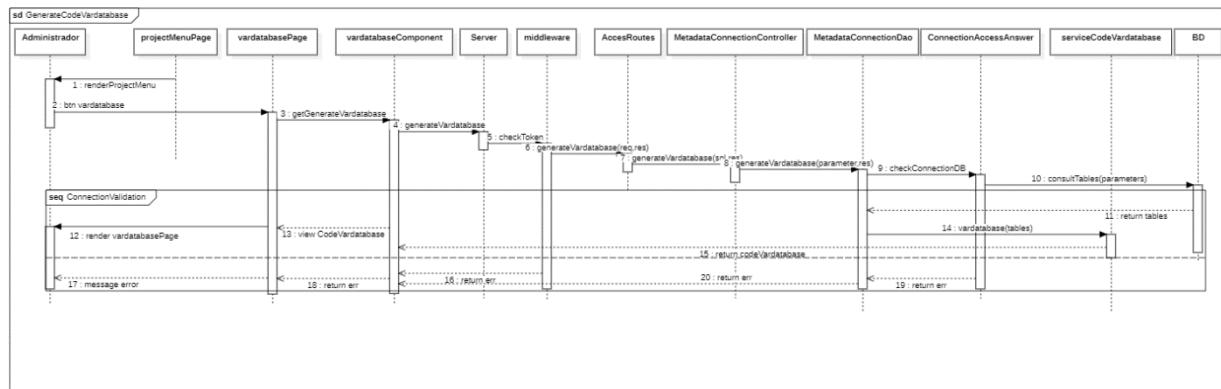
Fuente: Autores

Figura 25. Diagrama de secuencia "Generate code connectionDB (Administrator)"



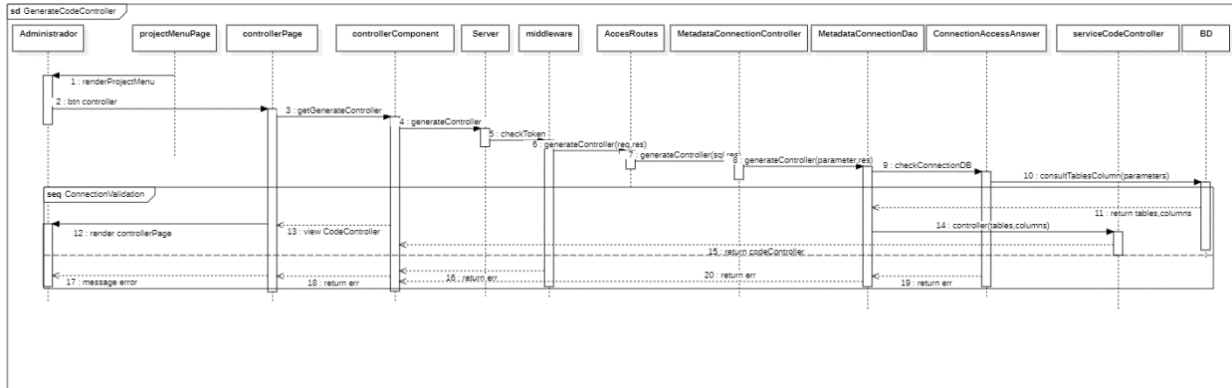
Fuente: Autores

Figura 26. Diagrama de secuencia "Generate code vardatabase (Administrator)"



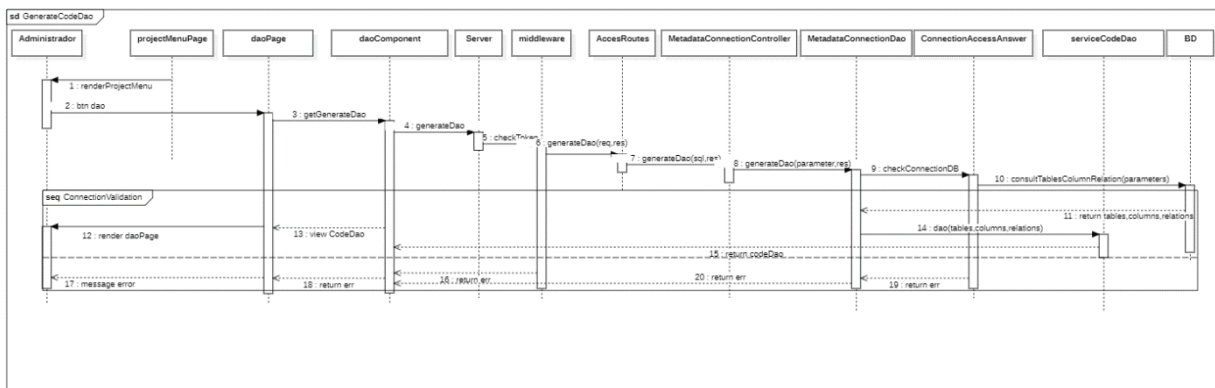
Fuente: Autores

Figura 27. Diagrama de secuencia "Generate code controllers (Administrator)"



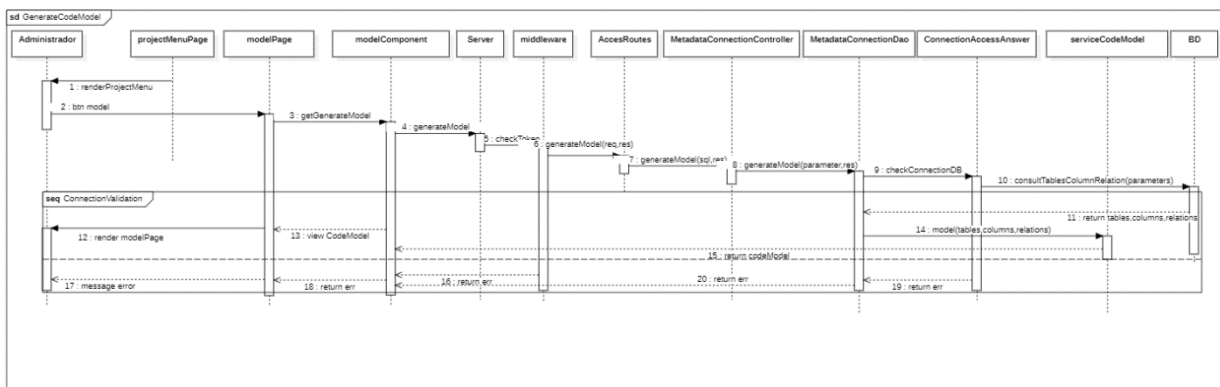
Fuente: Autores

Figura 28. Diagrama de secuencia "Generate code daos (Administrator)"



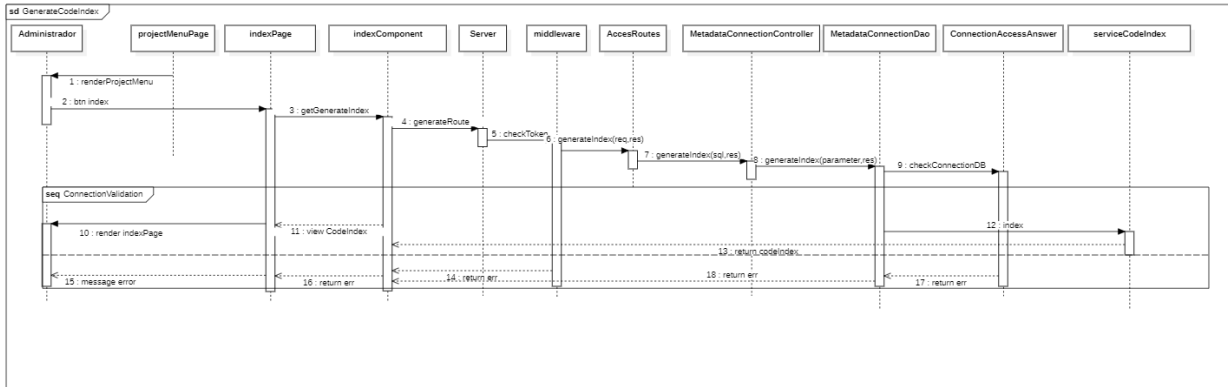
Fuente: Autores

Figura 29. Diagrama de secuencia "Generate code models (Administrator)"



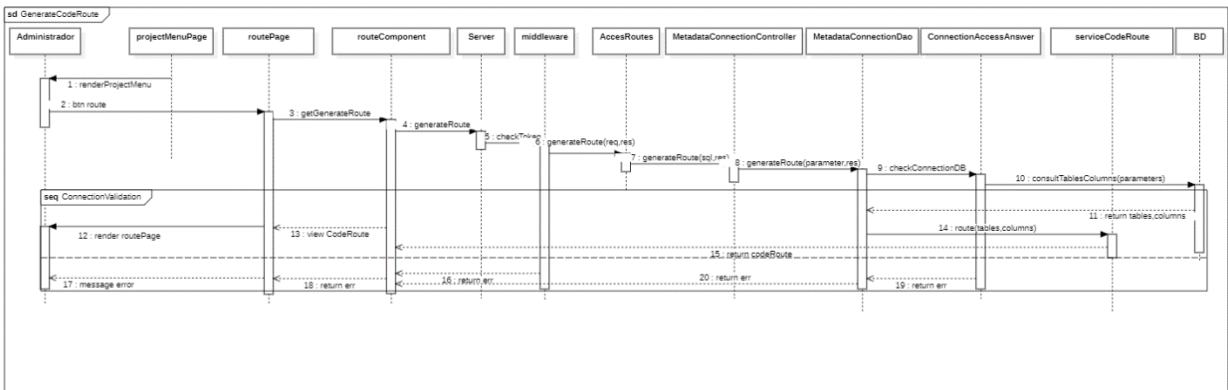
Fuente: Autores

Figura 30. Diagrama de secuencia “Generate code routes (Administrador)”



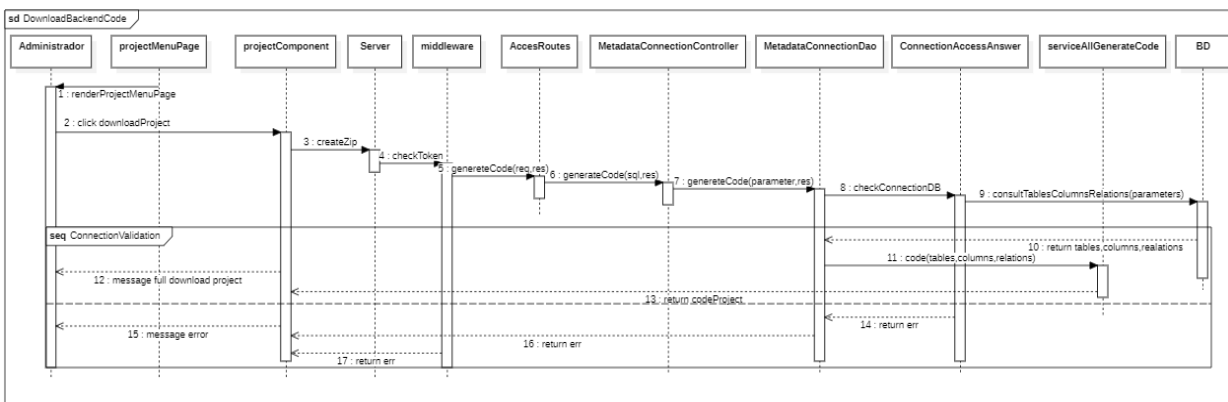
Fuente: Autores

Figura 31. Diagrama de secuencia “Generate code index (Administrador)”



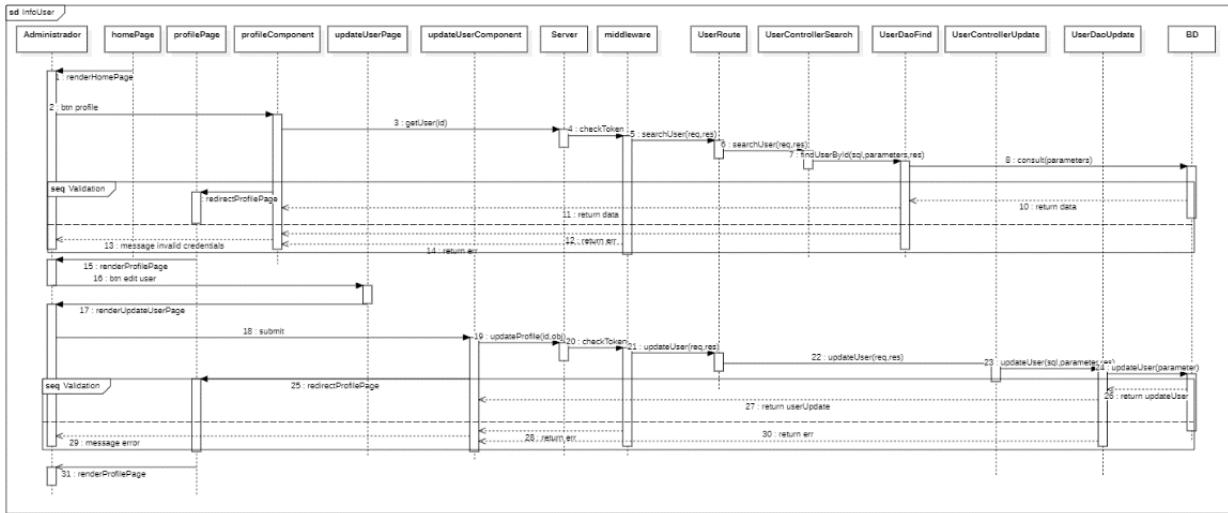
Fuente: Autores

Figura 32. Diagrama de secuencia “Download backend code (Administrador)”



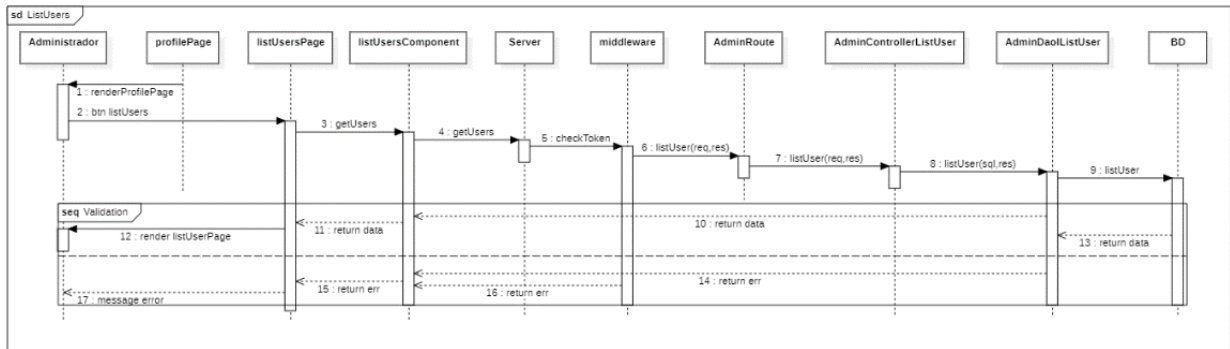
Fuente: Autores

Figura 33. Diagrama de secuencia "Setting and profile (Administrator)"



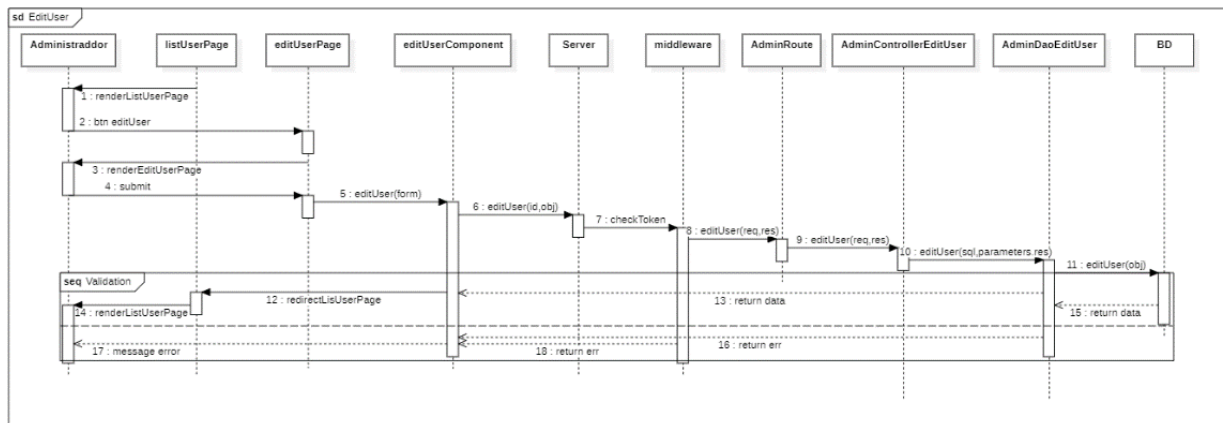
Fuente: Autores

Figura 34. Diagrama de secuencia "List users (Administrator)"



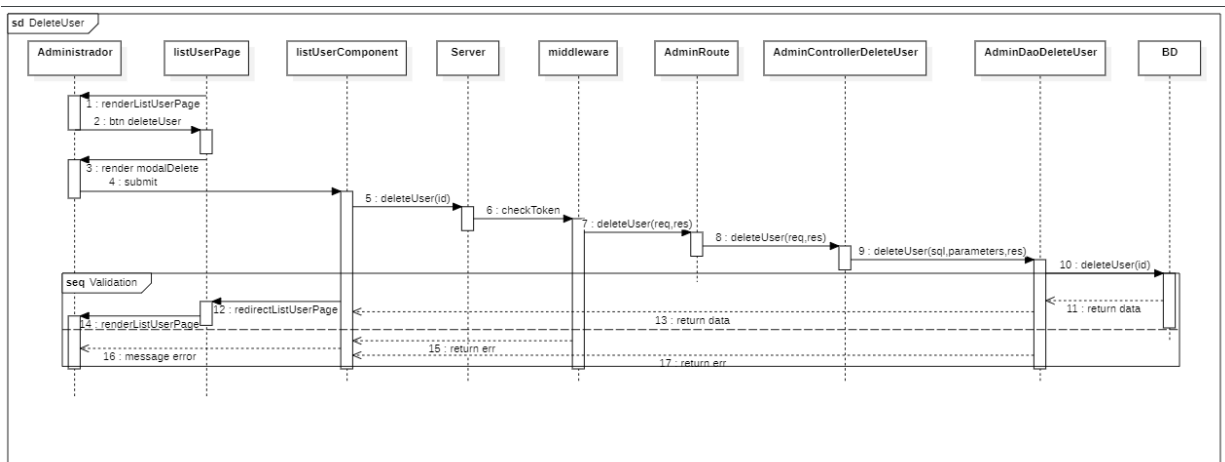
Fuente: Autores

Figura 35. Diagrama de secuencia "Edit users (Administrator)"



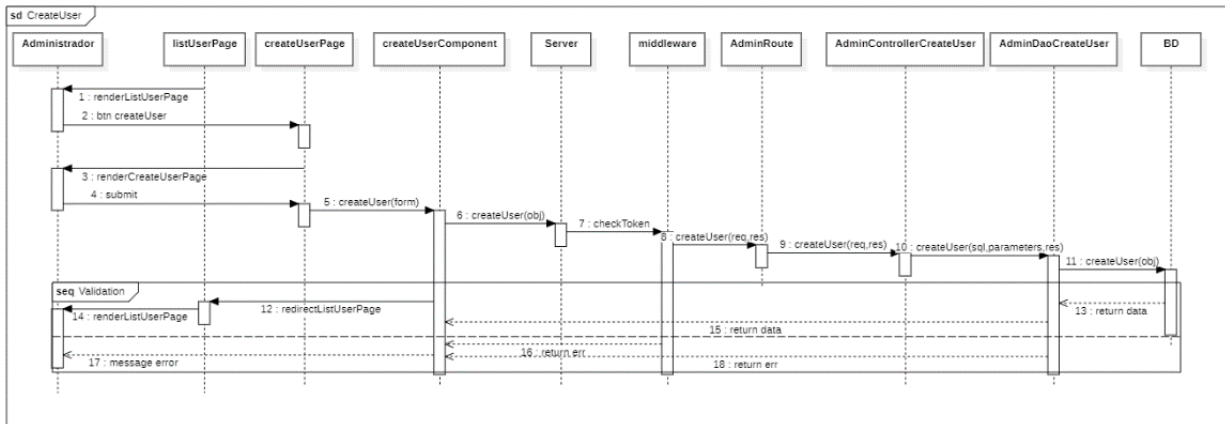
Fuente: Autores

Figura 36. Diagrama de secuencia "Delete users (Administrator)"



Fuente: Autores

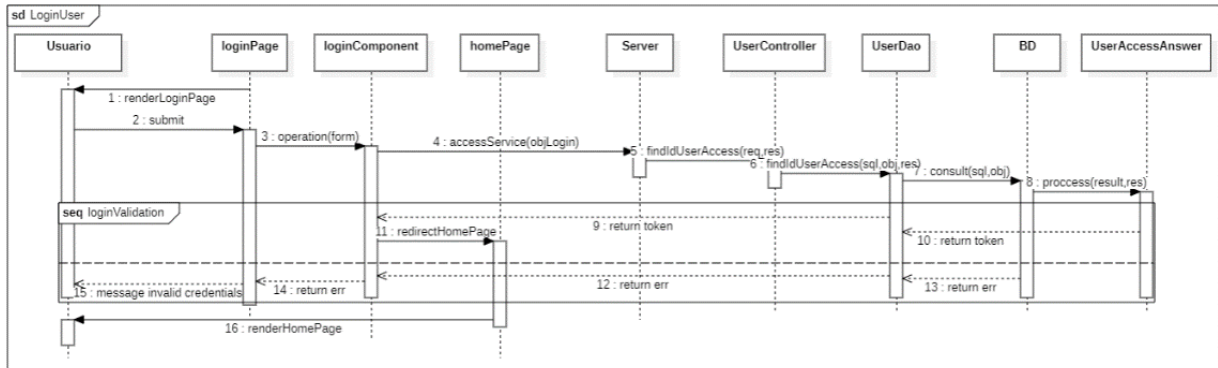
Figura 37. Diagrama de secuencia "Create users (Administrator)"



Fuente: Autores

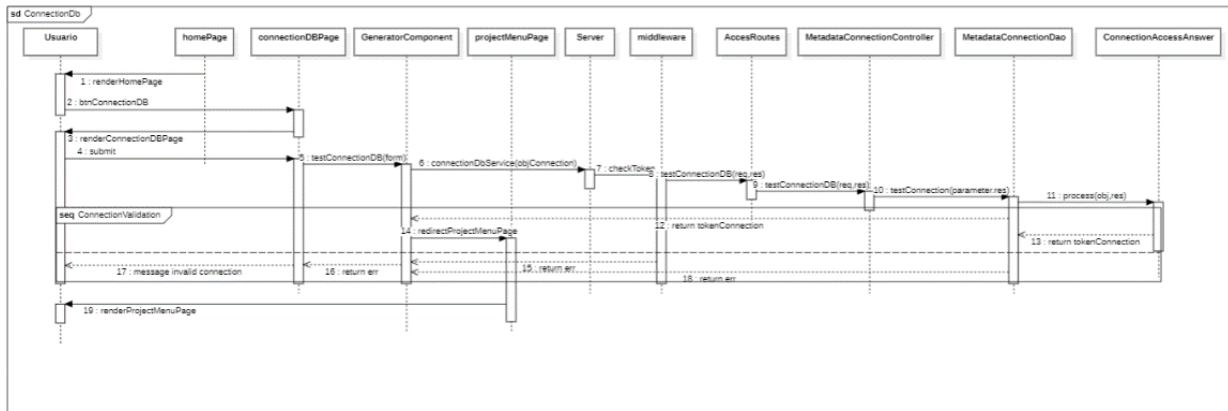
Por otro lado, en la Figura 38, Figura 39, Figura 40, Figura 41, Figura 42, Figura 43, Figura 44, Figura 45, Figura 46, Figura 47, Figura 48 y en la Figura 49 se muestra los diagramas de secuencias del usuario estándar. Estos diagramas describen las interacciones típicas entre el usuario y el sistema, centrándose en las funcionalidades disponibles para el mismo. Estas interacciones pueden incluir la autenticación, la navegación por el sistema, la realización de acciones específicas y la gestión de recursos disponibles para el usuario. [23]

Figura 38. Diagrama de secuencia "Login (User)"



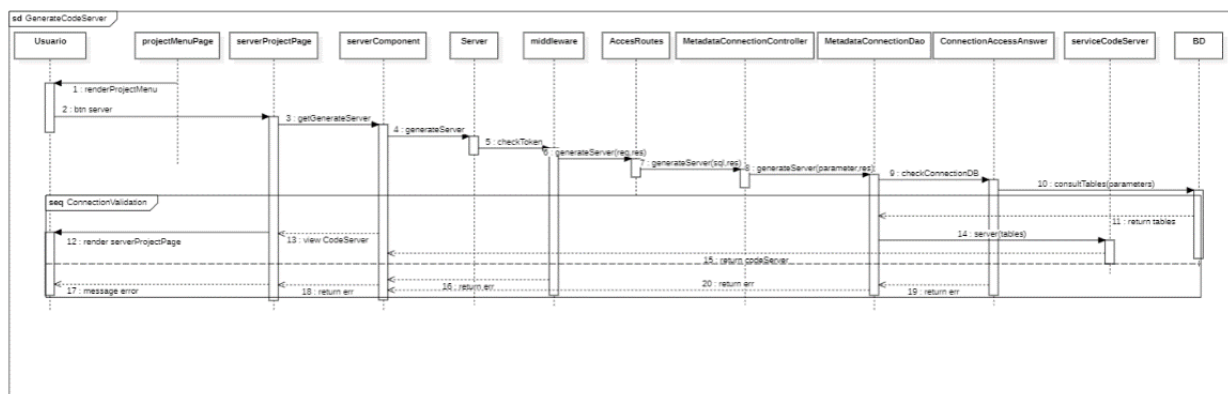
Fuente: Autores

Figura 39. Diagrama de secuencia "Connection database (User)"



Fuente: Autores

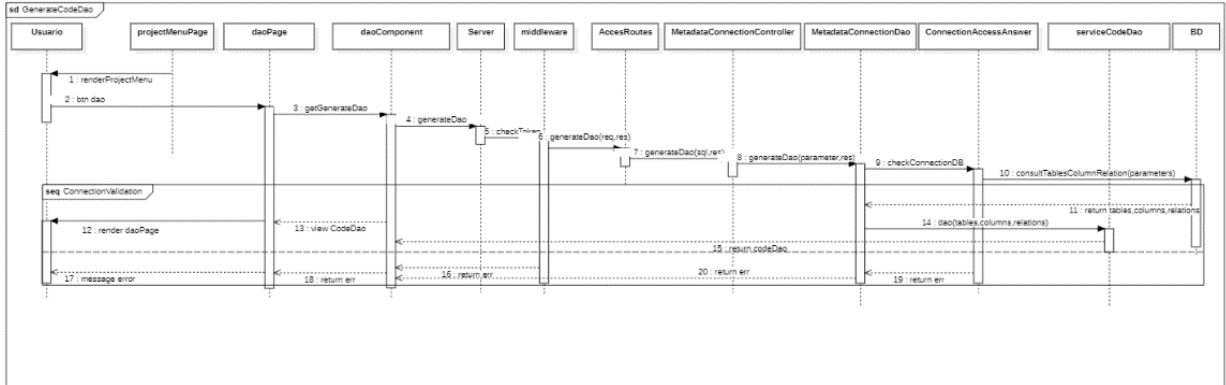
Figura 40. Diagrama de secuencia "Generate code server (User)"



Fuente: Autores

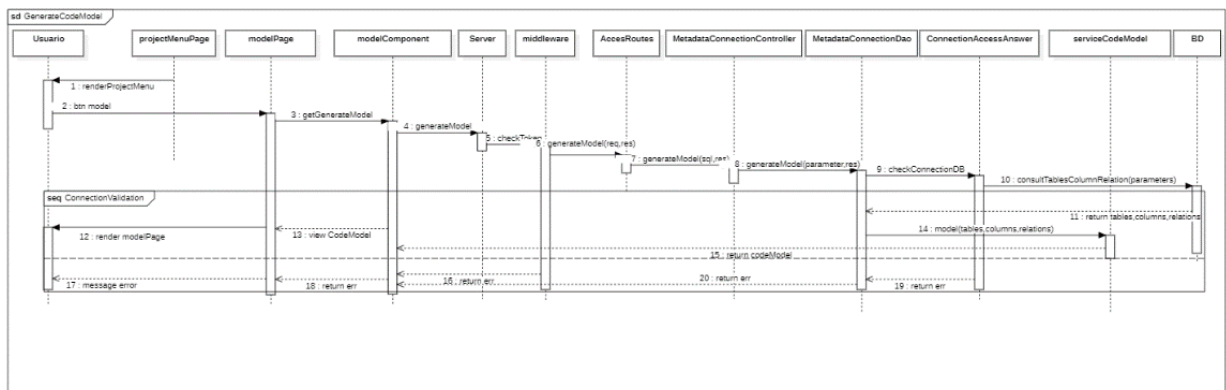


Figura 44. Diagrama de secuencia "Generate code daos (User)"



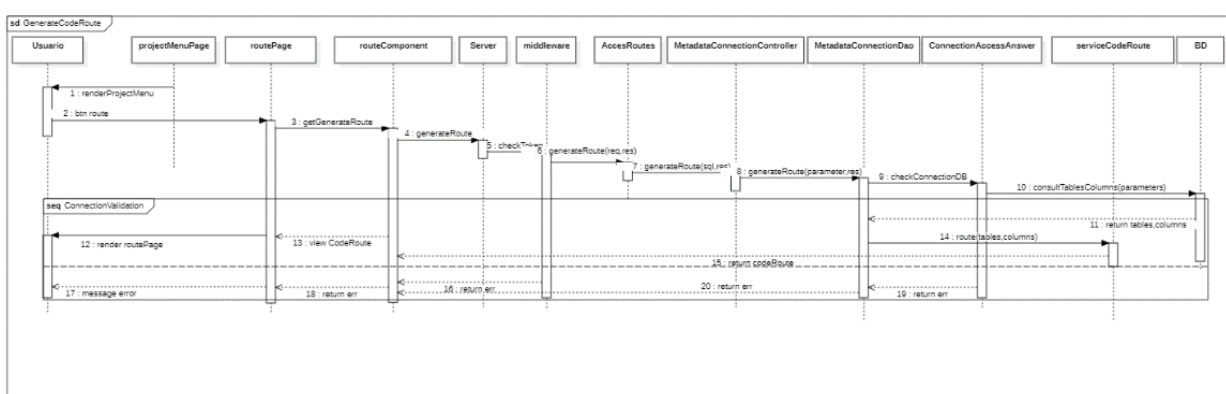
Fuente: Autores

Figura 45. Diagrama de secuencia "Generate code models (User)"



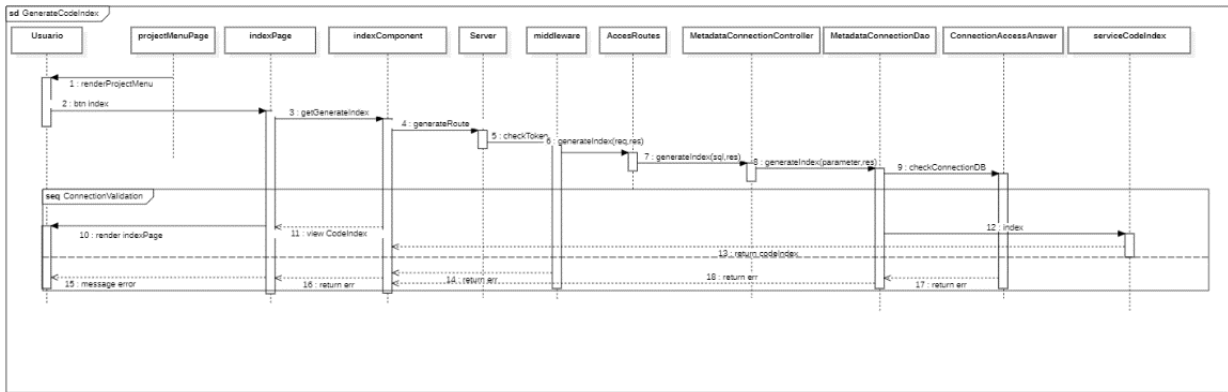
Fuente: Autores

Figura 46. Diagrama de secuencia "Generate code routes (User)"



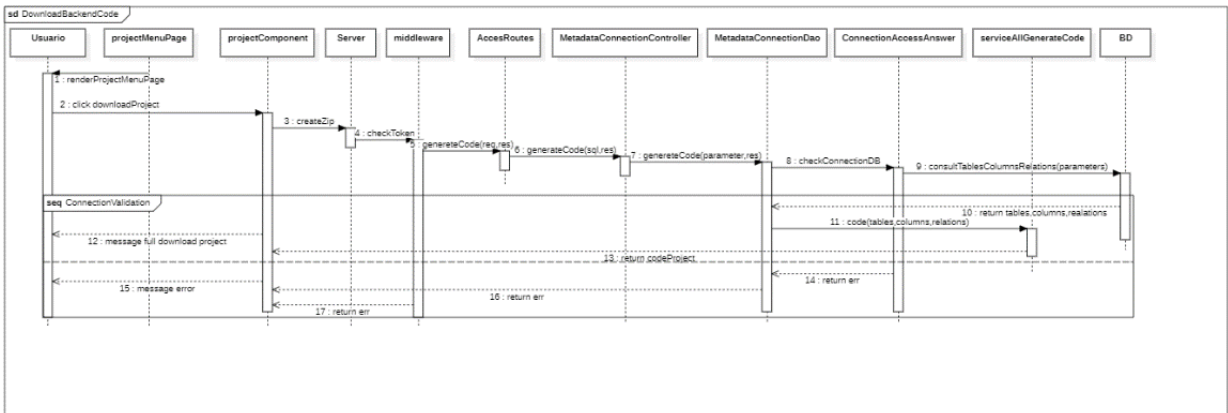
Fuente: Autores

Figura 47. Diagrama de secuencia "Generate code index (User)"



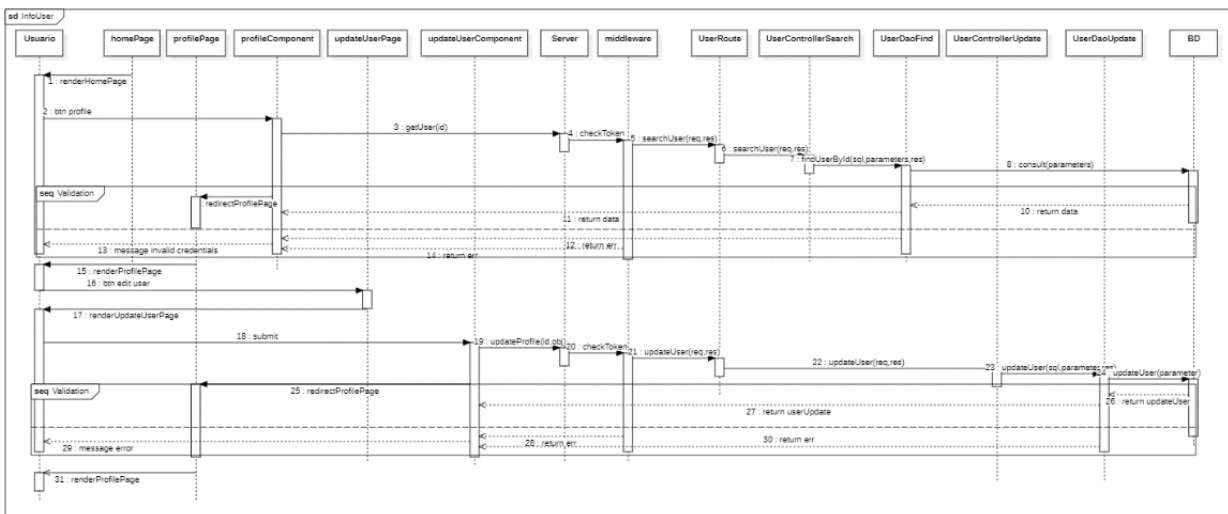
Fuente: Autores

Figura 48. Diagrama de secuencia "Download backend code (User)"



Fuente: Autores

Figura 49. Diagrama de secuencia "Setting and profile (User)"



Fuente: Autores

## 10.7. Anexo: Requerimientos funcionales y no funcionales

### 10.7.1. Requerimientos funcionales

Identificador	Nombre
RF 1	Iniciar Sesión
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
<p>Implementar un sistema de autenticación seguro que permita a los usuarios acceder a sus cuentas mediante la generación y verificación de tokens de sesión. Los tokens tendrán una duración de una hora y se aplicarán medidas de seguridad para proteger la información de los usuarios.</p>	
Precondición	
<ul style="list-style-type: none"> <li>• Los usuarios podrán iniciar sesión utilizando sus credenciales previamente registradas.</li> <li>• Se generará un token de sesión único y seguro para cada usuario válido.</li> <li>• Los tokens de sesión serán válidos por una duración de una hora desde el momento de la generación.</li> <li>• Se implementará un algoritmo de generación de tokens seguro y no predecible.</li> <li>• Las contraseñas de los usuarios se almacenarán utilizando algoritmos de seguros, como pgcrypto extensión para encriptar datos en Postgresql.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ Usuario</li> <li>✓ Contraseña</li> </ul>	<ul style="list-style-type: none"> <li>✓ Menú principal</li> <li>✓ Token valido por 1 hora</li> </ul>
Postcondición	
<ul style="list-style-type: none"> <li>• El sistema verificará los datos de entrada usuario y contraseña con el que el usuario va a ingresar a sistema.</li> <li>• El usuario puede realizar acciones correctivas, como volver a ingresar las credenciales.</li> </ul>	

Manejo de situaciones anormales
<ul style="list-style-type: none"> <li>• Se proporcionarán mensajes de error claros y específicos en caso de fallos en el inicio de sesión debido a credenciales incorrectas o tokens inválidos.</li> <li>• Se proporciona al usuario la opción de intentar iniciar sesión nuevamente ingresando las credenciales correctas.</li> <li>• Después de transcurrido el tiempo de expiración, los tokens serán inválidos y requerirán un nuevo inicio de sesión para obtener uno nuevo.</li> </ul>

Identificador	Nombre
RF 2	Menú lateral
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
<p>Este requerimiento describe la implementación de un menú lateral en la interfaz de usuario, que proporciona acceso a diversas funciones y opciones de navegación. El menú lateral contendrá opciones para la navegación en el menú principal, formulario para la conexión a base de datos, ajustes de perfil y salida del sistema.</p>	
Precondición	
<ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en el sistema y tiene acceso a la interfaz de usuario.</li> <li>• La interfaz de usuario incluye un área designada para el menú lateral.</li> </ul>	
Entrada	Salida
Interacción del usuario con el menú lateral, como clics en los íconos o enlaces correspondientes.	<ul style="list-style-type: none"> <li>✓ Navegación a diferentes secciones o acciones dentro del sistema.</li> <li>✓ Visualización de opciones específicas según la selección del usuario.</li> </ul>
Postcondición	
<ul style="list-style-type: none"> <li>• El usuario puede acceder y utilizar las funciones y opciones proporcionadas en el menú lateral.</li> <li>• La interfaz de usuario refleja las selecciones y acciones realizadas por el usuario</li> </ul>	

en el menú lateral.
<b>Manejo de situaciones anormales</b>
<p>Si se detecta una interacción inválida o inesperada por parte del usuario, como un clic en un ícono no funcional:</p> <ul style="list-style-type: none"> <li>• No se realizará ninguna acción y se mantendrá el estado actual de la interfaz.</li> <li>• Se proporciona un mensaje de retroalimentación para informar al usuario sobre la interacción no válida.</li> </ul>
<b>Características del Menú lateral</b>
<ul style="list-style-type: none"> <li>✓ <b>Ícono de Menú:</b> Despliega el menú principal para acceder a toda la información de la página web.</li> <li>✓ <b>Ícono del generador:</b> Proporciona un formulario de conexión a la base de datos, permitiendo a los usuarios atraer metadatos y generar el código CRUD del backend, ya sea desde una base de datos local o en la nube.</li> <li>✓ <b>Ícono de pantalla completa:</b> Permite al usuario cambiar a modo de pantalla completa para una mejor visualización.</li> <li>✓ <b>Ícono de configuración de perfil:</b> Permite al usuario acceder y modificar la configuración de su perfil.</li> <li>✓ <b>Ícono de cerrar sesión:</b> Permite al usuario cerrar sesión y salir del sistema.</li> </ul>

Identificador	Nombre
RF 3	Menú principal
Actor	Prioridad de desarrollo
Usuario	Media
Descripción	
<p>Este requerimiento describe la implementación del menú principal de la aplicación web, que proporcionará acceso a información sobre el generador de backends y guiará a los usuarios a través de un paso a paso para utilizar esta herramienta, desde la conexión hasta la descarga del código generado. Además, se identifican los puntos altos y los problemas a solucionar con esta aplicación web.</p>	

<b>Precondición</b>	
<ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en la aplicación web y tiene acceso al menú principal.</li> <li>• La interfaz de usuario incluye un área designada para el menú principal y la presentación de información detallada sobre el generador de backends.</li> </ul>	
<b>Entrada</b>	<b>Salida</b>
Interacción del usuario con el menú principal y las opciones relacionadas con el generador de backends.	<ul style="list-style-type: none"> <li>✓ Navegación a diferentes secciones de la aplicación web, incluyendo información detallada sobre el generador de backends y el proceso de generación de código.</li> <li>✓ Instrucciones paso a paso para utilizar el generador de backends, desde la conexión hasta la descarga del código generado.</li> </ul>
<b>Postcondición</b>	
<ul style="list-style-type: none"> <li>✓ Los usuarios pueden acceder y utilizar la información proporcionada sobre el generador de backends.</li> <li>✓ Los usuarios pueden seguir correctamente el paso a paso para utilizar el generador de backends y descargar el código generado.</li> </ul>	
<b>Manejo de situaciones anormales</b>	
Si se detecta una interacción inválida o inesperada por parte del usuario, como un clic en un ícono no funcional: <ul style="list-style-type: none"> <li>• No se realizará ninguna acción y se mantendrá el estado actual de la interfaz.</li> <li>• Se proporciona un mensaje de retroalimentación para informar al usuario sobre la interacción no válida.</li> </ul>	
<b>Características del Menú Principal</b>	
<ul style="list-style-type: none"> <li>✓ Información sobre el generador de Backends:                              Proporciona detalles sobre las características, funcionalidades y ventajas del generador de backends.</li> <li>✓ Paso a paso para utilizar el generador:                              Guía a los usuarios a través de los pasos necesarios para conectar, configurar y utilizar el generador de backends.</li> <li>✓ Puntos altos:</li> </ul>	

Destaca los principales beneficios y puntos fuertes del generador de backends, como la eficiencia, la flexibilidad y la facilidad de uso.

✓ Problemas por solucionar:  
 Identifica los problemas comunes que la aplicación web busca resolver, como la complejidad en el desarrollo de backends, la falta de conocimientos técnicos y los retrasos en los plazos de entrega de proyectos.

Identificador	Nombre
RF 4	Formulario de conexión para el generador
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
<p>Este requerimiento describe la implementación de un formulario de conexión que se activa cuando el usuario hace clic en el icono del generador. La vista mostrará un menú lateral inicial con opciones de navegación y el formulario que recopilará la información necesaria para establecer una conexión con la base de datos y permitirá al usuario especificar detalles como el nombre de usuario, el host, la base de datos, la contraseña y el puerto.</p>	
Precondición	
<ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión en la aplicación web.</li> <li>• El usuario ha hecho clic en el icono del generador en el menú principal.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ Nombre de usuario para la conexión a la base de datos.</li> <li>✓ Dirección del host donde se encuentra la base de datos.</li> <li>✓ Nombre de la base de datos a la que se desea conectar.</li> <li>✓ Contraseña asociada al usuario para la conexión.</li> <li>✓ Puerto de conexión utilizado para acceder a la base de datos.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Establecimiento de una conexión exitosa con la base de datos utilizando los detalles proporcionados por el usuario.</li> <li>✓ Notificación al usuario en caso de fallo de conexión o errores en el formulario.</li> <li>✓ Notificación al usuario en caso de éxito de la conexión.</li> </ul>

<b>Postcondición</b>
<ul style="list-style-type: none"> <li>• Los usuarios pueden proporcionar la información necesaria para conectarse a la base de datos utilizando el formulario.</li> <li>• El sistema utiliza los detalles de conexión para establecer una conexión con la base de datos y activar las funcionalidades del generador.</li> </ul>
<b>Manejo de situaciones anormales</b>
<p>Si la conexión a la base de datos falla debido a credenciales incorrectas, falta de conexión a Internet u otros problemas:</p> <ul style="list-style-type: none"> <li>• Se muestra un mensaje de error al usuario indicando el problema y proporcionando instrucciones para solucionarlo.</li> <li>• Se ofrece la opción de volver a intentar la conexión o de revisar y corregir los detalles de conexión ingresados.</li> </ul>

<b>Identificador</b>	<b>Nombre</b>
RF 5	Vista de proyecto backend (Directorio de archivos)
<b>Actor</b>	<b>Prioridad de desarrollo</b>
Usuario	Alta
<b>Descripción</b>	
<p>Este requerimiento describe la vista que se presenta al usuario después de una conexión exitosa desde el formulario de conexión. La vista mostrará un menú superior que incluye el nombre de la base de datos, botones para descargar el proyecto backend y desconectarse de la base de datos. Además, se presentará el menú lateral inicial con opciones de navegación, y otro menú lateral que muestra la estructura del proyecto backend.</p>	
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>• El usuario ha establecido una conexión exitosa con la base de datos utilizando el formulario de conexión.</li> <li>• La aplicación web ha recibido confirmación de la conexión exitosa y ha cargado correctamente la vista correspondiente.</li> </ul>	
<b>Entrada</b>	<b>Salida</b>

Interacción del usuario con los elementos de la vista, como hacer clic en los botones del menú superior o en las opciones del menú lateral.	Visualización de la vista que muestra el menú superior con información de la base de datos, botones para descargar el proyecto backend y desconectarse, y menús laterales para navegación y visualización de la estructura del proyecto backend.
<b>Postcondición</b>	
El usuario puede navegar por la vista y acceder a las funcionalidades proporcionadas, como descargar el proyecto backend y desconectarse de la base de datos.	
<b>Manejo de situaciones anormales</b>	
Si se produce un error al cargar la vista debido a problemas técnicos o de comunicación: <ul style="list-style-type: none"> <li>• Se muestra un mensaje de error al usuario indicando el problema y proporcionando instrucciones para intentar nuevamente o contactar al soporte técnico.</li> </ul>	
<b>Características de la vista del proyecto backend</b>	
<ul style="list-style-type: none"> <li>✓ <b>Menú superior:</b> Incluye el nombre de la base de datos conectada y botones para descargar el proyecto backend y desconectarse de la base de datos.</li> <li>✓ <b>Menú lateral de navegación:</b> Muestra opciones de navegación para acceder a diferentes secciones de la aplicación web.</li> <li>✓ <b>Menú lateral del proyecto backend:</b> Muestra la estructura del proyecto backend, incluyendo controladores, modelos, rutas, y otros archivos como index.ts, package.json y package-lock.json.</li> </ul>	

Identificador	Nombre
RF 6	Visualización de código
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
Este requerimiento detalla cómo los usuarios pueden ver el código correspondiente a cada	

sección del proyecto backend al hacer clic en los elementos del árbol de la estructura del proyecto. Al hacer clic en controladores, modelos, rutas y otros archivos como index.ts, package.json y package-lock.json, se mostrará un área de texto que contiene el código correspondiente a la sección seleccionada.

#### Precondición

- El usuario ha establecido una conexión exitosa con la base de datos y ha cargado la vista correspondiente que muestra el menú lateral con la estructura del proyecto backend.
- La aplicación web ha cargado correctamente la estructura del proyecto backend en el menú lateral y tiene acceso al código de cada sección.

#### Entrada

Interacción del usuario con los elementos del árbol de la estructura del proyecto backend, como hacer clic en controladores, modelos, rutas, index.ts, package.json o package-lock.json.

#### Salida

Visualización del código correspondiente a la sección seleccionada en un área de texto.

#### Postcondición

Los usuarios pueden ver el código de cada sección del proyecto backend al hacer clic en los elementos del árbol de la estructura del proyecto.

#### Manejo de situaciones anormales

Si el elemento seleccionado no se encuentra o no existe en el proyecto backend:

- Se muestra un mensaje de advertencia al usuario indicando que el elemento seleccionado no se pudo encontrar.

#### Características

- ✓ Árbol de estructura del proyecto backend:  
Muestra una lista jerárquica de controladores, modelos, rutas y otros archivos del proyecto backend.
- ✓ Área de texto para visualización de código:  
Muestra el código correspondiente a la sección seleccionada del proyecto backend.

Identificador	Nombre
RF 7	Gestión de contenido en la sección seleccionada del proyecto backend
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
<p>Este requerimiento detalla cómo los usuarios pueden gestionar el contenido en la sección seleccionada del proyecto backend. Cada sección mostrará el nombre del archivo y un área de texto que contiene el código correspondiente. Se proporcionará un botón para copiar todo el contenido del archivo, así como la capacidad de seleccionar y copiar dos o tres líneas específicas del área de texto.</p>	
Precondición	
<ul style="list-style-type: none"> <li>• El usuario ha establecido una conexión exitosa con la base de datos y ha cargado la vista correspondiente que muestra la estructura del proyecto backend.</li> <li>• La aplicación web ha cargado correctamente la estructura del proyecto backend y tiene acceso al código de cada sección.</li> </ul>	
Entrada	Salida
<p>Interacción del usuario con la sección seleccionada del proyecto backend, incluyendo el nombre del archivo y el área de texto que contiene el código.</p>	<ul style="list-style-type: none"> <li>✓ Visualización del nombre del archivo y el contenido correspondiente en el área de texto.</li> <li>✓ Copia del contenido completo del archivo al hacer clic en el botón de copiar.</li> <li>✓ Copia de dos o tres líneas específicas del área de texto al seleccionarlas.</li> </ul>
Postcondición	
<p>Los usuarios pueden gestionar el contenido en la sección seleccionada del proyecto backend, copiando el contenido completo o partes específicas según sea necesario.</p>	
Manejo de situaciones anormales	
<p>Si el elemento seleccionado no se encuentra o no existe en el proyecto backend:</p>	

<ul style="list-style-type: none"> <li>• Se muestra un mensaje de advertencia al usuario indicando que el elemento seleccionado no se pudo encontrar.</li> </ul>
<b>Características</b>
<ul style="list-style-type: none"> <li>✓ <b>Nombre del archivo:</b> Se muestra el nombre del archivo en la parte superior de la sección seleccionada del proyecto backend.</li> <li>✓ <b>Área de texto para visualización de código:</b> Muestra el contenido del archivo seleccionado en un área de texto.</li> <li>✓ <b>Botón para copiar contenido completo:</b> Permite al usuario copiar todo el contenido del archivo seleccionado con un solo clic.</li> <li>✓ <b>Capacidad de seleccionar y copiar líneas específicas:</b> Permite al usuario seleccionar dos o tres líneas específicas del área de texto y copiarlas.</li> </ul>

Identificador	Nombre
RF 8	Configuración de perfil
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
<p>Esta funcionalidad permite a los usuarios ver los detalles completos de su perfil al hacer clic en su propio icono de usuario. La vista de detalles del usuario incluye su nombre, correo electrónico, fecha de creación de la cuenta y su foto de perfil. Además, con botón para edición de información.</p>	
Precondición	
<ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en la aplicación web.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ Clic en su propio icono de usuario.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Se carga una nueva vista que muestra los detalles del usuario, incluyendo su nombre, correo electrónico, fecha de creación de la cuenta y su foto de perfil, si está disponible; de lo contrario se visualizará una por defecto. Además, se incluye un botón con icono de "Editar información" que permite al usuario</li> </ul>

	cargar la vista de edición.
Postcondición	
<ul style="list-style-type: none"> <li>La vista de detalles del usuario se muestra de manera correcta y completa, mostrando la información actualizada y precisa del usuario.</li> </ul>	
Manejo de situaciones anormales	
<p><b>Fallo al cargar los detalles del usuario:</b> Si ocurre un error al intentar cargar los detalles del usuario, se mostrará un mensaje de error indicando la causa del fallo y se registrará el error en un registro de errores.</p> <p><b>Foto de perfil no disponible:</b> Si el usuario no tiene una foto de perfil almacenada en la base de datos, se mostrará una imagen por defecto en su lugar.</p> <p><b>Retraso en la carga de la vista:</b> Si hay un retraso excesivo en la carga de la vista de detalles del usuario, se optimizará el rendimiento para garantizar una experiencia de usuario fluida.</p>	

Identificador	Nombre
RF 9	Edición de información de usuario
Actor	Prioridad de desarrollo
Usuario	Alta
Descripción	
<p>Esta funcionalidad permite a los usuarios editar su información personal, incluyendo nombre, correo electrónico, contraseña y foto de perfil. Al hacer clic en el botón de editar desde la vista de detalles del usuario, se muestra un formulario prellenado con los datos actuales del usuario.</p>	
Precondición	
<ul style="list-style-type: none"> <li>El usuario debe estar autenticado en la aplicación web.</li> <li>Acceder a la vista de edición de información de usuario desde la vista de detalles de su perfil.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ El usuario hace clic en el botón de editar desde la vista de detalles del usuario.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Se muestra un formulario prellenado con los datos actuales del usuario, incluyendo nombre, correo electrónico y una vista previa de la foto de perfil actual. El formulario incluye campos para editar la información del usuario, así como</li> </ul>

	botones para mostrar y ocultar la contraseña y para seleccionar una nueva foto de perfil.
<b>Postcondición</b>	
<ul style="list-style-type: none"> <li>Después de editar la información del usuario y enviar el formulario correctamente, los datos del usuario se actualizan en la base de datos. Se muestra un mensaje de confirmación de que la operación se realizó con éxito y se redirecciona al usuario a la vista de información del usuario.</li> </ul>	
<b>Manejo de situaciones anormales</b>	
<ul style="list-style-type: none"> <li>Si la contraseña y la re-contraseña no coinciden al enviar el formulario, se muestra un mensaje de error indicando que las contraseñas no son iguales.</li> <li>Si el usuario intenta editar su correo electrónico a uno que ya está en uso, se muestra un mensaje de error indicando que el correo electrónico debe ser único.</li> <li>Si ocurre un error al enviar el formulario de edición de información del usuario, se muestra un mensaje de error.</li> <li>Selección de foto no válida: Si el usuario selecciona un archivo que no es una imagen válida al intentar cambiar la foto de perfil, se muestra un mensaje de error indicando que el formato de archivo no es compatible.</li> </ul>	

<b>Identificador</b>	<b>Nombre</b>
RF 10	Administrar usuarios
<b>Actor</b>	<b>Prioridad de desarrollo</b>
Administrador	Alta
<b>Descripción</b>	
El administrador debe poder realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la información de los usuarios del sistema a través de una interfaz de administración.	
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>El usuario debe estar autenticado en la aplicación web.</li> <li>Acceder a la vista de administrar perfiles desde la vista de su perfil.</li> </ul>	
<b>Entrada</b>	<b>Salida</b>
<ul style="list-style-type: none"> <li>La entrada consiste en los datos proporcionados por el administrador para realizar operaciones sobre los usuarios. Esto puede incluir datos como nombre, correo electrónico,</li> </ul>	La salida varía según la acción realizada: <ul style="list-style-type: none"> <li>Si se crea un usuario, se devuelve un mensaje de éxito y los detalles del nuevo usuario creado.</li> </ul>

<p>contraseña, rol, entre otros.</p> <ul style="list-style-type: none"> <li>Además, el administrador puede especificar la acción que desea realizar (crear, leer, actualizar o eliminar un usuario).</li> </ul>	<ul style="list-style-type: none"> <li>Si se lee un usuario, se devuelve la información del usuario solicitado.</li> <li>Si se actualiza un usuario, se devuelve un mensaje de éxito y los detalles actualizados del usuario.</li> <li>Si se elimina un usuario, se devuelve un mensaje de éxito confirmando la eliminación.</li> </ul>
<b>Postcondición</b>	
<ul style="list-style-type: none"> <li>La información del usuario se actualiza de acuerdo con las operaciones realizadas por el administrador.</li> <li>Se notifica al administrador sobre el resultado de la operación (éxito o fallo).</li> </ul>	
<b>Manejo de situaciones anormales</b>	
<ul style="list-style-type: none"> <li>Si el administrador no tiene permisos suficientes para acceder a la funcionalidad de administración de usuarios, se muestra un mensaje de error indicando la falta de autorización.</li> <li>Si se produce un error durante la ejecución de una operación (por ejemplo, error de validación de datos o error de base de datos), se muestra un mensaje de error detallado y se proporcionan instrucciones sobre cómo resolver el problema.</li> </ul>	

<b>Identificador</b>	<b>Nombre</b>
RF 11	Cierre de Sesión (Logout)
<b>Actor</b>	<b>Prioridad de desarrollo</b>
Usuario	Alta
<b>Descripción</b>	
<p>La funcionalidad de cierre de sesión permite a los usuarios salir de su cuenta actual y cerrar su sesión en la aplicación.</p>	
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>El usuario debe estar autenticado en la aplicación y haber iniciado sesión correctamente.</li> </ul>	
<b>Entrada</b>	<b>Salida</b>
<p>✓ El usuario realiza una acción específica para salir de su cuenta, como hacer clic en un botón de icono de "Cerrar sesión" o seleccionar la opción de "Logout" en el</p>	<p>✓ El usuario es redirigido a la vista de inicio de sesión de la aplicación.</p>

menú de la aplicación.
<b>Postcondición</b>
<ul style="list-style-type: none"> <li>• La sesión del usuario se cierra y se invalidan las credenciales de autenticación, impidiendo el acceso a cualquier función protegida por inicio de sesión hasta que el usuario vuelva a autenticarse.</li> <li>• El token JWT asociado con la sesión del usuario se elimina del almacenamiento local del navegador, asegurando que no pueda ser utilizado para acceder a recursos protegidos una vez que se ha cerrado la sesión.</li> <li>• Los datos de conexión a base de datos en la aplicación se pierden.</li> </ul>
<b>Manejo de situaciones anormales</b>
<b>Redirección inesperada:</b> Si el usuario es redirigido a una página incorrecta después de cerrar sesión, se revisará la configuración de redirección y se corregirá según sea necesario para asegurar una experiencia coherente y correcta.

#### 10.7.2. Requerimientos no funcionales

Identificador	Nombre
RNF 1	Modo pantalla completa
Actor	Prioridad de desarrollo
Usuario	Media
Descripción	
La funcionalidad de modo de pantalla completa permite a los usuarios expandir la aplicación web para ocupar todo el espacio disponible en el navegador, maximizando así la visualización de contenido y eliminando distracciones externas.	
Precondición	
<ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en la aplicación web.</li> <li>• El navegador web debe admitir el modo de pantalla completa y la aplicación web debe estar en ejecución y accesible para el usuario.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ El usuario activa el modo de pantalla completa mediante una acción específica, como hacer clic en un botón de pantalla completa dentro de la aplicación web.</li> </ul>	<ul style="list-style-type: none"> <li>✓ La aplicación web se expande para ocupar toda la ventana del navegador, ocultando barras de herramientas, pestañas y otros elementos que no formen parte del contenido principal.</li> <li>✓ La interfaz se ajusta automáticamente</li> </ul>

	para adaptarse al nuevo tamaño de la ventana del navegador.
<b>Postcondición</b>	
La aplicación web permanece en modo de pantalla completa hasta que el usuario lo desactive explícitamente o hasta que se cierre la aplicación web.	
<b>Manejo de situaciones anormales</b>	
<b>Interrupción inesperada:</b> Si el modo de pantalla completa se interrumpe abruptamente debido a un error del navegador o de la aplicación web, la aplicación debe volver al modo de ventana normal de forma segura, conservando los datos y el estado del usuario tanto como sea posible.	

Identificador	Nombre
RNF 2	Icono de usuario
Actor	Prioridad de desarrollo
Usuario	Media
<b>Descripción</b>	
La funcionalidad garantiza que al pasar el cursor sobre el icono de perfil de un usuario en la aplicación web, se muestre su nombre.	
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>El usuario debe estar autenticado en la aplicación web.</li> </ul>	
Entrada	Salida
✓ El usuario pasa el cursor sobre su icono de perfil	✓ Se muestra el nombre del usuario en una etiqueta emergente,
<b>Postcondición</b>	
<ul style="list-style-type: none"> <li>La funcionalidad permanece disponible mientras el usuario esté autenticado en la aplicación web.</li> </ul>	
<b>Manejo de situaciones anormales</b>	
<b>Falta de nombre de usuario:</b> Si el sistema no puede obtener un nombre de usuario o el campo de nombre de usuario está vacío, no se mostrará ningún nombre al pasar el cursor sobre el icono de perfil.	

Identificador	Nombre
RNF 3	Manual de usuario
Actor	Prioridad de desarrollo
Usuario	Media
Descripción	
<p>La aplicación debe incluir un manual de usuario dentro de la aplicación. El manual proporcionará instrucciones detalladas y claras para guiar a los usuarios en el uso de la aplicación y sus características, facilitando su aprendizaje y evitando errores de uso.</p>	
Precondición	
<ul style="list-style-type: none"> <li>• El usuario debe estar autenticado en la aplicación web.</li> <li>• La aplicación está en funcionamiento y disponible para su uso.</li> <li>• El usuario tiene la intención de utilizar la aplicación o aprender más sobre ella.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ El usuario accede al manual de usuario a través del menú lateral en el icono de "Home".</li> </ul>	<ul style="list-style-type: none"> <li>✓ El usuario recibe información clara y precisa sobre cómo utilizar la aplicación.</li> <li>✓ El usuario puede acceder a las secciones relevantes del manual para encontrar instrucciones para llevar a cabo tareas específicas.</li> </ul>
Postcondición	
<ul style="list-style-type: none"> <li>• El usuario comprende mejor la aplicación y sus funcionalidades.</li> <li>• Se reduce la probabilidad de errores de uso o malentendidos por parte del usuario.</li> <li>• El usuario es capaz de utilizar la aplicación de manera más eficiente y efectiva.</li> </ul>	
Impacto	
<ul style="list-style-type: none"> <li>• Mejora la satisfacción del usuario al poder acceder a información clara y precisa sobre el uso de la aplicación.</li> <li>• Facilita la adopción y el uso efectivo de la aplicación por parte de nuevos usuarios.</li> </ul>	

Identificador	Nombre
RNF 4	Rendimiento
Actor	Prioridad de desarrollo

Desarrolladores	Alta
<b>Descripción</b>	
<p>Este requerimiento no funcional se refiere a la capacidad del sistema para mantener un rendimiento adecuado, incluso bajo cargas de trabajo pesadas, asegurando tiempos de respuesta rápidos y una experiencia de usuario fluida.</p>	
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>• El sistema debe estar correctamente desplegado en un entorno de prueba con los recursos necesarios para manejar la carga esperada.</li> <li>• Se deben realizar pruebas unitarias y de experiencia de usuarios para evaluar el rendimiento del sistema bajo diferentes escenarios de uso.</li> </ul>	
<b>Entrada</b>	<b>Salida</b>
<ul style="list-style-type: none"> <li>✓ El sistema debe estar correctamente desplegado en un entorno de prueba con los recursos necesarios para manejar la carga esperada.</li> <li>✓ Se deben realizar pruebas unitarias para evaluar el rendimiento del sistema bajo diferentes escenarios de uso.</li> <li>✓ Se deben realizar pruebas de experiencia de usuarios para evaluar el sistema.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Tiempos de respuesta rápidos, generalmente definidos como menos de 4 segundos para completar una acción o cargar una página, incluso considerando la generación de al menos 12 archivos para descargar.</li> </ul>
<b>Postcondición</b>	
<p>El sistema proporciona una experiencia de usuario satisfactoria con tiempos de carga rápidos y respuestas ágiles, incluso bajo cargas de trabajo pesadas.</p>	
<b>Impacto</b>	
<p>Un rendimiento deficiente puede resultar en una mala experiencia del usuario, pérdida de clientes y daño a la reputación de la marca.</p>	

<b>Identificador</b>	<b>Nombre</b>
RNF 5	Seguridad
<b>Actor</b>	<b>Prioridad de desarrollo</b>
Administradores del	Alta

sistema	
<b>Descripción</b>	
Este requerimiento no funcional aborda la necesidad de implementar medidas de seguridad robustas para proteger la confidencialidad, integridad y disponibilidad de los datos del sistema y prevenir accesos no autorizados o violaciones de seguridad.	
<b>Precondición</b>	
<ul style="list-style-type: none"> <li>Se deben identificar y evaluar los posibles riesgos de seguridad del sistema.</li> </ul>	
<b>Entrada</b>	<b>Salida</b>
✓ Análisis de riesgos de seguridad que identifica posibles vulnerabilidades y amenazas al sistema.	✓ Implementación de medidas de seguridad adecuadas, como cifrado de datos, autenticación de usuarios, control de acceso y auditorías de seguridad.
<b>Postcondición</b>	
El sistema garantiza la confidencialidad, integridad y disponibilidad de los datos del usuario y protege contra accesos no autorizados y violaciones de seguridad.	
<b>Impacto</b>	
Una violación de seguridad puede resultar en la pérdida de datos confidenciales, multas regulatorias, daño a la reputación y pérdida de confianza del cliente.	

<b>Identificador</b>	<b>Nombre</b>
RNF 6	Usabilidad
<b>Actor</b>	<b>Prioridad de desarrollo</b>
Desarrolladores y equipo de diseño	Alta
<b>Descripción</b>	
Este requerimiento no funcional se refiere a la facilidad de uso y la experiencia del usuario al interactuar con la aplicación. La interfaz de usuario debe ser intuitiva y fácil de entender, lo que garantiza una curva de aprendizaje baja para los usuarios nuevos. Además, el diseño debe ser responsivo y compatible con una variedad de dispositivos y tamaños de pantalla para mejorar la accesibilidad y la comodidad del usuario.	
<b>Precondición</b>	

<ul style="list-style-type: none"> <li>• Se debe realizar un análisis de usuarios para comprender las necesidades y expectativas de los usuarios.</li> <li>• Deben seguirse las pautas de diseño de interfaz de usuario (UI) y experiencia de usuario (UX) establecidas.</li> </ul>	
Entrada	Salida
<ul style="list-style-type: none"> <li>✓ Investigación de usuarios que identifica requisitos de usabilidad y preferencias de diseño.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Una interfaz de usuario intuitiva que garantice una curva de aprendizaje baja y una experiencia positiva para el usuario.</li> <li>✓ Pruebas de usabilidad que validen la efectividad del diseño y la facilidad de uso de la aplicación.</li> </ul>
Postcondición	
<p>Los usuarios pueden interactuar fácilmente con la aplicación y realizar tareas sin problemas, lo que mejora la satisfacción y la retención del usuario.</p>	
Impacto	
<p>Una interfaz de usuario intuitiva puede aumentar la retención de usuarios y mejorar la reputación de la marca.</p>	

## 10.8. Anexo: Selección de tecnología

- ✓ **Node.js:** Node.js es un entorno de ejecución de JavaScript multiplataforma y de código abierto. Una aplicación Node.js se ejecuta en un único proceso, sin crear un nuevo hilo para cada solicitud. Node.js proporciona un conjunto de primitivas de E/S asíncronas en su biblioteca estándar que evitan que el código JavaScript se bloquee y, en general, las bibliotecas en Node.js se escriben utilizando paradigmas sin bloqueo, lo que hace que el comportamiento de bloqueo sea la excepción y no la norma.

Cuando Node.js realiza una operación de E/S, como leer desde la red, acceder a una base de datos o al sistema de archivos, en lugar de bloquear el subproceso y desperdiciar ciclos de CPU esperando, Node.js reanudará las operaciones cuando llegue la respuesta. Esto permite a Node.js manejar miles de conexiones simultáneas con un único servidor sin introducir la carga de administrar la concurrencia de subprocesos, lo que podría ser una fuente importante de errores.

Node.js tiene una ventaja única porque millones de desarrolladores frontend que escriben JavaScript para el navegador ahora pueden escribir el código del lado del servidor además del código del lado del cliente sin la necesidad de aprender un idioma completamente diferente.

En Node.js los nuevos estándares ECMAScript se pueden utilizar sin problemas, ya que no tienes que esperar a que todos tus usuarios actualicen sus navegadores: tú eres el encargado de decidir qué versión de ECMAScript usar cambiando la versión de Node.js. y también puede habilitar funciones experimentales específicas ejecutando Node.js con banderas. [24]

- ✓ **TypeScript:** TypeScript es un lenguaje de código abierto moderno mantenido y desarrollado por Microsoft. Es amado y utilizado por muchos desarrolladores de software de todo el mundo.

Básicamente, es un superconjunto de JavaScript que agrega nuevas capacidades al lenguaje. La adición más notable son las definiciones de tipos estáticos, algo que no está presente en JavaScript simple. Gracias a los tipos es posible, por ejemplo, declarar qué tipo de argumentos esperamos y qué se devuelve exactamente en nuestras funciones o cuál es la forma exacta del objeto que estamos creando. TypeScript es una herramienta realmente poderosa y abre un nuevo mundo de posibilidades en proyectos de JavaScript. Hace que nuestro código sea más seguro y robusto al evitar muchos errores incluso antes de que se envíe el código; detecta problemas durante el desarrollo del código y se integra maravillosamente con editores de código como Visual Studio Code.

Algunos de los otros beneficios de TypeScript que vale la pena mencionar son que se puede adoptar progresivamente, ayuda a que el código sea más legible y comprensible y permite a los desarrolladores utilizar funciones de lenguaje moderno mientras envían código para versiones anteriores de Node.js.

TypeScript está bien establecido en el mundo Node.js y lo utilizan muchas empresas, proyectos, herramientas y marcos de código abierto. Algunos de los ejemplos notables de proyectos de código abierto que utilizan TypeScript son:

- NestJS: marco robusto y con todas las funciones que hace que la creación de sistemas escalables y bien diseñados sea fácil y agradable
- TypeORM: excelente ORM influenciado por otras herramientas conocidas de otros lenguajes como Hibernate, Doctrine o Entity Framework
- Prisma: ORM de próxima generación que presenta un modelo de datos declarativo, migraciones generadas y consultas de bases de datos totalmente seguras.
- RxJS: biblioteca ampliamente utilizada para programación reactiva
- AdonisJS: un marco web con todas las funciones con Node.js
- FoalTS: el marco elegante de Nodejs. [25]

- ✓ **PostgreSQL:** PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas funciones que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES de la Universidad de California en Berkeley y cuenta con más de 35 años de desarrollo activo en la plataforma central.

PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características sólidas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento.

PostgreSQL viene con muchas funciones destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los datos y crear entornos tolerantes a fallas, y ayudarlo a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos. Además de ser gratuito y de código abierto, PostgreSQL es altamente extensible.

Además, PostgreSQL es altamente extensible: muchas funciones, como los índices, tienen API definidas para que pueda desarrollarlas con PostgreSQL para resolver sus desafíos.[26]

- ✓ **Angular:** Angular es una plataforma de desarrollo, construida sobre TypeScript.

Como plataforma, Angular incluye:

- Un marco basado en componentes para crear aplicaciones web escalables
- Una colección de bibliotecas bien integradas que cubren una amplia variedad de funciones, incluido el enrutamiento, la administración de formularios, la comunicación cliente-servidor y más.
- Un conjunto de herramientas de desarrollo para ayudarle a desarrollar, crear, probar y actualizar su código

Con Angular, estás aprovechando una plataforma que puede escalar desde proyectos de un solo desarrollador hasta aplicaciones de nivel empresarial. Lo mejor de todo es que el ecosistema Angular consta de un grupo diverso de más de 1,7 millones de desarrolladores, autores de bibliotecas y creadores de contenido. [27]

## 10.9. Anexo: Arquitectura del sistema

En el contexto del desarrollo de aplicaciones web, la arquitectura de software desempeña un papel fundamental en la eficiencia, seguridad y mantenibilidad de un sistema.

En la Figura 50 se muestra la arquitectura general de la aplicación, que consta de tres componentes principales: cliente, servidor y base de datos. Esta arquitectura sigue un modelo cliente-servidor, donde el cliente realiza solicitudes al servidor para interactuar con la aplicación y obtener o modificar datos.

**Cliente:** La capa de cliente representa la interfaz de usuario de la aplicación, la cual fue desarrollada utilizando el framework Angular. Esta capa es responsable de representar la interfaz gráfica con la que los usuarios interactúan.

- **Interfaz usuario:** El cliente proporciona una interfaz de usuario interactiva y receptiva.
- **Consumo de servicios Backend:** Utilizando servicios HTTP, el cliente envía solicitudes al servidor para obtener datos o para realizar operaciones.

**Servidor:** La capa del servidor está implementada en Node.js es el núcleo de la lógica de negocio y la manipulación de datos de la aplicación. Esta capa se encarga de procesar solicitudes del cliente, interactuar con la base de datos y proporcionar las respuestas.

- **Lógica:** Implementa la lógica de la aplicación, procesando las solicitudes del cliente de acuerdo con las reglas y requisitos específicos del sistema.
- **Conexión con las Base de Datos:** Utilizando la librería 'pg' para interactuar con la base de datos PostgreSQL, para la ejecución de consultas SQL.
- **Seguridad y Autenticación:** Implementa la capa de seguridad y autenticación utilizando JSON Web Tokens (JWT) para verificar la identidad de los usuarios, controlar el acceso a recursos protegidos y mantener sesiones seguras.

**Autorización:** La capa de autorización basada en JSON Web Tokens (JWT) proporciona un mecanismo seguro para controlar el acceso a recursos protegidos dentro de la aplicación.

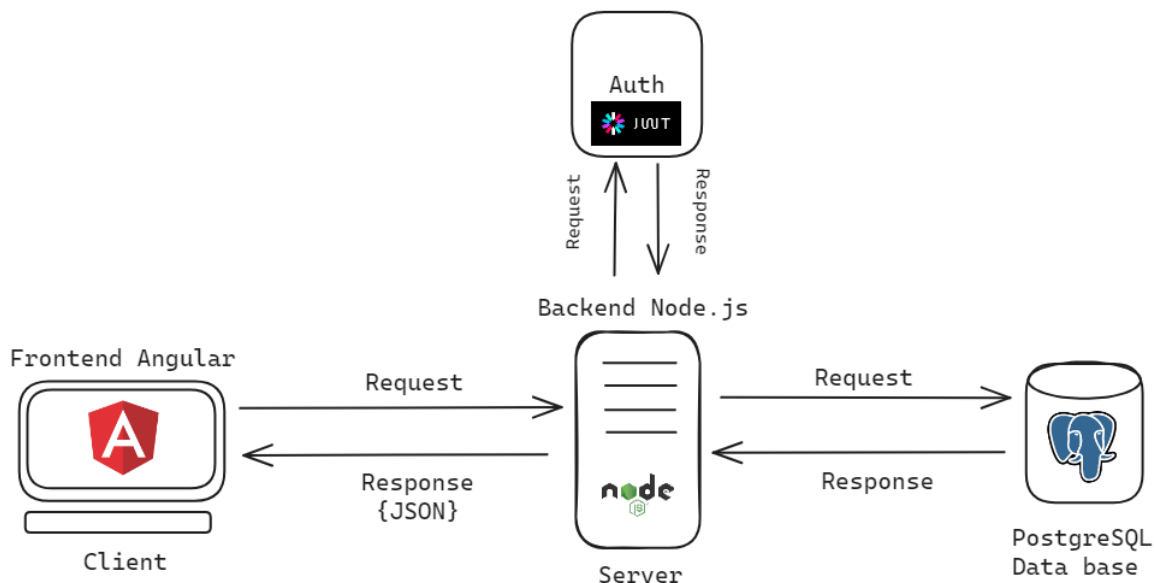
- **Generación de Tokens:** Genera tokens JWT cuando un usuario inicia sesión correctamente.

- **Verificación de Tokens:** Valida y verifica los tokens JWT incluidos en la solicitudes entrantes del cliente para garantizar la autenticidad y autorizacion de la solicitud.
- **Control de acceso:** Utiliza la información del token para determinar si un usuario tiene permiso para acceder a ciertos recursos o realizar ciertas acciones dentro de la aplicación
- **Tiempo de expiración:** Getiona el tiempo de expiración de los tokes JWT para garantizar la seguridad y evitar el uso indebido de tokens caducados.

**Base de datos:** la capa de base de datos utiliza PostgreSQL como sistema de gestión de base de datos relacional. Esta capa almacena y gestiona los datos de la aplicación de manera persistente, proporcionando acceso seguro y eficiente a la información.

- **Almacenamiento de datos:** La base de datos PostgreSQL almacena todos los datos relevantes, incluyendo informacion de usuarios.
- **Interaccion con el servidor:** El servidor backend interactúa con la base de datos mientas consulatas SQL, utilizando librería 'pg' en Node.js para ejecturar operaciones de lectura y escritura.

Figura 50. Diseño de la arquitectura del sistema



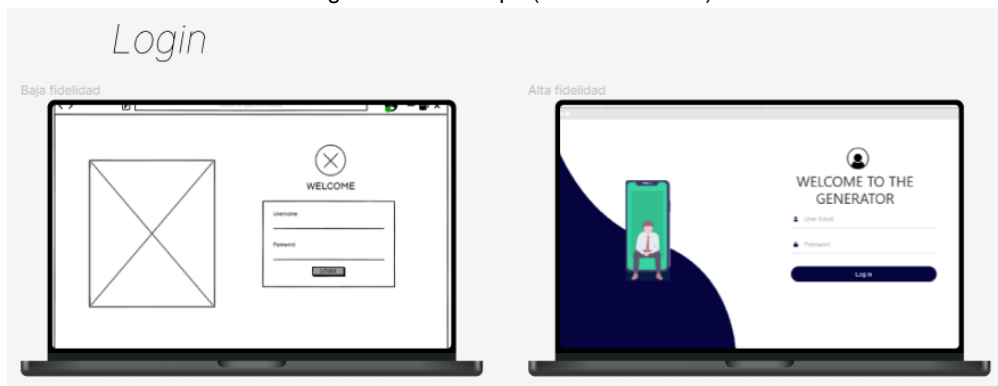
Fuente: Autores

## 10.10. Anexo: Visualización de la interfaz de Usuario (Mockups)

Este anexo presenta una serie de mockups que representan la interfaz de usuario propuesta para la meta herramienta de generación de código TypeScript en plataformas backend. Estos mockups han sido creados utilizando las herramientas Figma y Balsamiq, lo que permitió ofrecer una representación visual detallada de la experiencia del usuario al interactuar con la herramienta. Cada mockup incluye elementos gráficos como pantallas, botones, formularios y menús, con el fin de mostrar la disposición y funcionalidades de la interfaz. Estos mockups son una herramienta útil para validar y comunicar el diseño de la interfaz antes de su implementación.

- **Mockups (Iniciar Sesión)**

Figura 51. Mockups (Inicio de sesión)



Fuente: Autores

- **Mockups (Menú Principal)**

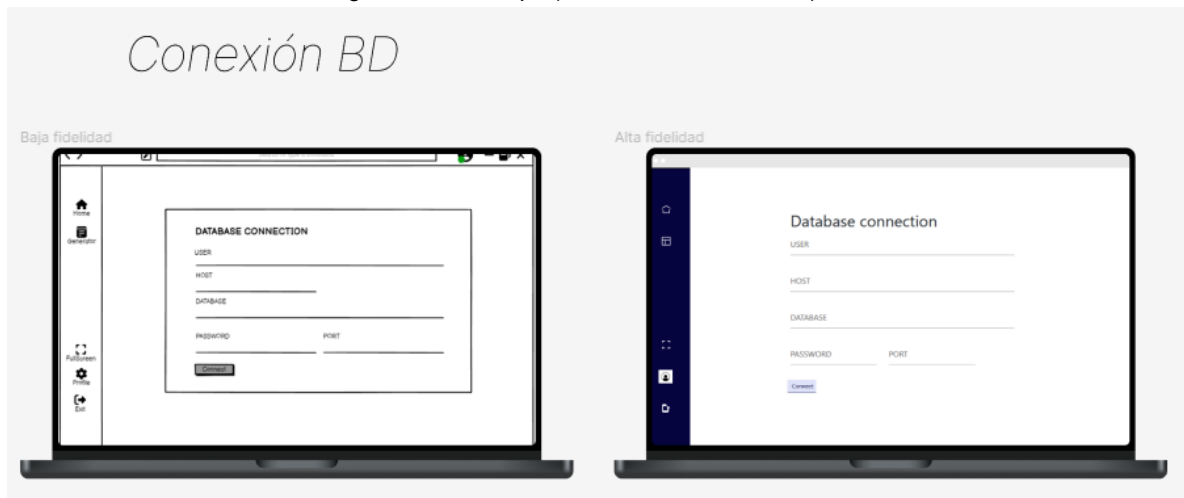
Figura 52. Mockups (Menú Principal)



Fuente: Autores

- **Mockups (Formulario de conexión)**

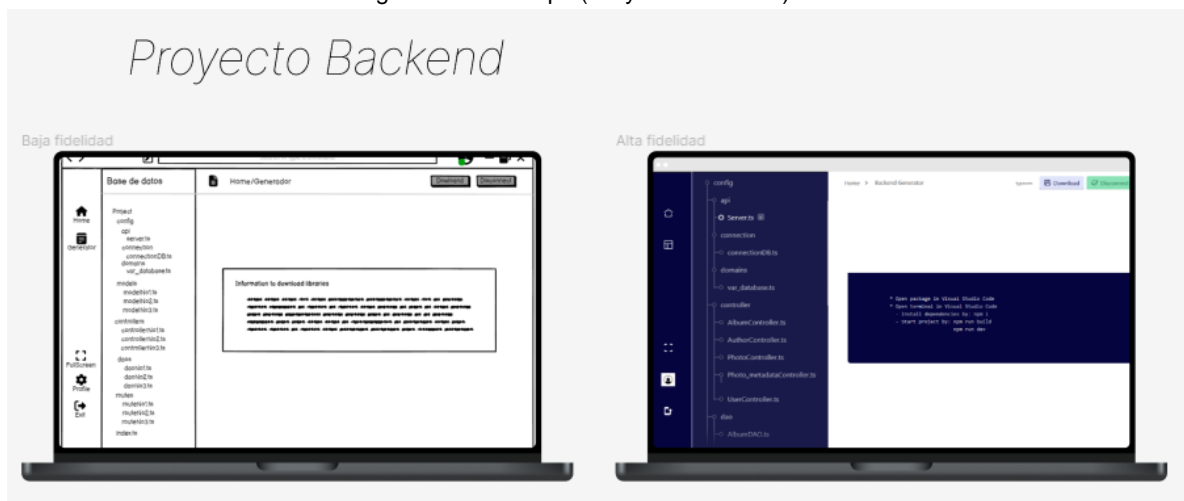
Figura 53. Mockups (Formulario de conexión)



Fuente: Autores

- **Mockups (Proyecto Backend)**

Figura 54. Mockups (Proyecto Backend)



Fuente: Autores

- **Mockups (Visualización de código)**

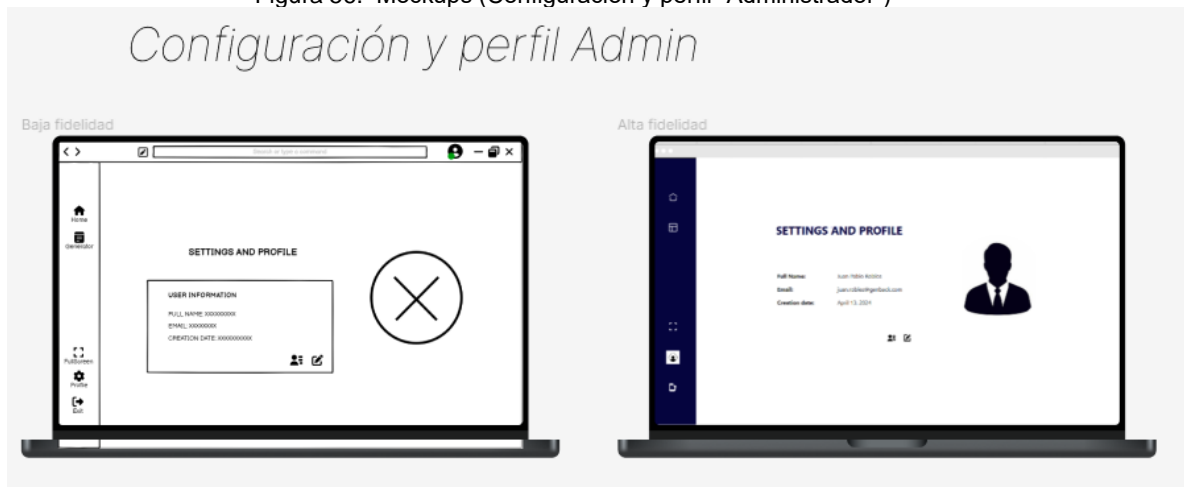
Figura 55. Mockups (Visualización de código)



Fuente: Autores

- **Mockups (Configuración y perfil “Administrador”)**

Figura 56. Mockups (Configuración y perfil “Administrador”)



Fuente: Autores

- **Mockups (Configuración y perfil “Usuario”)**

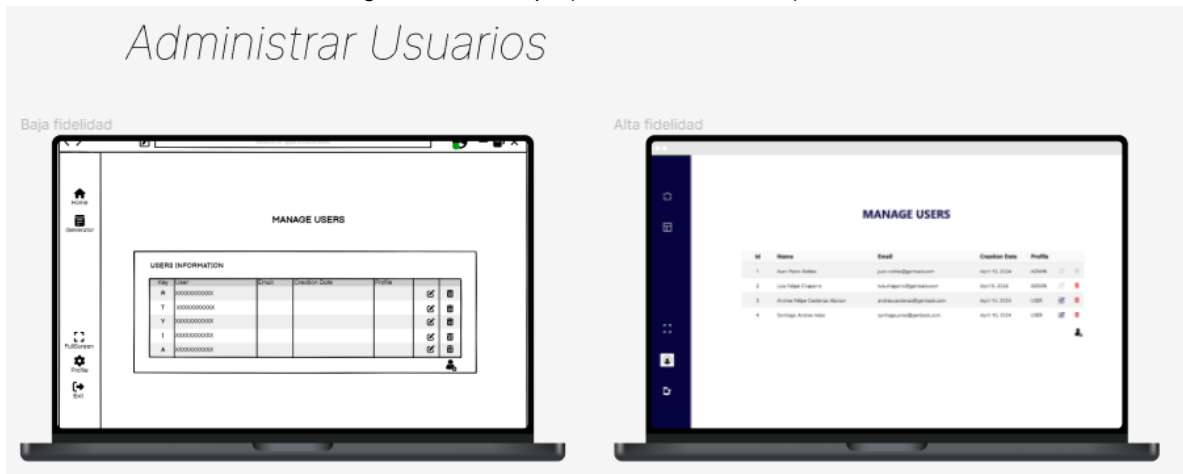
Figura 57. Mockups (Configuración y perfil “Usuario”)



Fuente: Autores

- **Mockups (Configuración y perfil “Usuario”)**

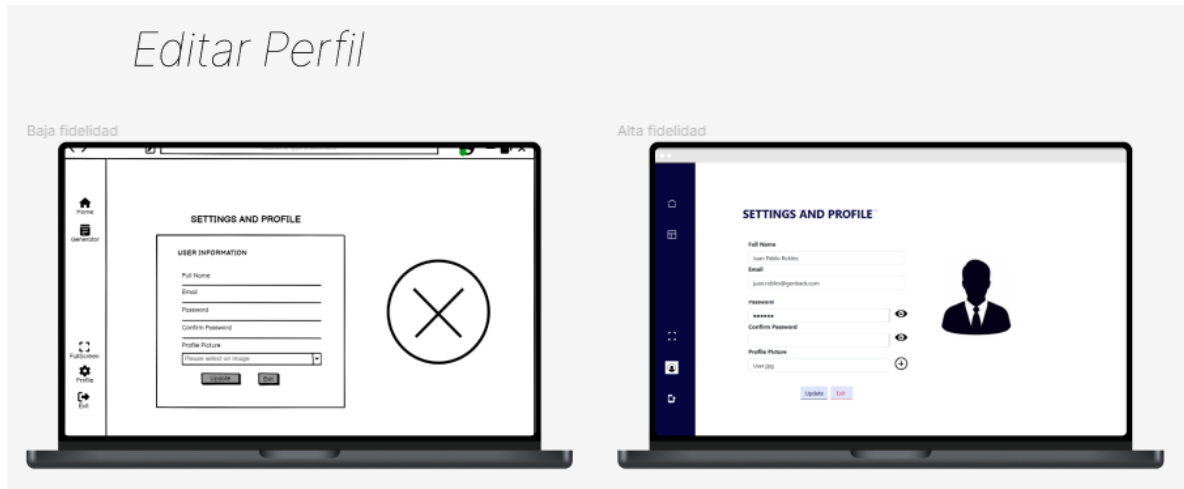
Figura 58. Mockups (Administrar usuarios)



Fuente: Autores

- **Mockups (Editar perfil)**

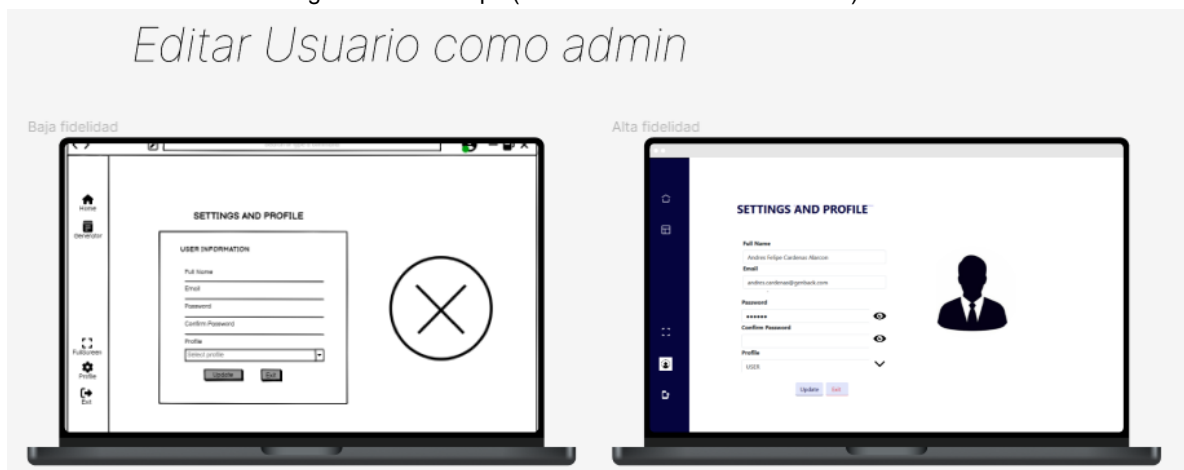
Figura 59. Mockups (Editar perfil)



Fuente: Autores

- **Mockups (Editar usuarios “Administrador”)**

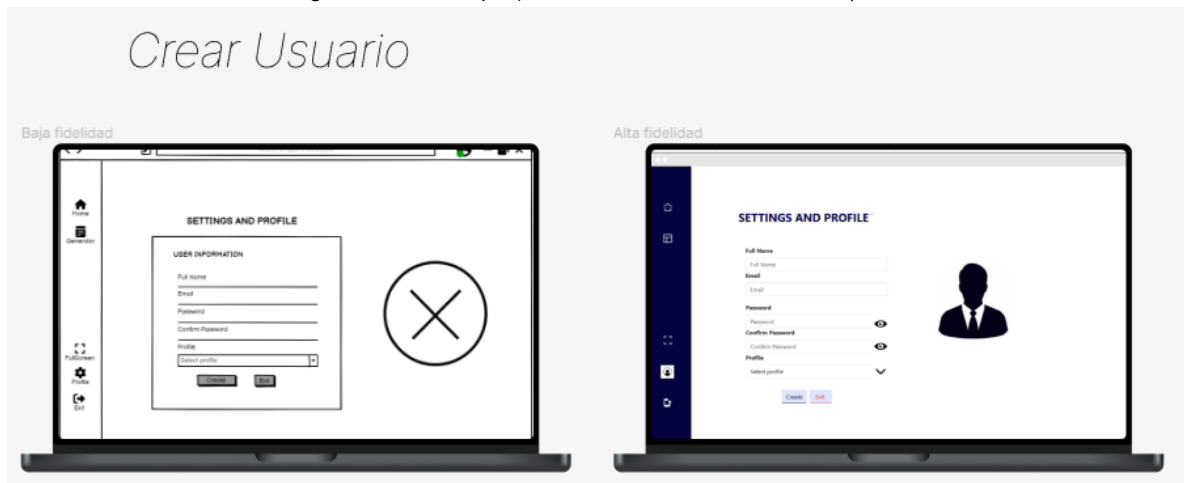
Figura 60. Mockups (Editar usuarios “Administrador”)



Fuente: Autores

- **Mockups (Crear usuarios “Administrador”)**

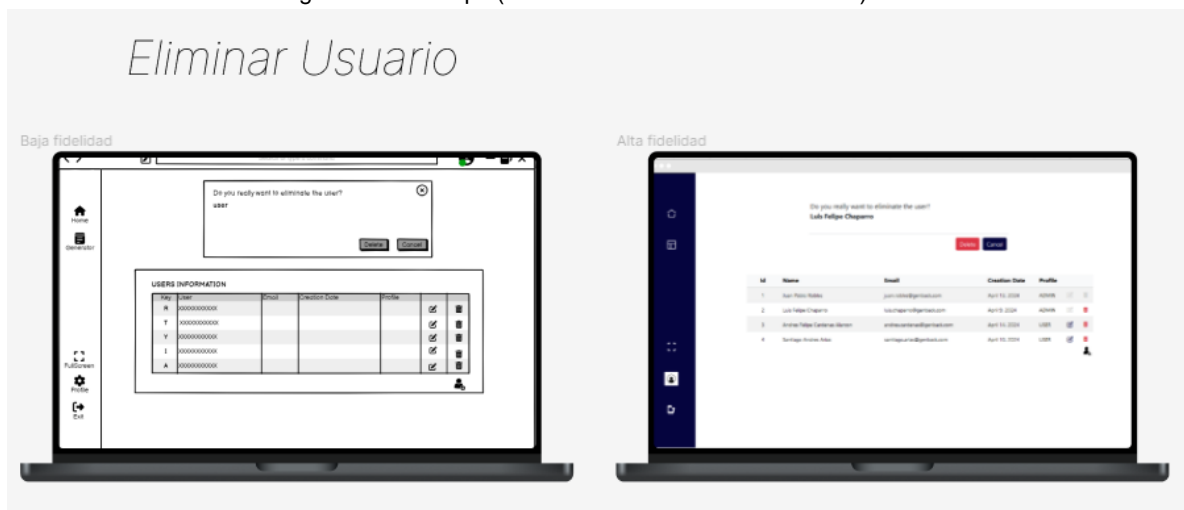
Figura 61. Mockups (Crear usuarios “Administrador”)



Fuente: Autores

- **Mockups (Eliminar usuarios “Administrador”)**

Figura 62. Mockups (Eliminar usuarios “Administrador”)



Fuente: Autores

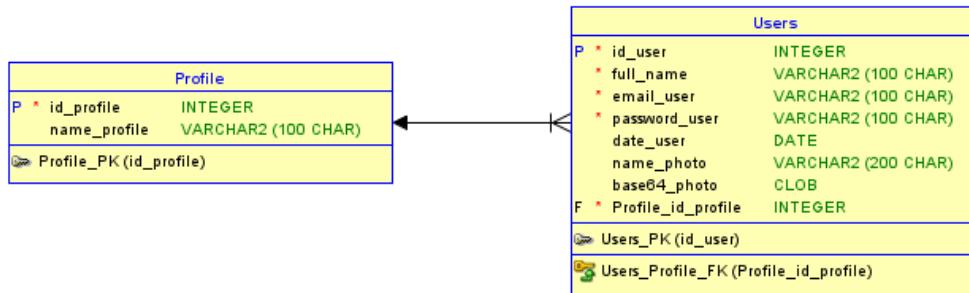
El siguiente enlace contiene los mockups de la herramienta de generación de código TypeScript para plataformas backend, creados en Figma. Estos mockups ofrecen una representación visual detallada de la interfaz de usuario propuesta. Puedes acceder al enlace para visualizar cada uno de los mockups:

<https://www.figma.com/file/qPytrwUv6Wki21hFWI8qVC/MockupsGeneradorBackend?type=design&node-id=0%3A1&mode=design&t=PrBBO9IC4su3kJLF-1>

### 10.11. Anexo: Diagrama de base de datos

Este anexo muestra el diagrama de la base de datos utilizada para el generador de backend. En la Figura 63 se representan las tablas users y profile, así como la relación entre ellas. El diseño se ha optimizado para que sea sencillo, eficiente y adecuado para las necesidades del proyecto, sin almacenar información innecesaria.

Figura 63. Diagrama de base datos

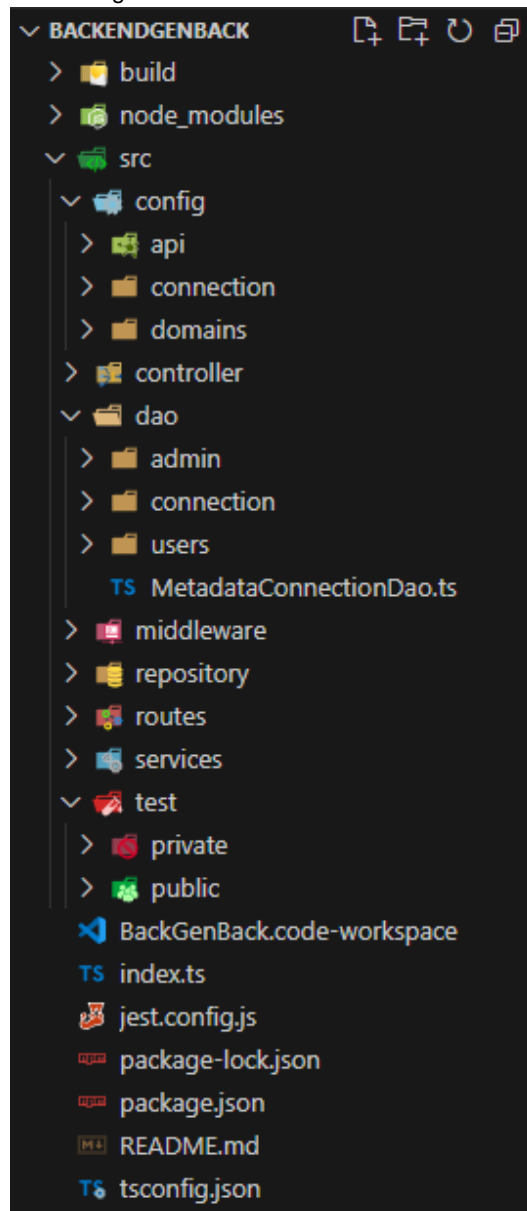


Fuente: Autores

## 10.12. Anexo: Desarrollo Backend

Este anexo detalla la estructura del backend, cuidadosamente diseñada para asegurar una organización eficiente y modular del código. En la Figura 64 se puede apreciar la distribución de los distintos componentes que trabajan en conjunto para ofrecer una funcionalidad sólida y escalable. Cada uno de estos componentes tiene un propósito definido y se ha configurado de forma que sea comprensible y fácil de mantener.

Figura 64. Estructura del backend



Fuente: Autores

- **Directorio Raíz:**

Este directorio almacena los archivos de configuración clave, como la configuración del servidor, package.json, package-lock.json, tsconfig.json, y la documentación necesaria para el despliegue del proyecto backend. Estos archivos son esenciales para la inicialización y el correcto funcionamiento de la aplicación generada.

- **Directorio de Controladores:**

Contiene los controladores de la aplicación, que son responsables de procesar las solicitudes HTTP entrantes, interactuar con el modelo de datos y devolver respuestas adecuadas al usuario.

- **Directorio de Modelos:**

Contiene los modelos de datos de la aplicación, que definen la estructura y la lógica para interactuar con la base de datos.

- **Directorio de DAO (Data Access Objects):**

Este directorio contiene objetos responsables de interactuar directamente con la base de datos para realizar operaciones de lectura y escritura. Simplifica la gestión de datos y promueve la reutilización del código.

- **Directorio de Middleware:**

Aquí se encuentran los middlewares de la aplicación, que son funciones que se ejecutan antes o después de que se manejen las solicitudes HTTP y que pueden realizar tareas como la autenticación, la validación de datos o el registro de solicitudes.

- **Directorio de Repositorios:**

Contiene los repositorios, que actúan como una capa intermedia entre los DAO y los servicios de la aplicación. Simplifica el acceso a los datos y promueve una estructura modular y mantenible.

- **Directorio de Rutas:**

Aquí se definen las URL de solicitud y se las asocia con las funciones controladoras correspondientes. Facilita la gestión de solicitudes HTTP y promueve una arquitectura limpia y escalable.

- **Directorio de Servicios:**

Contiene los servicios de la aplicación, que encapsulan la lógica de negocio y proporcionan una interfaz para que los controladores accedan a ella. Los servicios pueden realizar operaciones complejas que involucren múltiples modelos o componentes del sistema.

- **Directorio de Utilidades:**

Alberga funciones y utilidades auxiliares que se utilizan en todo el proyecto, como funciones de ayuda, funciones de formato de datos o funciones de manipulación de archivos.

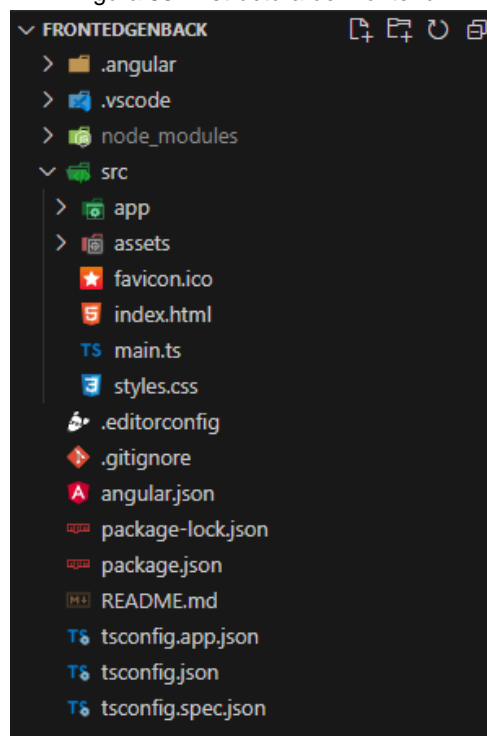
- **Directorio de Pruebas:**

Contiene los archivos de prueba para garantizar el correcto funcionamiento de las diferentes partes del proyecto backend. Estas pruebas pueden incluir pruebas unitarias, pruebas de integración y pruebas de extremo a extremo.

### 10.13. Anexo: Desarrollo Frontend

Este anexo proporciona una visión detallada de la estructura del proyecto frontend, desarrollado con Angular. La organización y disposición de los diferentes componentes del frontend se presentan de manera clara y sistemática. Desde la estructura de directorios hasta la arquitectura de los componentes individuales, este anexo ofrece una comprensión completa de cómo se ha diseñado y organizado el proyecto. Los componentes principales, como módulos, componentes, servicios y rutas, se describen en detalle, resaltando su función y relación dentro del ecosistema de la aplicación.

Figura 65. Estructura del frontend



Fuente: Autores

- **Directorio raíz:**

Este directorio contiene los archivos de configuración principales, como angular.json, package.json, tsconfig.json, entre otros. Aquí se definen las dependencias del proyecto, scripts de construcción, y configuraciones del entorno de desarrollo.

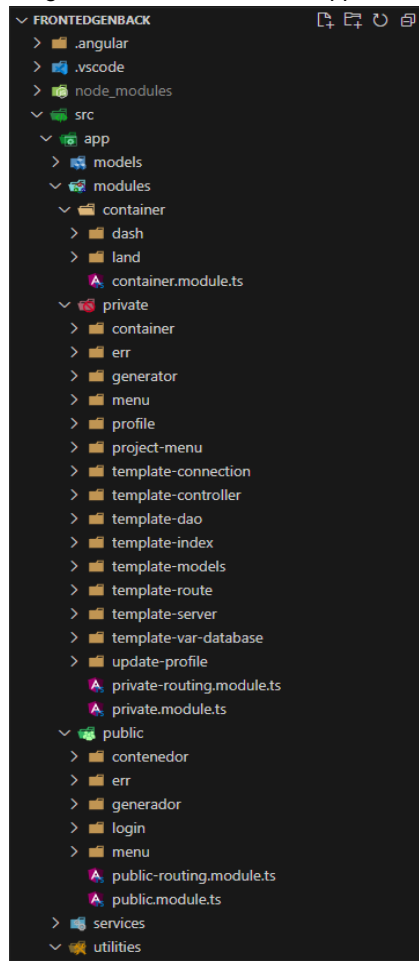
- **Directorio de src:**

Este es el directorio principal donde se encuentra el código fuente de la aplicación Angular. Contiene subdirectorios y archivos esenciales para el desarrollo de la aplicación.

- **Directorio de app:**

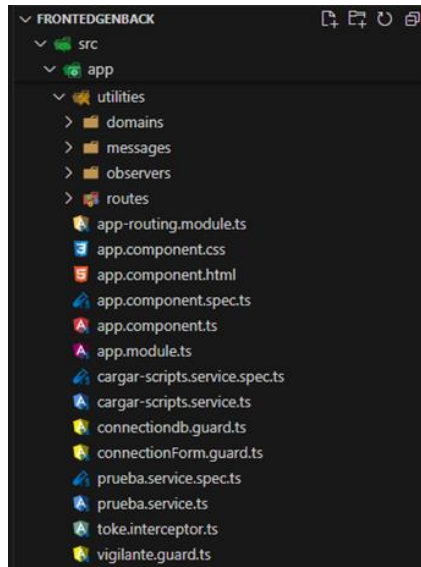
Este directorio contiene la lógica principal de la aplicación. En las Figura 66 y Figura 67 se encuentran los módulos, componentes, servicios y otros elementos necesarios para construir la interfaz de usuario y la lógica de la aplicación.

Figura 66. Directorio de la App No.1



Fuente: Autores

Figura 67. Directorio de la App No.2

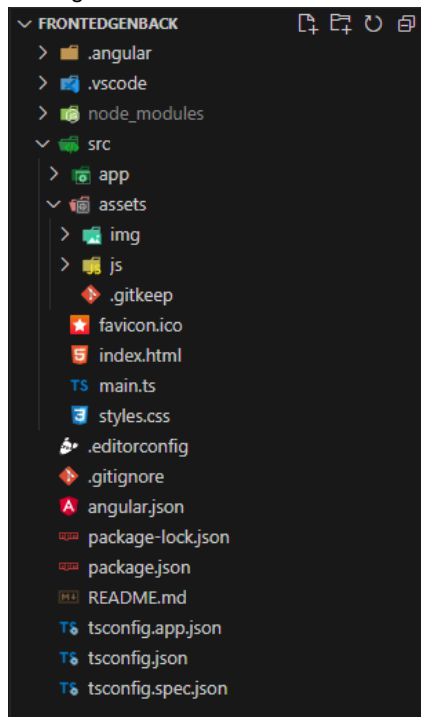


Fuente: Autores

- **Directorio de assets:**

En la Figura 68 se representa los archivos estáticos de la aplicación, como imágenes, fuentes, iconos, y otros recursos que serán utilizados por la interfaz de usuario.

Figura 68. Directorio de Assets

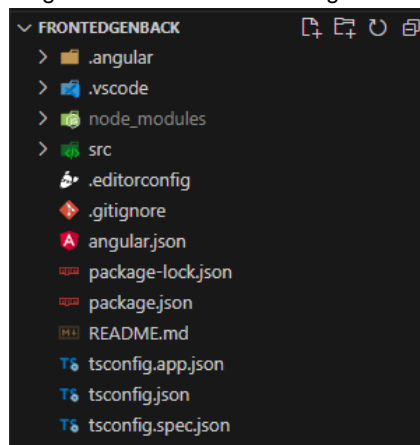


Fuente: Autores

- **Archivos de configuración:**

En la Figura 69 se pueden observar otros archivos de configuración adicionales, como angular.json, package.json, package-lock.json, tsconfig.json, entre otros, que pueden variar dependiendo de las necesidades y configuraciones específicas del proyecto.

Figura 69. Archivos de configuración



Fuente: Autores

## 10.14. Anexo: Pruebas Unitarias Back-end

En el siguiente anexo se presentan las pruebas unitarias realizadas en el backend de la aplicación utilizando la librería Jest. Jest es un framework de pruebas de desarrollo, se utiliza comúnmente para realizar pruebas unitarias en proyectos JavaScript, especialmente en entornos Node.js. [28]

A continuación, se muestran las evidencias de las pruebas realizadas a cada uno de los endpoints.

- La Figura 70 muestra las pruebas del endpoint de inicio de sesión (“/login”).

Figura 70. Test del endpoint “/login”

```
POST /api/public/user/login 200 54.016 ms - 216
PASS src/test/public/userlogin.spec.ts
Public Routes
  ✓ should respond with 200 status and valid response for POST /api/public/user/login (76 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.2 s, estimated 3 s
```

Fuente: Autores

- La Figura 71 muestra las pruebas del endpoint de listar usuarios (“/admin/list”).

Figura 71. Test del endpoint “/admin/list”

```
GET /api/private/admin/list 200 274.704 ms - 597
PASS src/test/private/adminlist.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for GET /api/private/admin/list (323 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.724 s, estimated 3 s
```

Fuente: Autores

- La Figura 72 muestra las pruebas del endpoint de crear usuario como administrador (“/admin/create”).

Figura 72. Test del endpoint “/admin/create”

```
POST /api/private/admin/create 200 87.141 ms - 27
PASS src/test/private/admincreate.spec.ts
Public Routes
  ✓ should respond with 200 status and valid response for POST /api/private/admin/create (113 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.565 s
```

Fuente: Autores

- La Figura 73 muestra las pruebas del endpoint de cargar información de usuario por ID como administrador (“/admin/find/:idUser”).

Figura 73. Test del endpoint “/admin/find/:idUser”

```
GET /api/private/admin/find/15 200 68.592 ms - 114
PASS src/test/private/adminfind.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for GET /api/private/admin/find/:idUser (98 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 2.479 s
```

Fuente: Autores

- La Figura 74 muestra las pruebas del endpoint de actualizar información de usuario como administrador (“/admin/update/:idUser”).

Figura 74. Test del endpoint “/admin/update/:idUser”

```
PUT /api/private/admin/update/18 200 249.582 ms - 32
PASS src/test/private/adminupdate.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for PUT /api/private/admin/update/:idUser (290 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 2.732 s
```

Fuente: Autores

- La Figura 75 muestra las pruebas del endpoint de eliminar usuario como administrador (“/admin/delete/:idUser”).

Figura 75. Test del endpoint “/admin/delete/:idUser”

```
DELETE /api/private/admin/delete/18 200 59.614 ms - 39
PASS src/test/private/admindelete.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for DELETE /api/private/admin/delete/:idUser (87 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 2.641 s
```

Fuente: Autores

- La Figura 76 muestra las pruebas del endpoint de cargar información de usuario (“/user/find/:idUser”).

Figura 76. Test del endpoint “/user/find/:idUser”

```
GET /api/private/user/find/1 200 57.845 ms - 21069
PASS src/test/private/userfind.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for GET /api/private/user/find/:idUser (84 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 2.401 s
```

Fuente: Autores

- La Figura 77 muestra las pruebas del endpoint de actualizar información de usuario (“/user/update/:idUser”).

Figura 77. Test del endpoint “/user/update/:idUser”

```
PUT /api/private/user/update/15 200 81.182 ms - 32
PASS src/test/private/user/userupdate.spec.ts
  Private Routes
    ✓ should respond with 200 status and valid response for PUT /api/private/user/update/:idUser (105 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.255 s, estimated 3 s
```

Fuente: Autores

- La Figura 78 muestra las pruebas del endpoint para establecer conexión con la base de datos para generar código (“/connection”).

Figura 78. Test del endpoint “/connection”

```
POST /api/private/connection 200 62.896 ms - 177
PASS src/test/private/generator/connection.spec.ts
  Private Routes
    ✓ should respond with 200 status and valid response for POST /api/private/connection (92 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.452 s
```

Fuente: Autores

- La Figura 79 muestra las pruebas del endpoint para generar el código del archivo Server.ts (“/server”).

Figura 79. Test del endpoint “/server”

```
GET /api/private/server 200 43.943 ms - 1653
PASS src/test/private/generator/server.spec.ts
  Private Routes
    ✓ should respond with 200 status and valid response for POST /api/private/connection (94 ms)
    ✓ should respond with 200 status and valid response for GET /api/private/server (49 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        2.515 s, estimated 3 s
```

Fuente: Autores

- La Figura 80 muestra las pruebas del endpoint para generar el código del archivo de cada modelo (“/models”).

Figura 80. Test del endpoint “/models”

```
GET /api/private/models 200 77.269 ms - 6381
PASS src/test/private/generator/model.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (96 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/models (83 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        2.524 s
```

Fuente: Autores

- La Figura 81 muestra las pruebas del endpoint para generar el código del archivo connectionDB.ts (“/connectiondb”).

Figura 81. Test del endpoint “/connectiondb”

```
GET /api/private/connectiondb 200 8.219 ms - 210
PASS src/test/private/generator/connectiondb.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (86 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/connectiondb (39 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        2.52 s, estimated 3 s
```

Fuente: Autores

- La Figura 82 muestra las pruebas del endpoint para generar el código del archivo var\_database.ts (“/vardatabase”).

Figura 82. Test del endpoint “/vardatabase”

```
GET /api/private/vardatabase 200 17.099 ms - 637
PASS src/test/private/generator/vardatabase.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (91 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/vardatabase (22 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        2.678 s
```

Fuente: Autores

- La Figura 83 muestra las pruebas del endpoint para generar el código del archivo de cada ruta (“/routes”).

Figura 83. Test del endpoint “/routes”

```
GET /api/private/routes 200 45.245 ms - 4647
PASS src/test/private/generator/routes.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (90 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/routes (50 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.54 s
```

Fuente: Autores

- La Figura 84 muestra las pruebas del endpoint para generar el código del archivo de cada controlador (“/controller”).

Figura 84. Test del endpoint “/controller”

```
GET /api/private/controller 200 47.240 ms - 8193
PASS src/test/private/generator/controller.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (93 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/controller (56 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.479 s
```

Fuente: Autores

- La Figura 85 muestra las pruebas del endpoint para generar el código del archivo de cada DAO (“/dao”).

Figura 85. Test del endpoint “/dao”

```
GET /api/private/dao 200 69.140 ms - 12338
PASS src/test/private/generator/dao.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (103 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/dao (74 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.655 s
```

Fuente: Autores

- La Figura 86 muestra las pruebas del endpoint para generar el código del archivo index.ts (“/index”).

Figura 86. Test del endpoint “/index”

```
GET /api/private/index 200 8.039 ms - 158
PASS src/test/private/generator/index.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (92 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/index (15 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.527 s
```

Fuente: Autores

- La Figura 87 muestra las pruebas del endpoint para generar el código del archivo package.json (“/package”).

Figura 87. Test del endpoint “/package”

```
GET /api/private/package 200 9.034 ms - 1053
PASS src/test/private/generator/package.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (104 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/package (15 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.479 s
```

Fuente: Autores

- La Figura 88 muestra las pruebas del endpoint para generar el código del archivo package-lock.json (“/package-lock”).

Figura 88. Test del endpoint “/package-lock”

```
GET /api/private/package-lock 200 15.703 ms - 121771
PASS src/test/private/generator/package-lock.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (215 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/package-lock (26 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 2.869 s
```

Fuente: Autores

- La Figura 89 muestra las pruebas del endpoint para generar el código del archivo tsconfig.json (“/tsconfig”).

Figura 89. Test del endpoint "/tsconfig"

```
GET /api/private/tsconfig 200 7.840 ms - 12230
PASS src/test/private/generator/tsconfig.spec.ts
Private Routes
  ✓ should respond with 200 status and valid response for POST /api/private/connection (90 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/tsconfig (13 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        2.433 s
```

Fuente: Autores

- En la Figura 90 se lleva a cabo la prueba para la construcción del proyecto que implica la conexión a la base de datos para la generación del código backend. Esta prueba demuestra que en esta fase de construcción se han utilizado 3.101 segundos, estimando que debería completarse en aproximadamente 4 segundos.

Figura 90. Prueba de construcción de proyecto del código backend generado

```
PASS src/test/private/generator/downloadproject.spec.ts
Build project with backend code
  ✓ should respond with 200 status and valid response for POST /api/private/connection (217 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/server (37 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/connectiondb (17 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/vardatabase (20 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/controller (59 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/dao (58 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/models (48 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/routes (37 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/index (12 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/package-lock (19 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/package (13 ms)
  ✓ should respond with 200 status and valid response for GET /api/private/tsconfig (12 ms)

Test Suites: 1 passed, 1 total
Tests:       12 passed, 12 total
Snapshots:  0 total
Time:        3.101 s, estimated 4 s
```

Fuente: Autores

Para concluir, en la Tabla 7 se evidencian los resultados de las pruebas unitarias realizadas, todas con un éxito absoluto, alcanzando un rendimiento del 100% en cada una.

Tabla 7. Tabla de pruebas unitarias

#	Endpoint	Estatus	Time
1	/login	passed	76 ms
2	/admin/list	passed	323 ms
3	/admin/create	passed	113 ms
4	/admin/find/:idUser	passed	98 ms
#	Endpoint	Estatus	Time
5	/admin/update/:idUser	passed	290 ms

6	/admin/delete/:idUser	passed	87 ms
7	/user/find/:idUser	passed	84 ms
8	/user/update/:idUser	passed	105 ms
9	/connection	passed	92 ms
10	/server	passed	49 ms
11	/models	passed	83 ms
12	/connectiondb	passed	39 ms
13	/vardatabase	passed	22 ms
14	/routes	passed	50 ms
15	/controller	passed	56 ms
16	/dao	passed	74 ms
17	/index	passed	15 ms
18	/package	passed	15 ms
19	/package-lock	passed	26 ms
20	/tsconfig	passed	13 ms
21	/downloadproject	passed	3.101 s

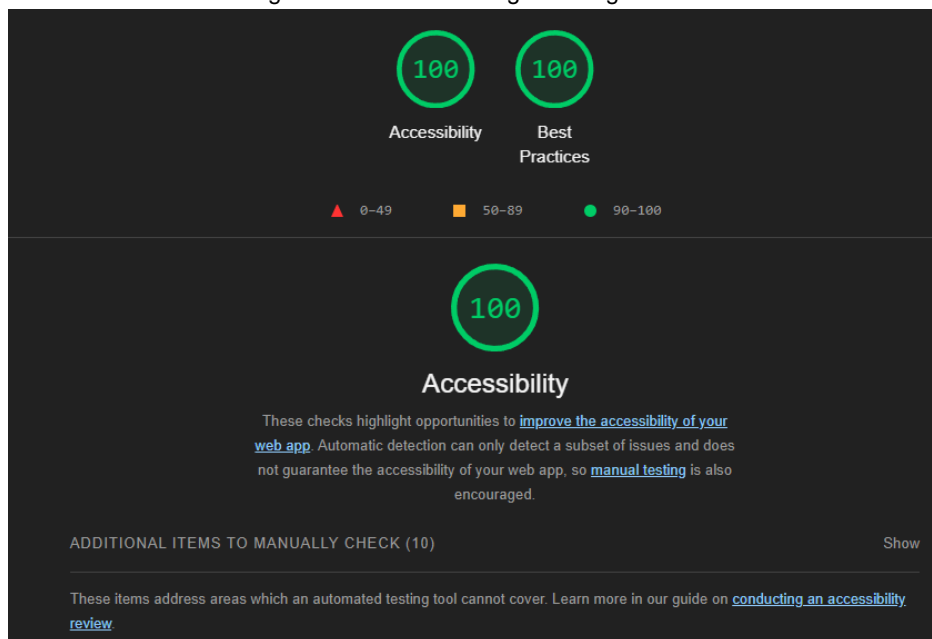
Fuente: Autor

## 10.15. Anexo: Pruebas de accesibilidad y mejores prácticas del Frontend

Este anexo presenta una visión general de las pruebas de accesibilidad y mejores prácticas implementadas en el desarrollo frontend de esta meta herramienta desplegada en un entorno de prueba. Para asegurar un sitio web accesible y eficiente, se llevaron a cabo diversas evaluaciones utilizando herramientas y extensiones especializadas, como SEOquake Analytics [29] y Lighthouse [30].

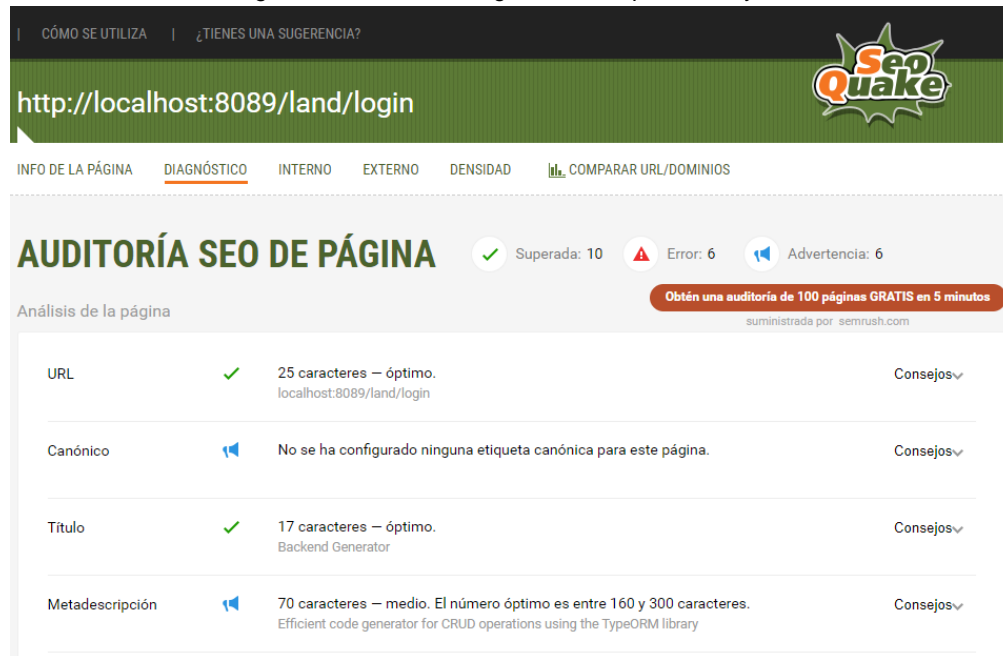
La Figura 91 muestra una evaluación de accesibilidad y mejores prácticas del “/login” de la meta herramienta, garantizando una experiencia de usuario agradable. En la Figura 92, se presenta una evaluación de SEO (Search Engine Optimization) [31] para analizar la visibilidad del sitio web en motores de búsqueda y optimizar su posicionamiento mediante ajustes en contenido, metaetiquetas, estructura y velocidad de carga.

Figura 91. Prueba a “/login” en Lighthouse



Fuente: Lighthouse

Figura 92. Prueba a “/login” en SEOquake Analytics

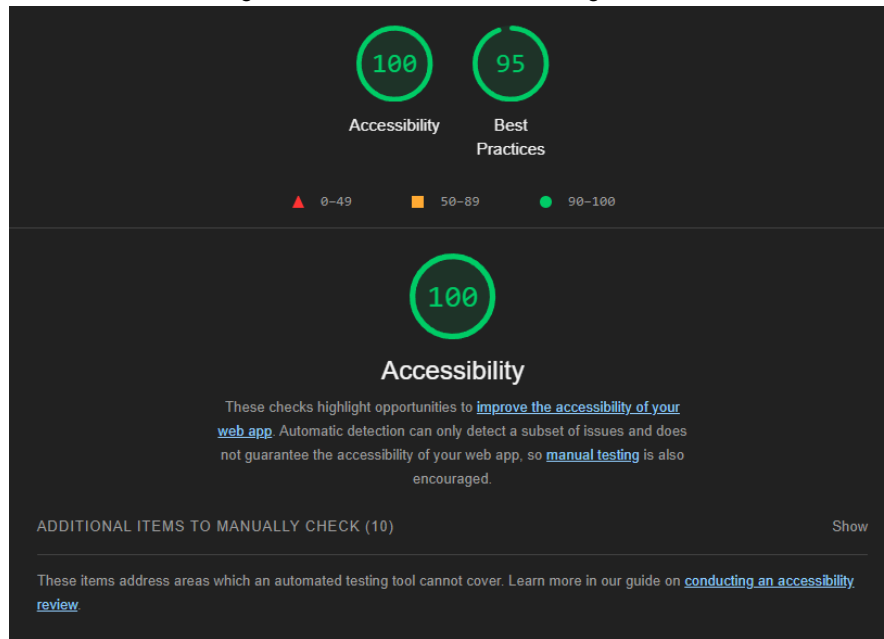


Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

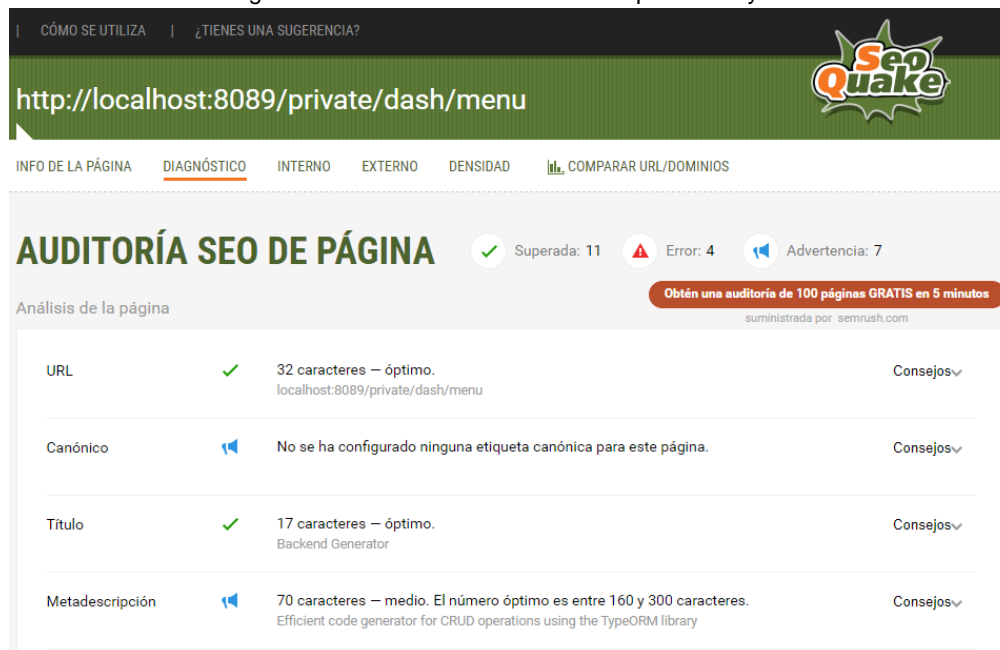
En la Figura 93 se muestra la prueba de Lighthouse, mientras que en la Figura 94 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/menu” de la meta herramienta.

Figura 93. Prueba a "/menu" en Lighthouse



Fuente: Lighthouse

Figura 94. Prueba a "/menu" en SEOquake Analytics



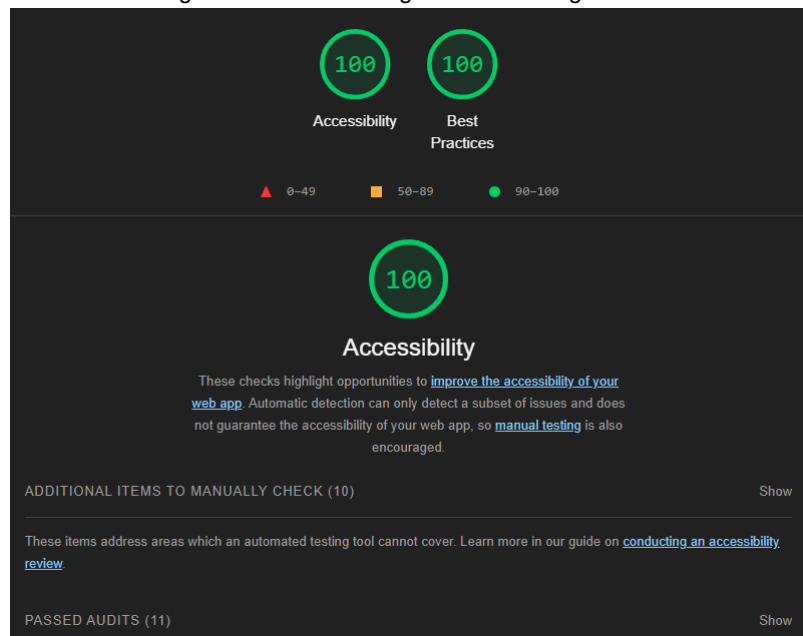
Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de

una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

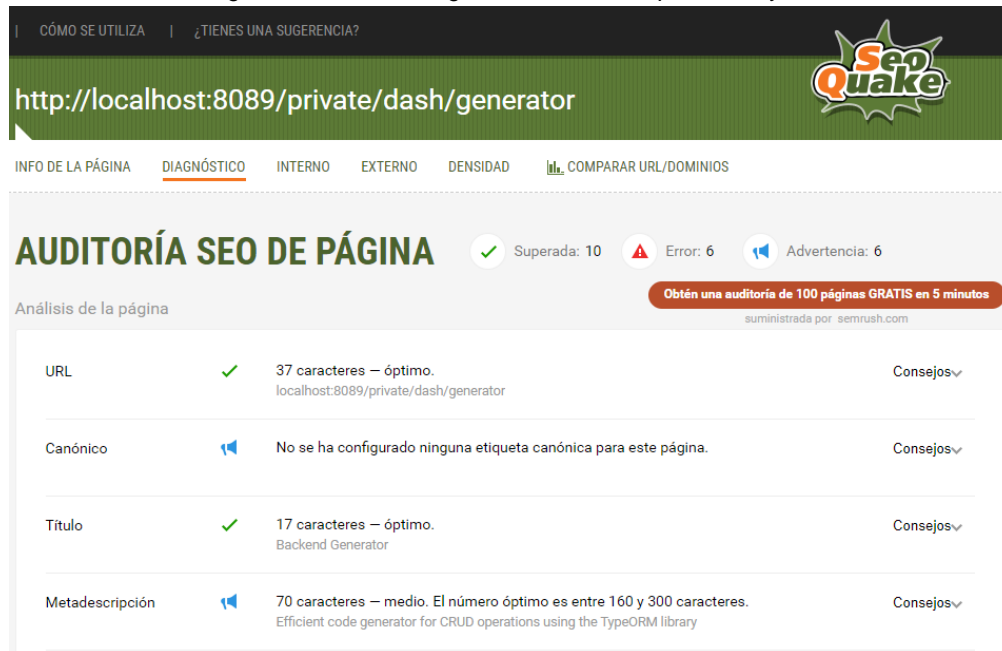
En la Figura 95 se muestra la prueba de Lighthouse, mientras que en la Figura 96 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/generator” de la meta herramienta.

Figura 95. Prueba a “/generator” en Lighthouse



Fuente: Lighthouse

Figura 96. Prueba a “/generator” en SEOquake Analytics

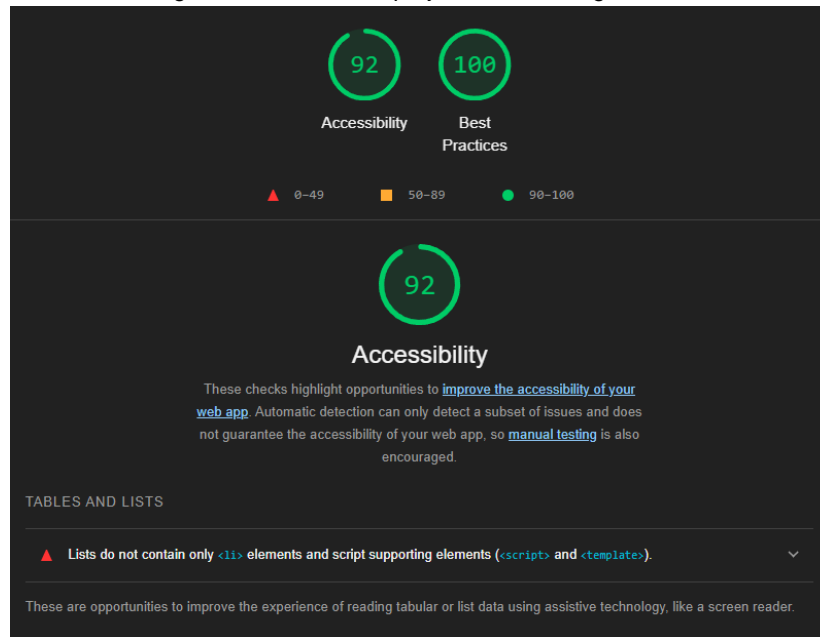


Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

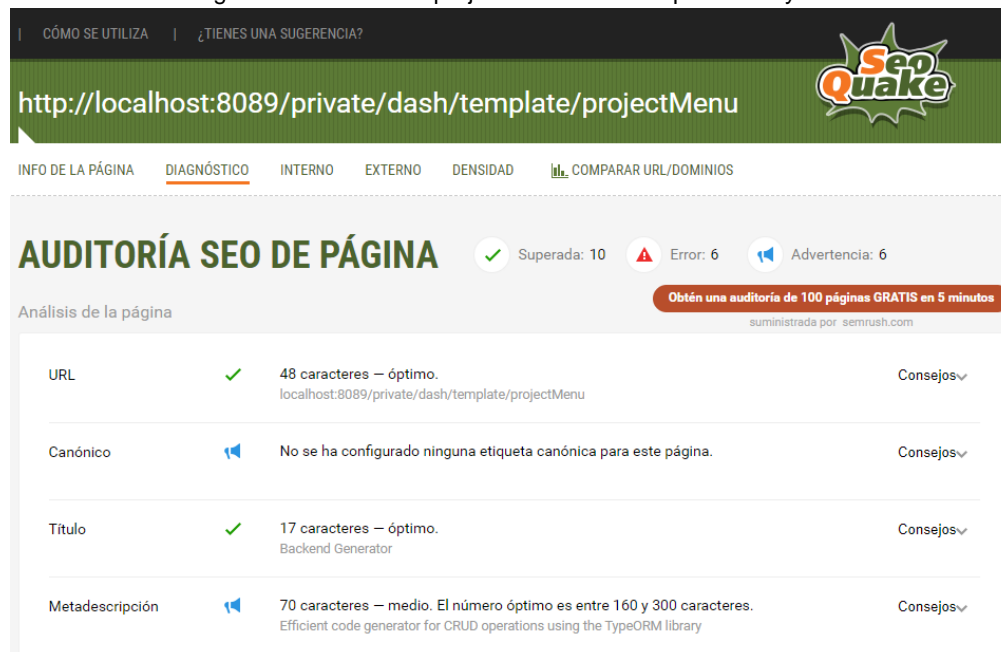
En la Figura 97 se muestra la prueba de Lighthouse, mientras que en la Figura 98 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/projectMenu” de la meta herramienta.

Figura 97. Prueba a “/projectMenu” en Lighthouse



Fuente: Lighthouse

Figura 98. Prueba a “/projectMenu” en SEOquake Analytics



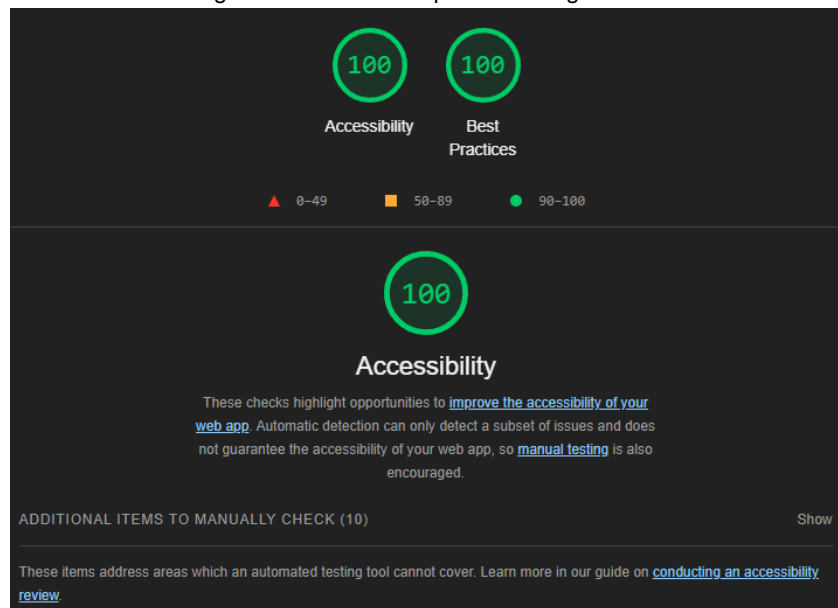
Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de

una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

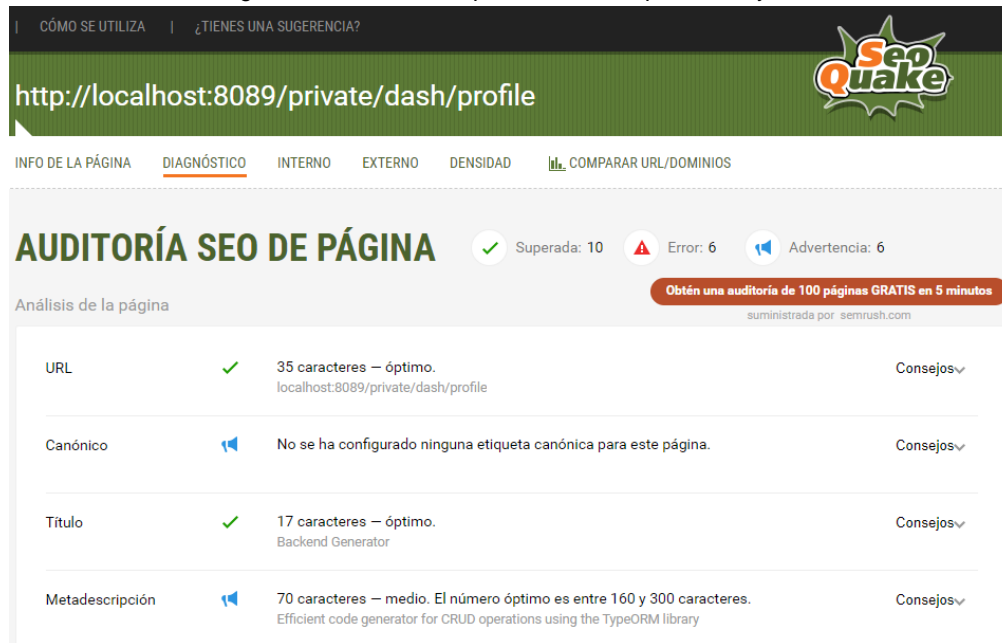
En la Figura 99 se muestra la prueba de Lighthouse, mientras que en la Figura 100 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/profile” de la meta herramienta.

Figura 99. Prueba a “/profile” en Lighthouse



Fuente: Lighthouse

Figura 100. Prueba a “/profile” en SEOquake Analytics

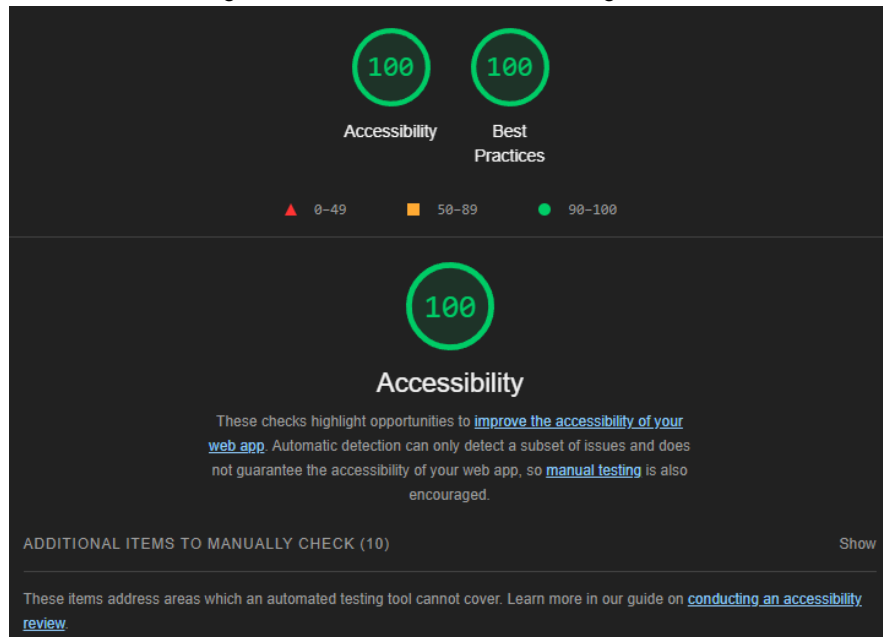


Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

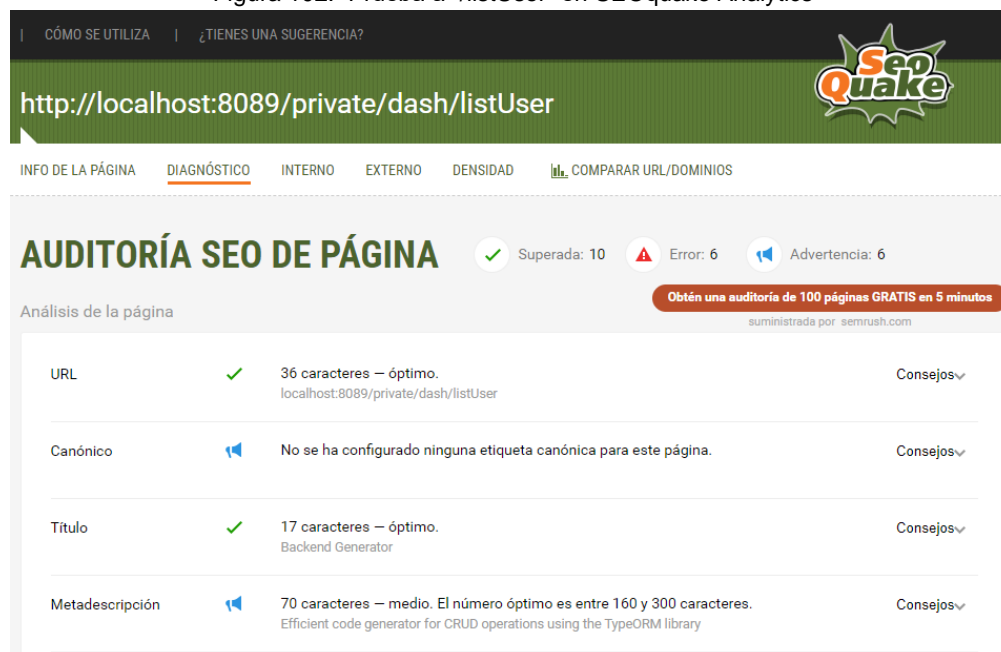
En la Figura 101 se muestra la prueba de Lighthouse, mientras que en la Figura 102 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/listUser” de la meta herramienta.

Figura 101. Prueba a "/listUser" en Lighthouse



Fuente: Lighthouse

Figura 102. Prueba a "/listUser" en SEOquake Analytics



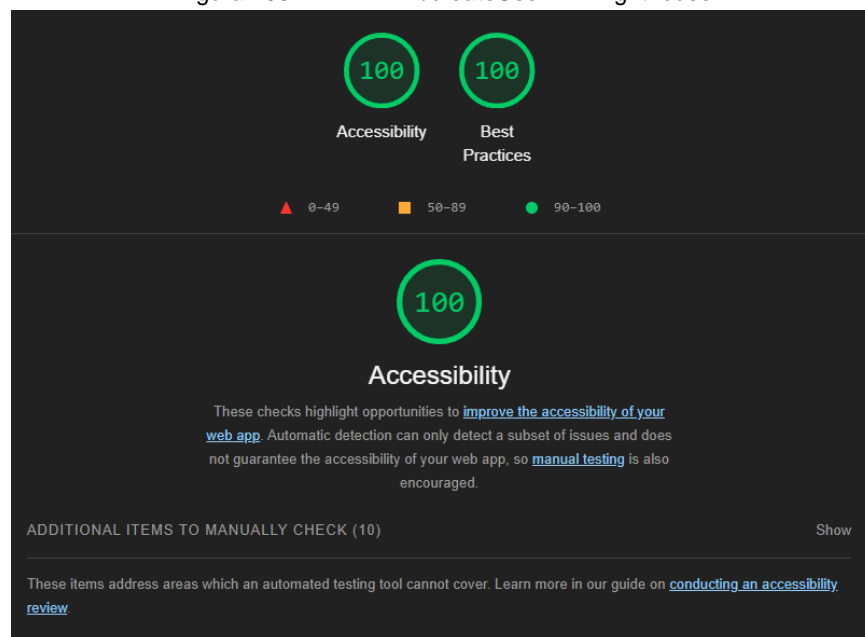
Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de

una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

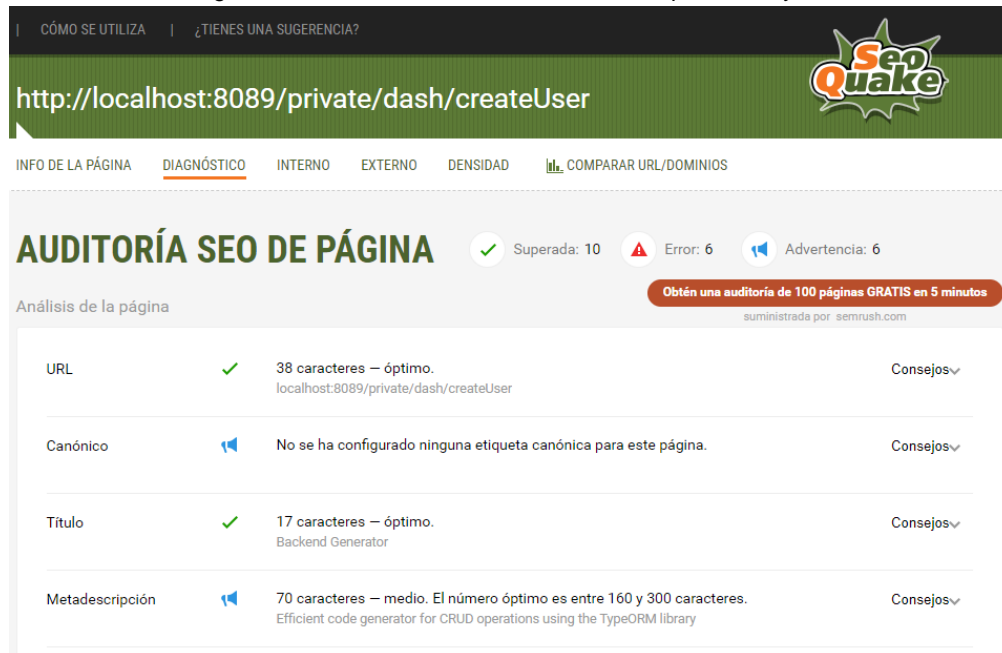
En la Figura 103 se muestra la prueba de Lighthouse, mientras que en la Figura 104 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/createUser” de la meta herramienta.

Figura 103. Prueba a “/createUser” en Lighthouse



Fuente: Lighthouse

Figura 104. Prueba a “/createUser” en SEOquake Analytics

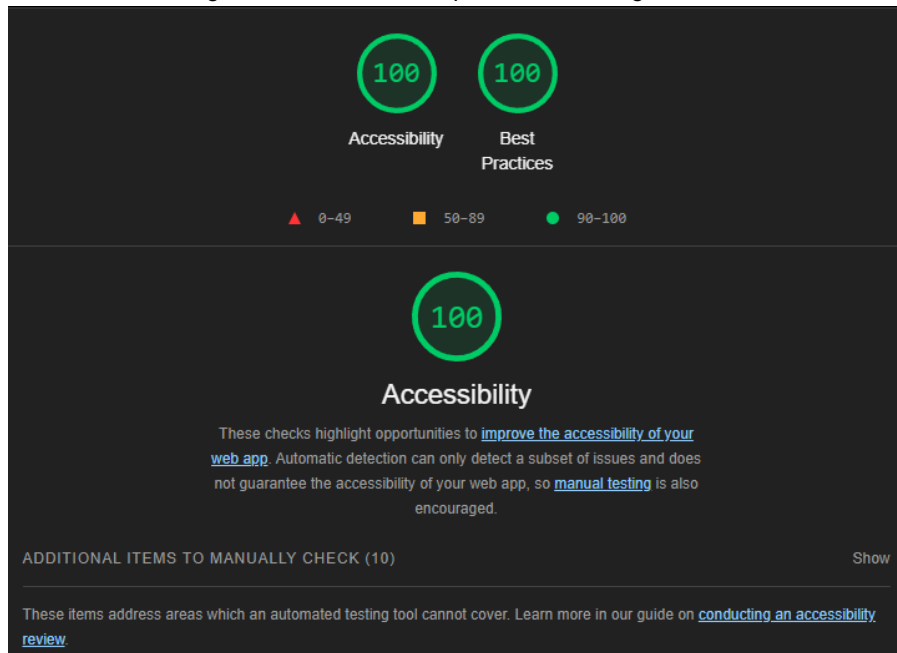


Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

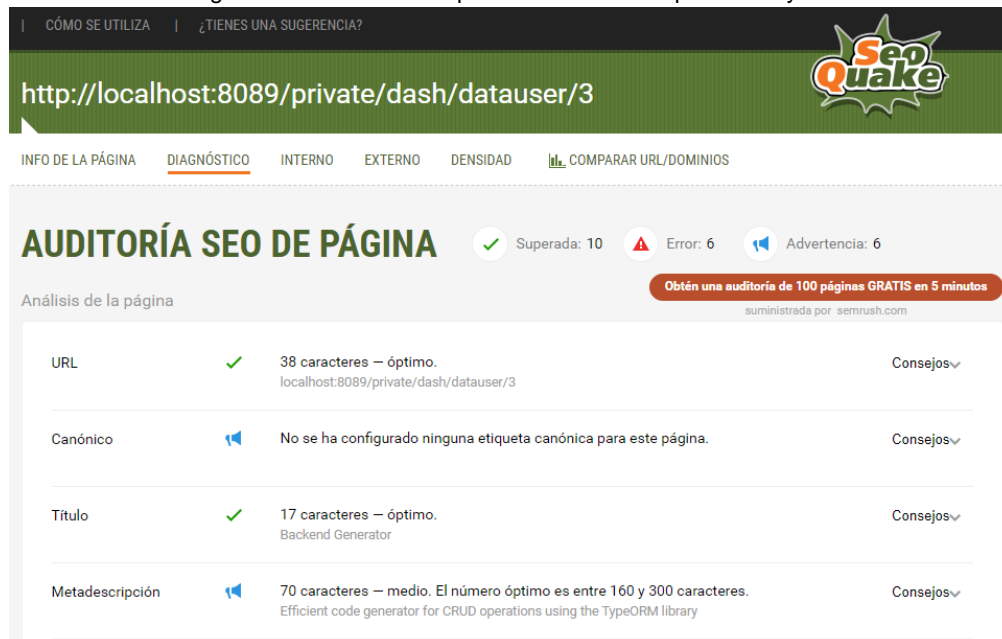
En la Figura 105 se muestra la prueba de Lighthouse, mientras que en la Figura 106 se presenta la prueba de SEO realizada con SEOquake Analytics desde el “/updateUser” de la meta herramienta.

Figura 105. Prueba a “/updateUser” en Lighthouse



Fuente: Lighthouse

Figura 106. Prueba a “/updateUser” en SEOquake Analytics



Fuente: SEOquake Analytics

Es importante destacar que los señalamientos sobre la falta de marcado Schema.org, el archivo robots.txt, sitemaps XML, Google™ Analytics y encabezados HTML son parte de una evaluación SEO. Si bien estos elementos son cruciales para mejorar nuestra visibilidad en los motores de búsqueda, no afectan la accesibilidad o usabilidad de nuestro sitio web en su estado actual.

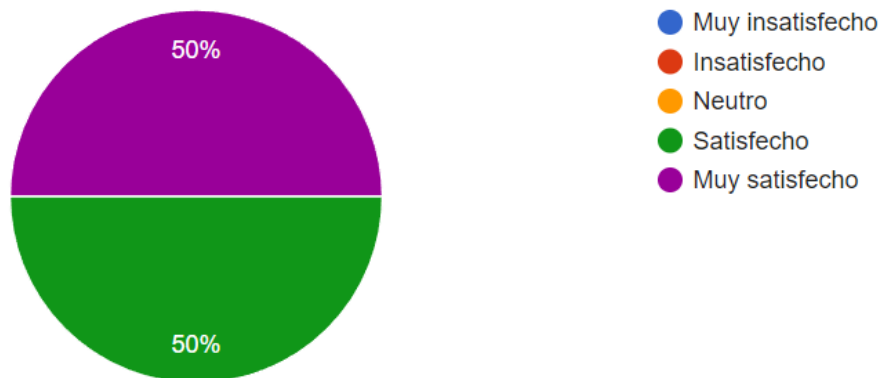
### 10.16. Anexo: Pruebas de experiencia de usuario

Este anexo presenta el análisis de las pruebas de experiencia de usuario de los resultados obtenidos de una encuesta realizada a 4 usuarios seleccionados. Este nos proporcionará una visión detallada de las percepciones, opiniones y comentarios de los participantes sobre la experiencia general de uso de la meta herramienta. Los datos recopilados incluyen aspectos como la facilidad de navegación, la claridad de la información, la utilidad de las funciones, así como sugerencias de mejora.

#### 1. ¿Cómo calificaría su experiencia general con nuestra meta herramienta?

La Figura 107 se observa que el (50%) de los usuarios calificaron su experiencia general con la meta herramienta como “satisfecho”, mientras que el (50%) restante se encuentran en la categoría de “Muy satisfechos”. No se registraron usuarios que calificaran la meta herramienta como “muy insatisfecho” o “insatisfecho”.

Figura 107. ¿Cómo calificaría su experiencia general con nuestra meta herramienta?

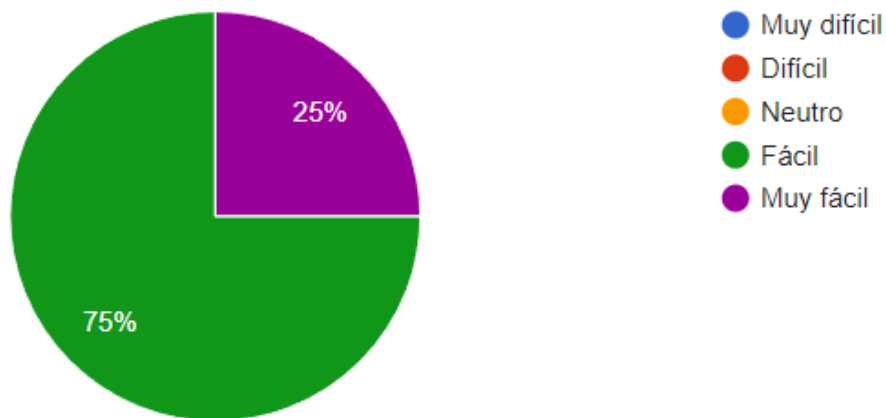


Fuente: Autores

#### 2. ¿Qué tan fácil le resultó navegar por la meta herramienta?

La Figura 108 se observa que el (75%) de los usuarios respondieron que les resulto “fácil” navegar por la meta herramienta, mientras que el (25%) restante se les resulto “muy fácil”. No se registraron usuarios que calificaran la meta herramienta como “neutro”, “difícil” o “muy difícil” de navegar.

Figura 108. ¿Qué tan fácil le resultó navegar por la meta herramienta?

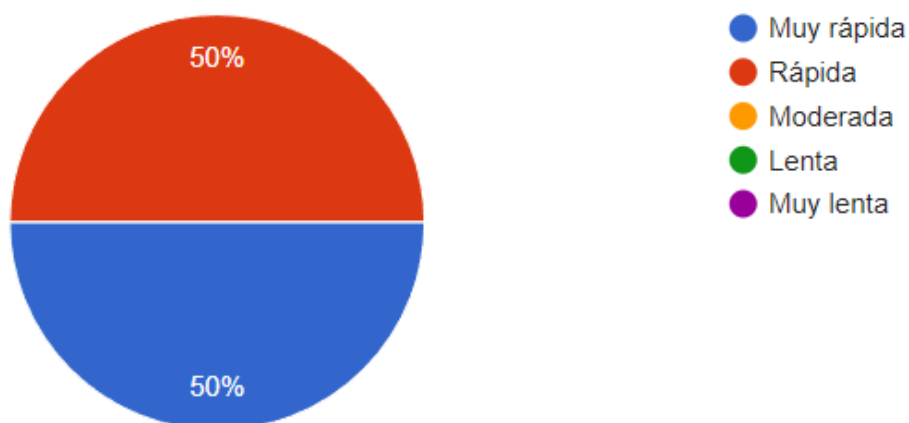


Fuente: Autores

3. ¿Cómo calificaría la velocidad de carga de la meta herramienta?

La Figura 109 se observa que el (50%) de los usuarios calificaron que les resulto “muy rápida” la carga de la meta herramienta, mientras que el (50%) restante se les resulto “rápida”. No se registraron usuarios que calificaran la meta herramienta como “moderada”, “lenta” o “muy lenta” de cargar.

Figura 109. ¿Cómo calificaría la velocidad de carga de la meta herramienta?

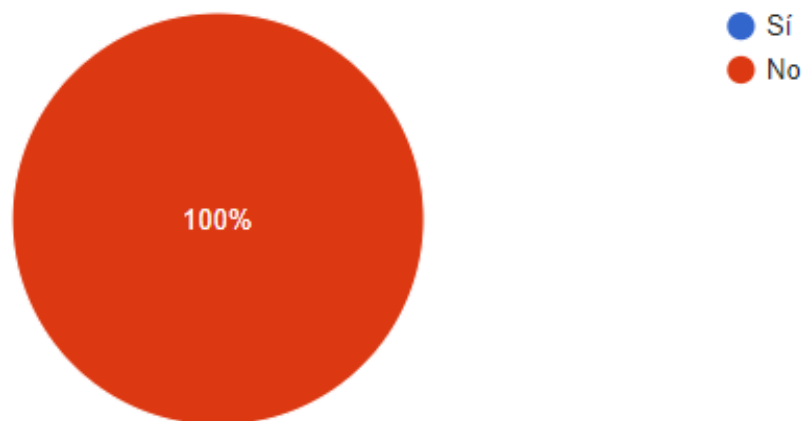


Fuente: Autores

4. ¿Ha experimentado errores o problemas técnicos durante su experiencia con el sitio web?

La Figura 110 se observa que el (100%) de los usuarios calificaron que no experimentaron errores o problemas técnicos durante su experiencia con el sitio web.

Figura 110. ¿Ha experimentado errores o problemas técnicos durante su experiencia con el sitio web?

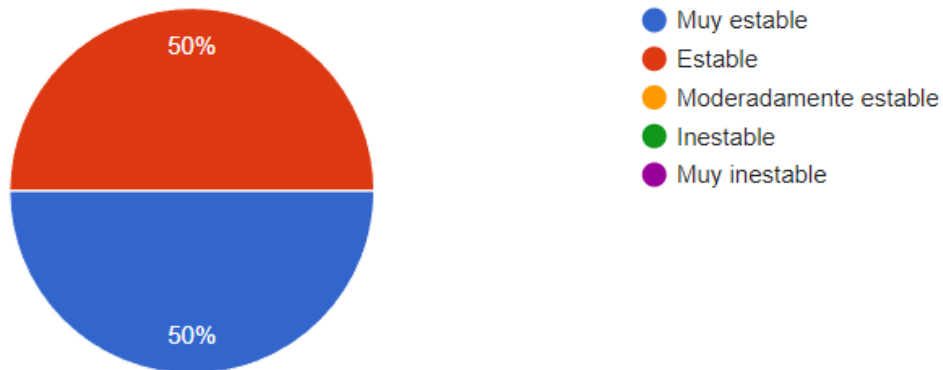


Fuente: Autores

5. ¿Cómo calificaría la estabilidad de la meta herramienta?

La Figura 111 se observa que el (50%) de los usuarios calificaron que les resultó “muy estable” la meta herramienta, mientras que el (50%) restante se les resultó “estable”. No se registraron usuarios que calificaran la meta herramienta como “moderadamente estable”, “inestable” o “muy inestable”.

Figura 111. ¿Cómo calificaría la estabilidad de la meta herramienta?

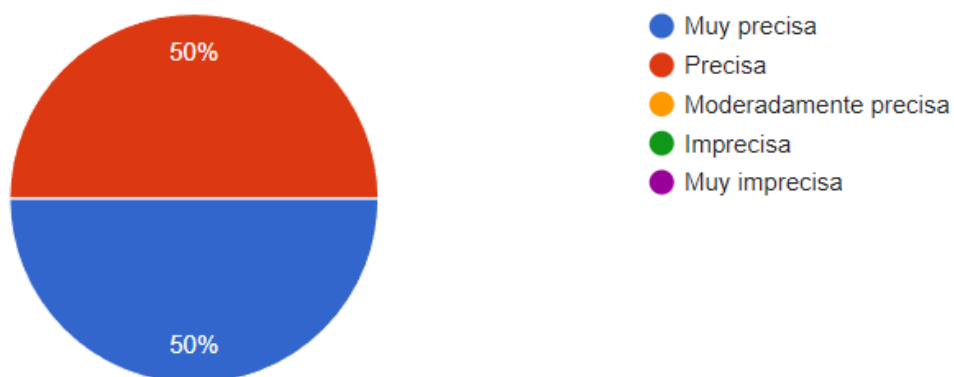


Fuente: Autores

6. ¿Cómo calificaría la precisión y consistencia de los datos presentados en la meta herramienta?

La Figura 112 se observa que el (50%) de los usuarios calificaron que les resultó “muy precisos” los datos presentados en la meta herramienta, mientras que el (50%) restante se les resultó unos datos “precisos”. No se registraron usuarios que calificaran la meta herramienta como “moderadamente precisos”, “imprecisos” o “muy imprecisos”.

Figura 112. ¿Cómo calificaría la precisión y consistencia de los datos presentados en la meta herramienta?



Fuente: Autores

7. ¿Cuál es el aspecto que más le gusta de nuestra meta herramienta?

A continuación, se presenta los aspectos positivos que los usuarios pudieron identificar de la meta herramienta:

- El formulario presenta una apariencia visualmente agradable y coherente. Los colores utilizados son suaves y no distraen del contenido, el manual de usuario es claro y completo, los pasos están ordenados de forma lógica y se incluyen ejemplos útiles para ayudar al usuario a comprender mejor cada sección
- El diseño, la página tiene muy claro los estándares de usabilidad, el responsive, y muy buena la distribución de la información de los iconos
- Lo que más me gusta de su página web es su alta interactividad y la opción de copiar o descargar código. Es genial tener acceso a estas funciones, especialmente teniendo en cuenta su manual de usuario ubicado en el menú principal. Esto hace que la experiencia del usuario sea muy completa y satisfactoria, ya que pueden interactuar fácilmente con el contenido y utilizarlo de manera práctica.
- El control de usuarios es muy llamativo, donde se le permite al usuario editar su información de perfil que mejora su experiencia en la aplicación y considero importante contar con el rol de perfiles que hay, ya que como administrador tengo el control de mis usuarios, permitiendo la creación, edición de usuarios que deseen acceder a la aplicación.

8. ¿Recomendaría nuestra meta herramienta a un amigo o colega?

A continuación, se presenta las aprobaciones de las recomendaciones de los usuarios a amigos o colegas de nuestra meta herramienta:

- Si, ya que es una página muy bien ordenada y adecuada además de que cumple con su objetivo de una forma muy satisfactoria
- Si la recomendaría a amigos o colegas del sector de programación que busquen automatizar el proceso de creación de una BD y ahorrar tiempos.

- Definitivamente recomendaría su página web a un amigo. Su capacidad para simplificar esta tarea compleja puede ahorrar mucho tiempo y esfuerzo a quienes trabajan en el desarrollo de aplicaciones.
- Si, el contar con una herramienta que automatice la generación de código backend para operaciones CRUD con solo conectarme a mi base de datos relacional es muy impresionante.

9. ¿Hay algún otro comentario o sugerencia que desee compartir?

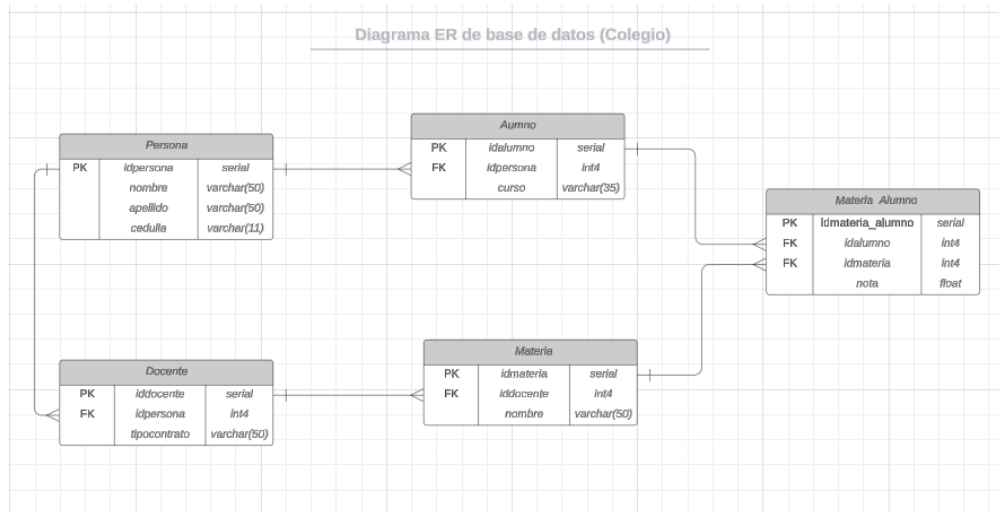
A continuación, se presenta los comentarios o sugerencias que los usuarios compartieron con respecto a nuestra meta herramienta:

- Me gustó que facilite la opción de guardar o descargar el código lo que considero muy importante y ayuda a agilizar los procesos de desarrollo
- Una gran idea, con grandes aplicaciones, desde el entorno de programación hasta en el uso de enseñanza y aprendizaje de generar y consumir servicios de una tecnología en cuestión.
- Ninguno.
- El diseño UX/UI es muy bueno para el usuario.

### 10.17. Anexo: Plan de pruebas Base de Datos (“Colegio”)

En la Figura 113 se encuentra el diagrama Entidad-Relación (ER) de la base de datos "Colegio", la cual será utilizada como el enfoque principal para las pruebas del generador de backend. Este diagrama representa la estructura conceptual de la base de datos, mostrando las entidades principales, sus atributos y las relaciones entre ellas.

Figura 113. Diagrama Entidad-Relación (ER) de la base de datos “Colegio”



Fuente: Autores

Se diligencia los campos del formulario de conexión con las credenciales de conexión a la base de datos con la cual queremos generar el backend, como se muestra en la Figura 114.

Figura 114. Formulario de conexión a base de datos

Database connection

postgres

localhost

Colegio

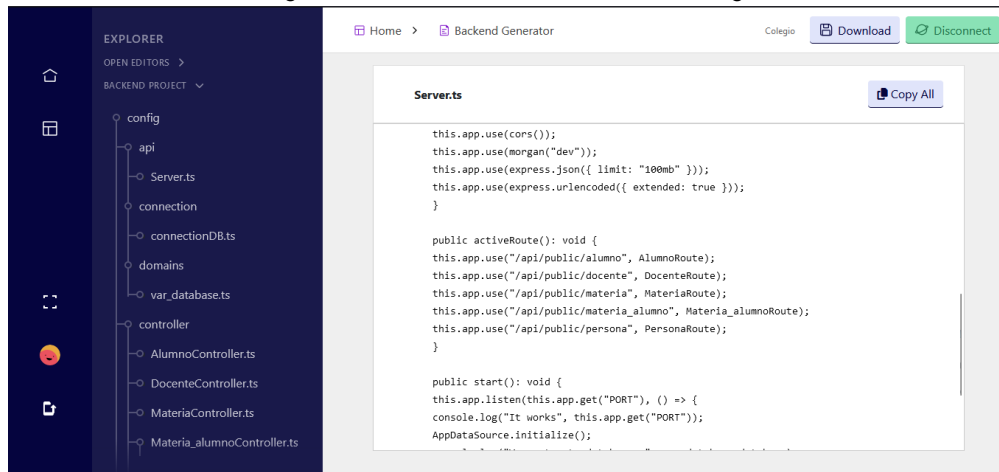
..... 5432

Connect

Fuente: Autores

Lograr establecer con éxito la conexión a nuestra base de datos principal. Esto es fundamental para avanzar en nuestro proyecto de desarrollo de software. En la Figura 115, se puede observar el proceso de conexión a la base de datos y la generación automática de código basada en su estructura.

Figura 115. Generación automática de código



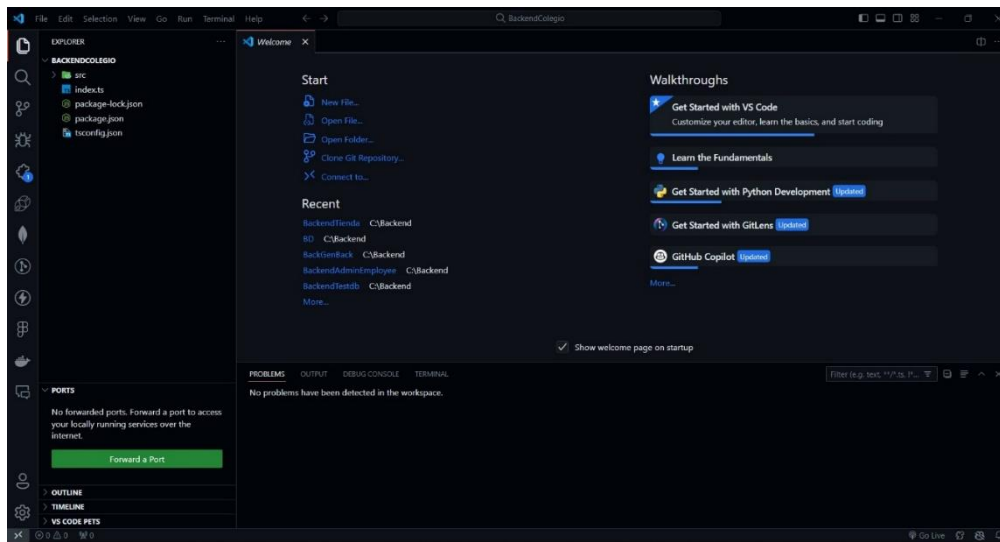
Fuente: Autores

La descarga del proyecto que contiene el código backend necesario para nuestro desarrollo. Este paso es fundamental para comenzar a trabajar en la implementación de las funcionalidades principales.

En la Figura 116, se muestra la interfaz de Visual Studio Code (VS Code), nuestra herramienta principal para el desarrollo de software. Esta interfaz proporciona un entorno integrado y altamente personalizable que facilita la navegación y modificación del código fuente.

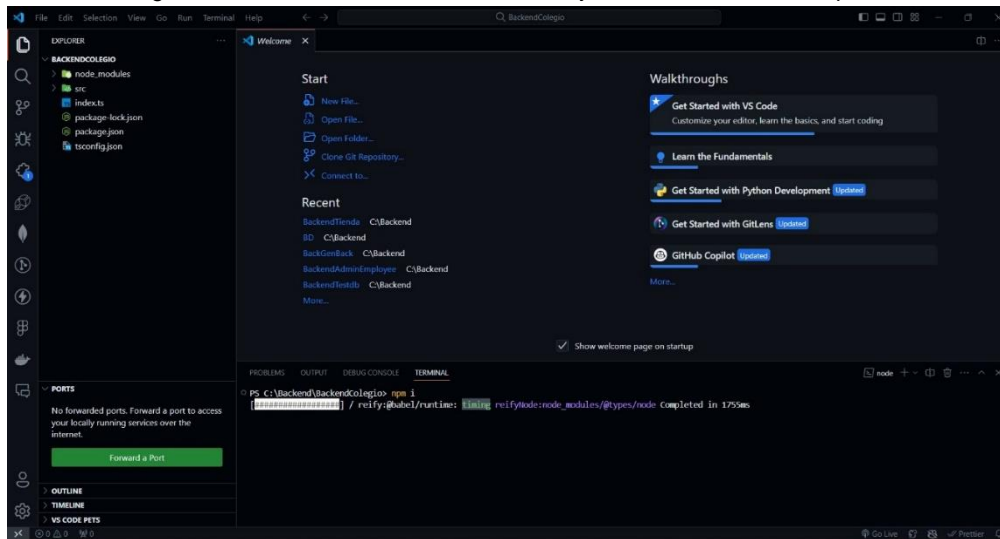
A la vez, en la Figura 117, se muestra la ejecución del comando npm i en la terminal integrada de VS Code. Este comando, abreviatura de npm install, se utiliza para instalar todas las dependencias del proyecto especificadas en el archivo package.json.

Figura 116. Interfaz de Visual Studio Code con código generado



Fuente: Autores

Figura 117. Interfaz de Visual Code con ejecución de comando “npm i”



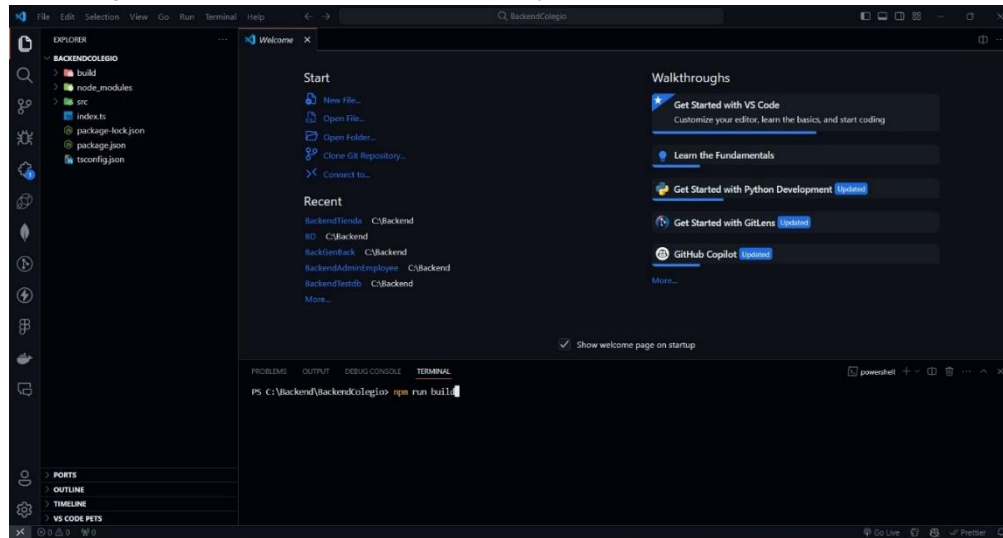
Fuente: Autores

- **Puesta en marcha:**

En la Figura 118, se puede observar la ejecución del comando `npm run build` en la terminal. Este comando se utiliza para compilar y construir los archivos necesarios para la aplicación, como transpilar código TypeScript a JavaScript o generar archivos estáticos.

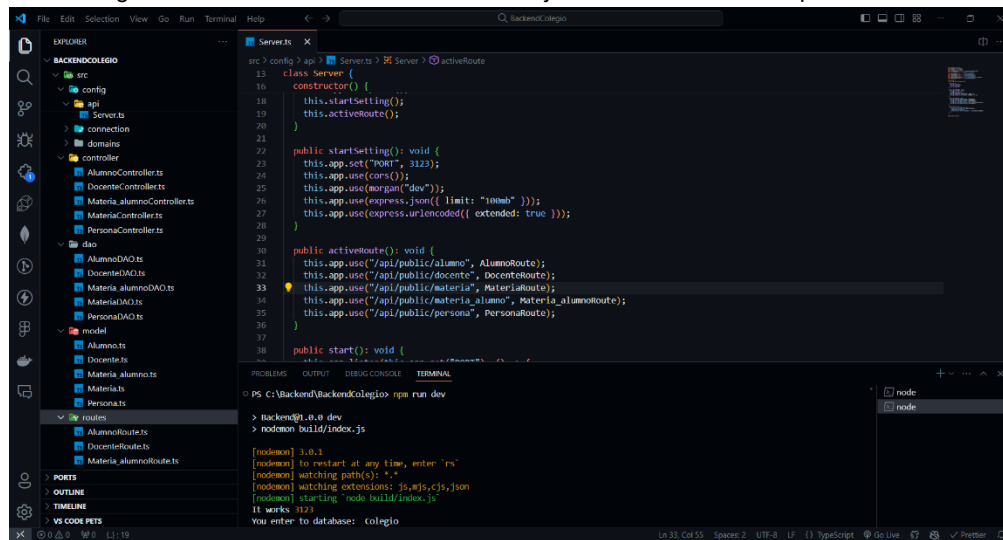
A la vez, en la Figura 119, se muestra la ejecución del comando `npm run dev` en una nueva terminal. Este comando inicia el servidor de desarrollo del backend, permitiendo que nuestra aplicación se ejecute localmente con capacidades de recarga automática.

Figura 118. Interfaz de Visual Code con ejecución de comando “npm run build”



Fuente: Autores

Figura 119. Interfaz de Visual Code con ejecución de comando “npm run dev”



Fuente: Autores

- **Pruebas a los EndPoint generados:**

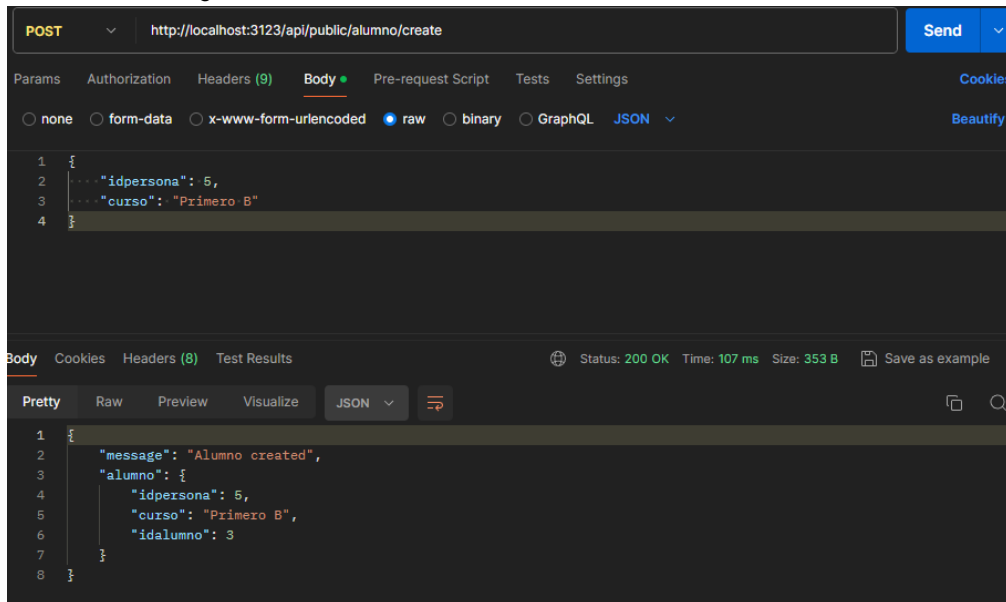
Estas pruebas son cruciales para asegurar el correcto funcionamiento de nuestras API y garantizar la integridad de las operaciones CRUD en relación con las tablas de nuestra base de datos de conexión.

Cada una de las tablas en la base de datos ha sido sometida a pruebas específicas para validar las siguientes operaciones:

- **Create (Crear):** Utilizando solicitudes POST, se crearon nuevos registros en cada tabla de la base de datos. Estas pruebas verifican que los datos sean correctamente persistidos y que se generen identificadores únicos para cada nuevo registro.
- **Read (Leer):** Empleando solicitudes GET, se recuperaron registros existentes de cada tabla. Estas pruebas validan que los datos se devuelvan correctamente.
- **Update (Actualizar):** A través de solicitudes PUT, se modificaron registros existentes en las tablas. Estas pruebas garantizan que los cambios en los datos se reflejen de manera adecuada y que las actualizaciones sean precisas y consistentes.
- **Delete (Eliminar):** Mediante solicitudes DELETE, se han eliminado registros seleccionados en cada tabla. Estas pruebas verifican que los datos sean eliminados correctamente.

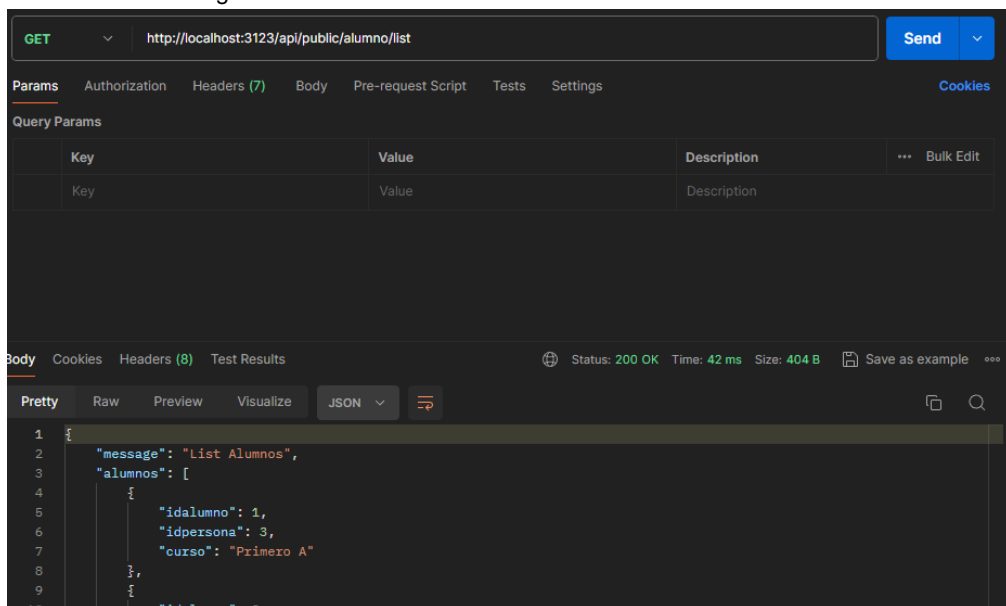
A continuación, en la Figura 120, Figura 121, Figura 122 y Figura 123 se podrá visualizar las pruebas a los endpoints generados de la tabla “alumno”.

Figura 120. Prueba de EndPoint "Create" de la tabla "alumno"



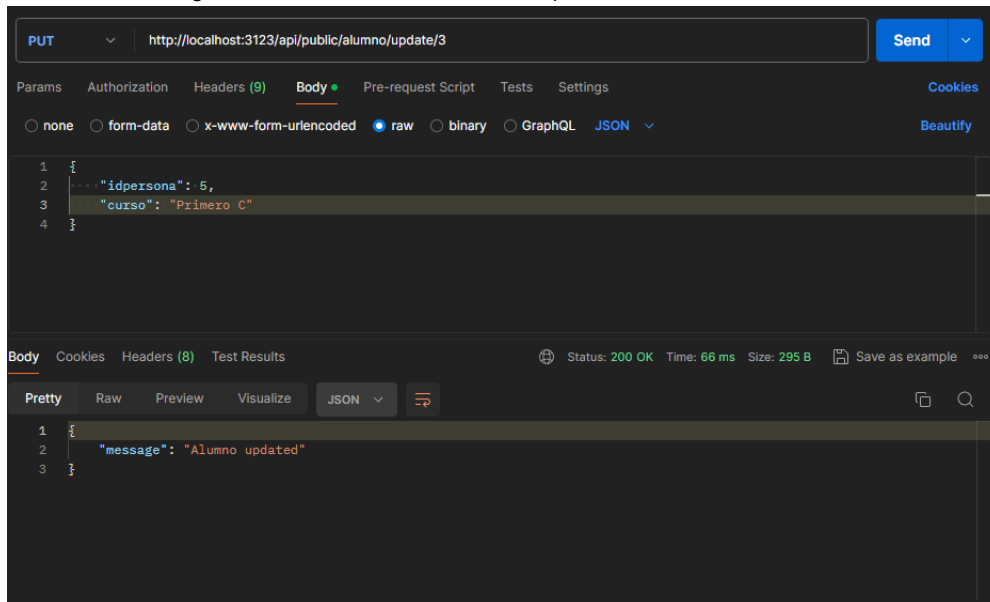
Fuente: Autores

Figura 121. Prueba de EndPoint "List" de la tabla "alumno"



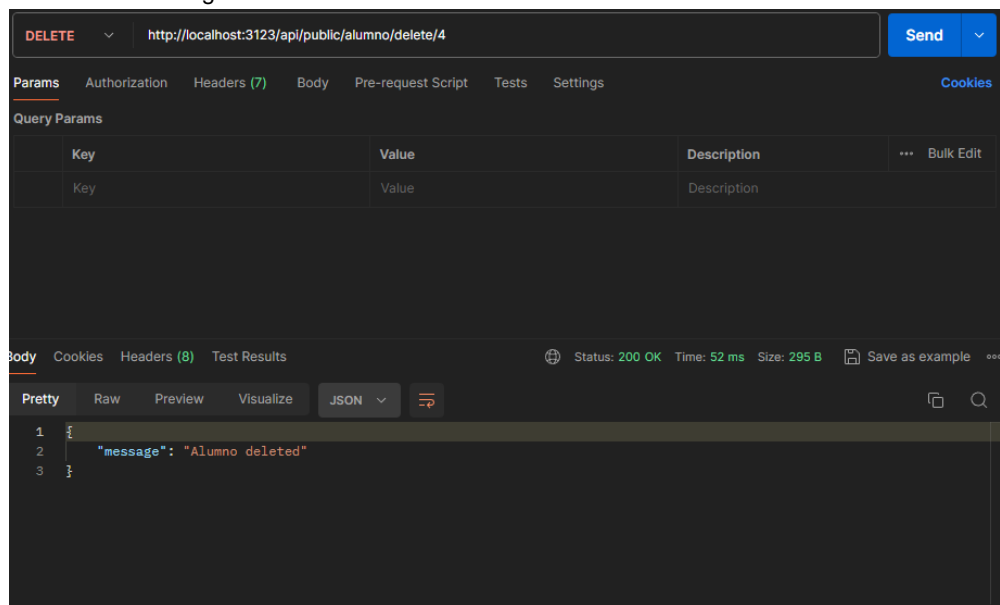
Fuente: Autores

Figura 122. Prueba de EndPoint “Update” de la tabla “alumno”



Fuente: Autores

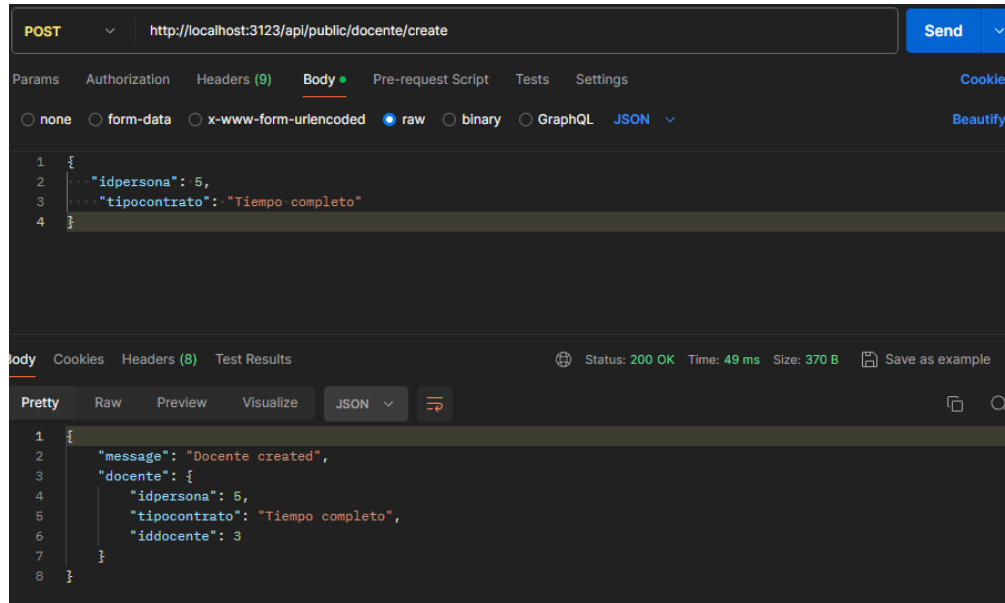
Figura 123. Prueba de EndPoint “Delete” de la tabla “alumno”



Fuente: Autores

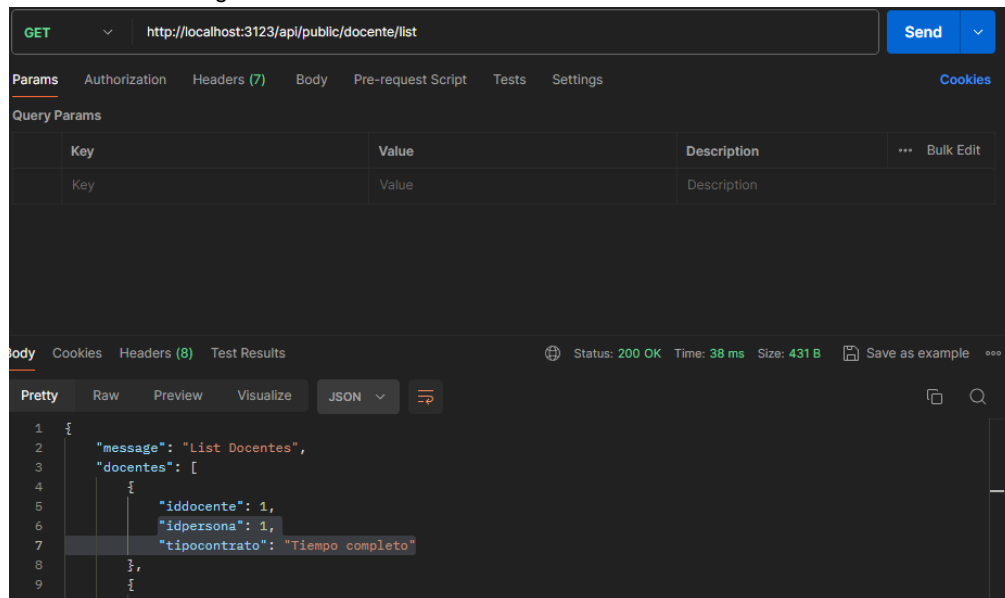
A continuación, en la Figura 124, Figura 125, Figura 126 y Figura 127 se podrá visualizar las pruebas a los endpoints generados de la tabla “docente”.

Figura 124. Prueba de EndPoint “Create” de la tabla “docente”



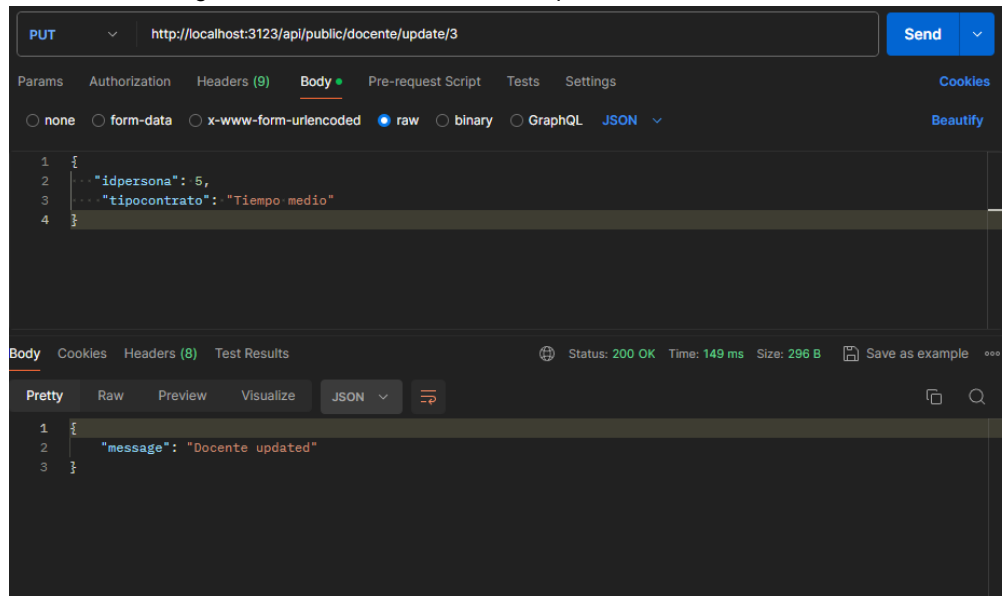
Fuente: Autores

Figura 125. Prueba de EndPoint “List” de la tabla “docente”



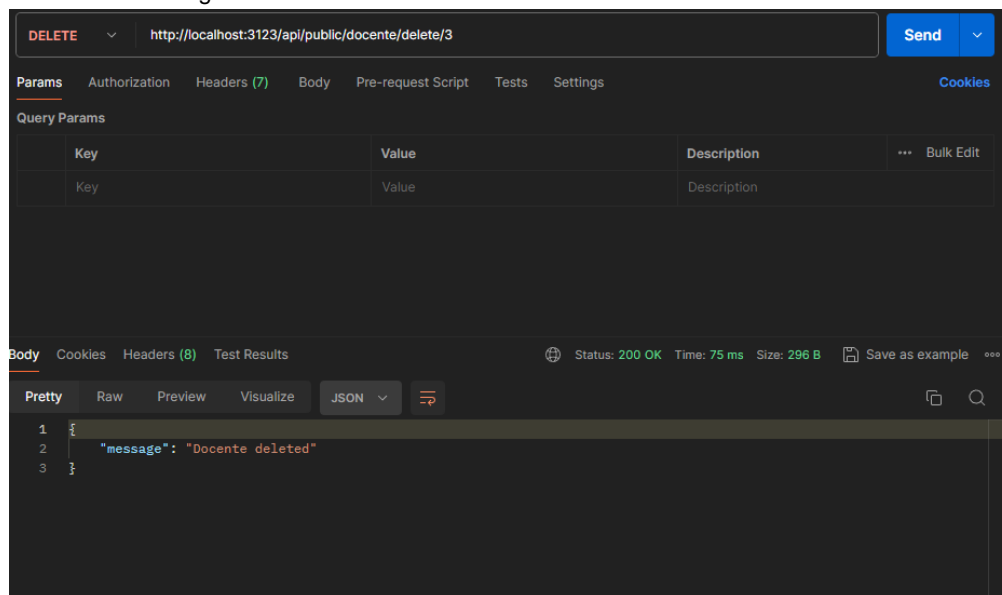
Fuente: Autores

Figura 126. Prueba de EndPoint “Update” de la tabla “docente”



Fuente: Autores

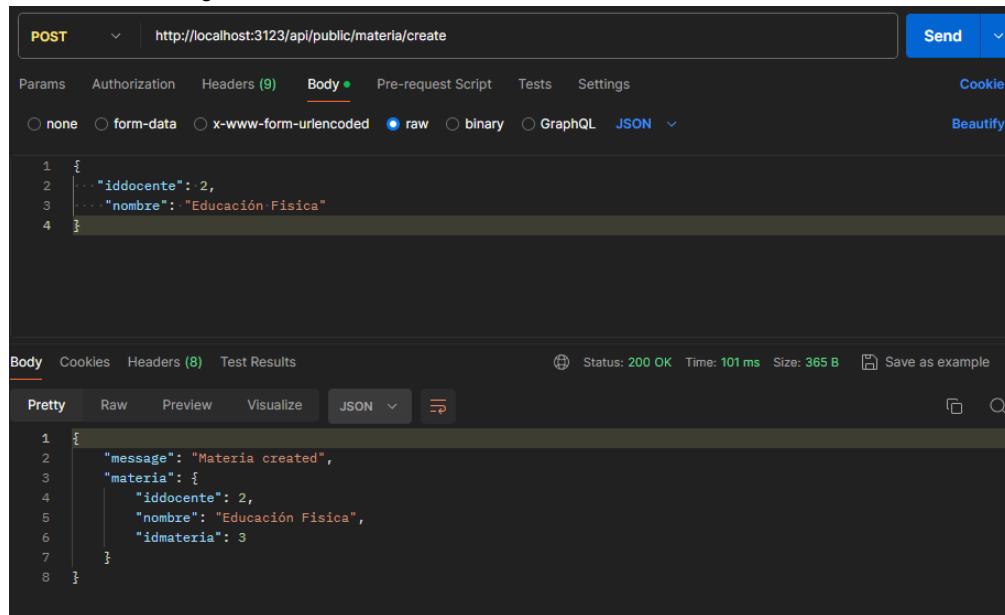
Figura 127. Prueba de EndPoint “Delete” de la tabla “docente”



Fuente: Autores

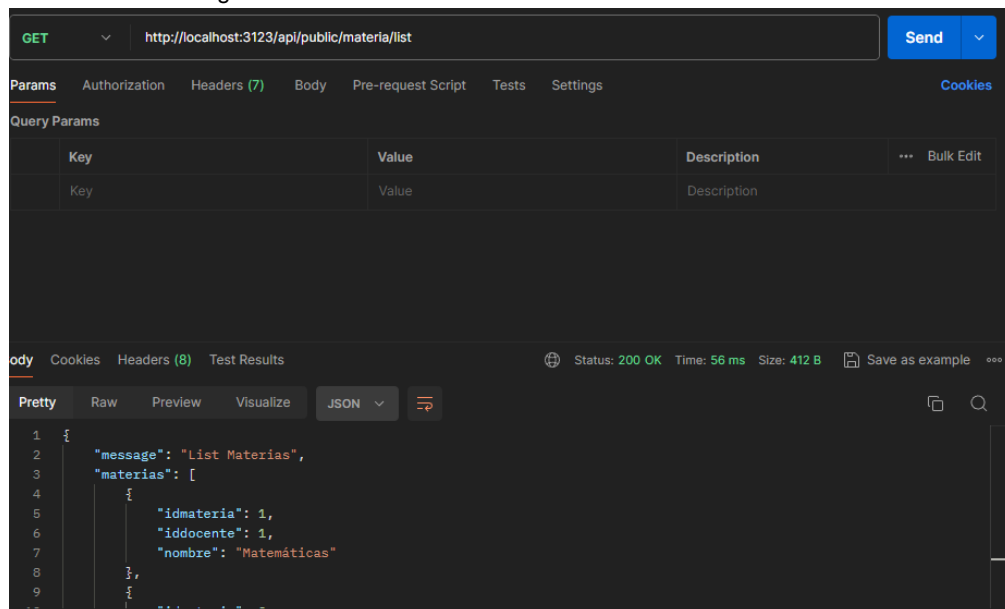
A continuación, en la Figura 128, Figura 129, Figura 130 y Figura 131 se podrá visualizar las pruebas a los endpoints generados de la tabla “materia”.

Figura 128. Prueba de EndPoint “Create” de la tabla “materia”



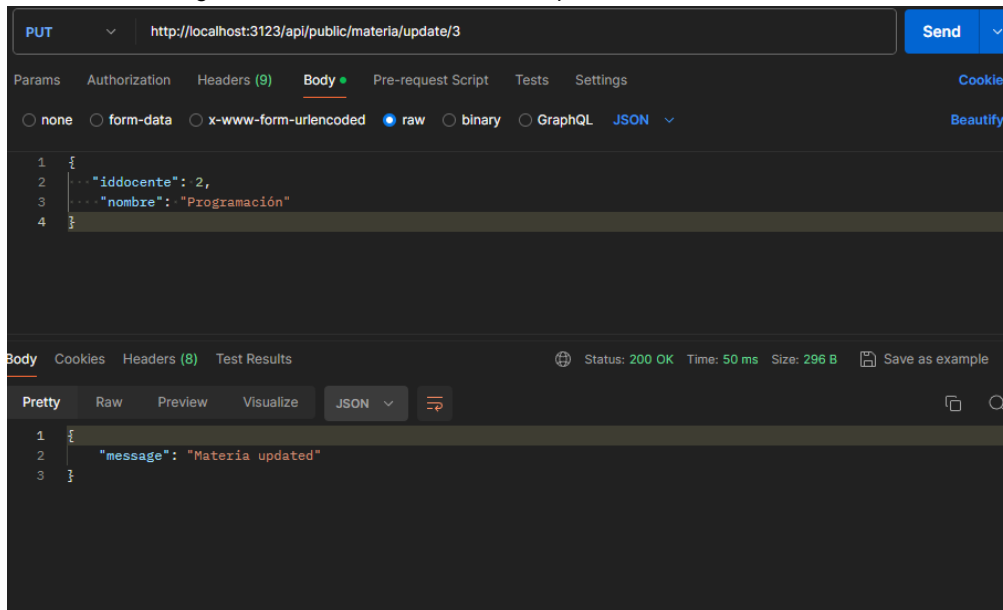
Fuente: Autores

Figura 129. Prueba de EndPoint “List” de la tabla “materia”



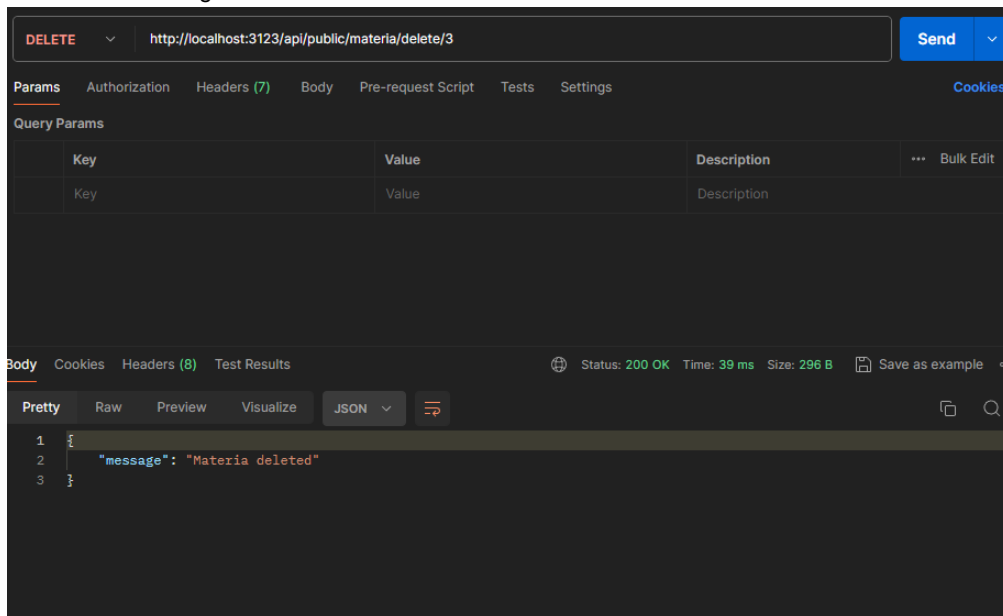
Fuente: Autores

Figura 130. Prueba de EndPoint “Update” de la tabla “materia”



Fuente: Autores

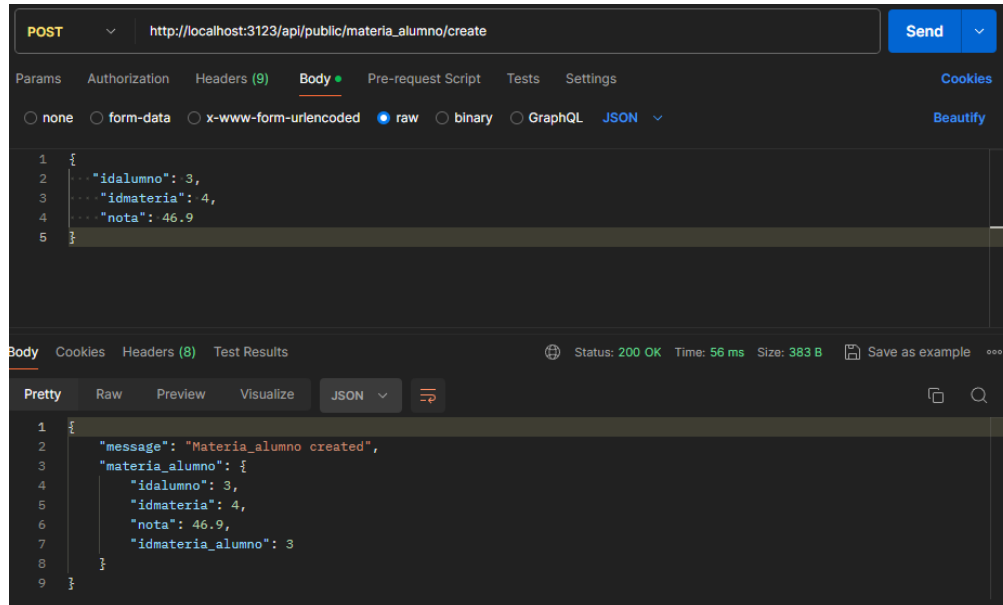
Figura 131. Prueba de EndPoint “Delete” de la tabla “materia”



Fuente: Autores

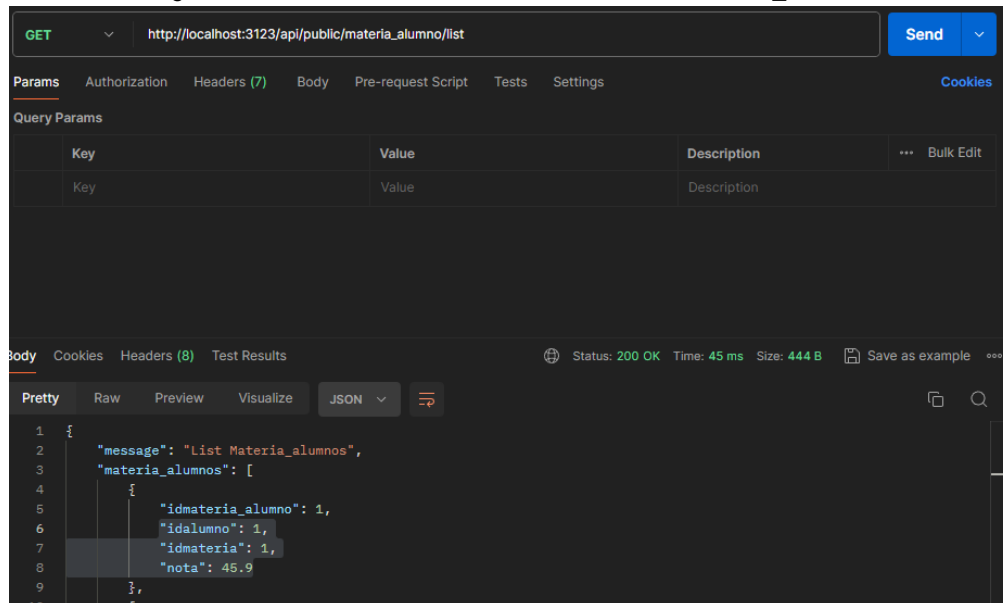
La A continuación, en la Figura 132, Figura 133, Figura 134 y Figura 135 se podrá visualizar las pruebas a los endpoints generados de la tabla “materia\_alumno”.

Figura 132. Prueba de EndPoint “Create” de la tabla “materia\_alumno”



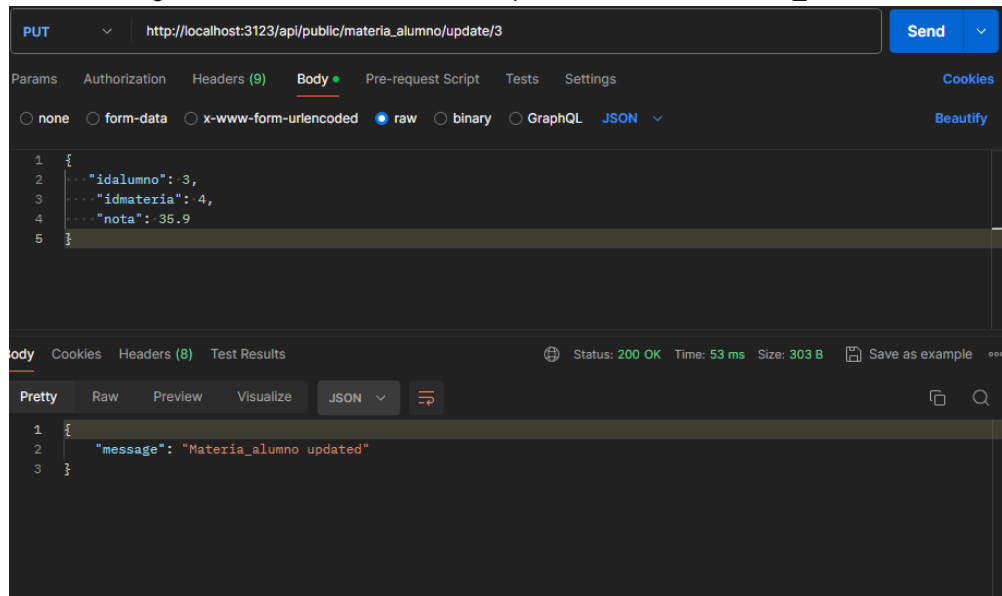
Fuente: Autores

Figura 133. Prueba de EndPoint “List” de la tabla “materia\_alumno”



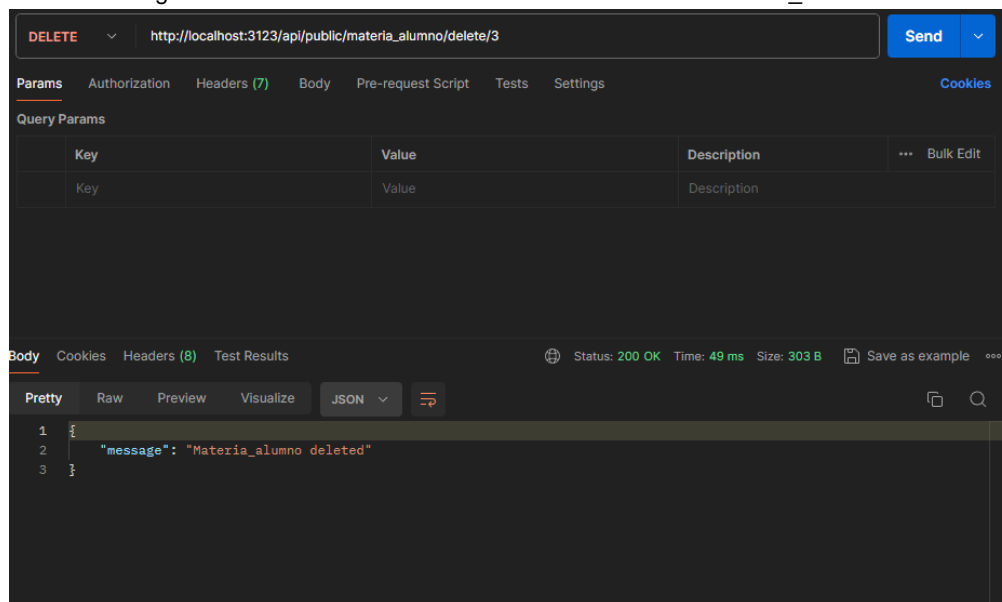
Fuente: Autores

Figura 134. Prueba de EndPoint “Update” de la tabla “materia\_alumno”



Fuente: Autores

Figura 135. Prueba de EndPoint “Delete” de la tabla “materia\_alumno”



Fuente: Autores

A continuación, en la Figura 136, Figura 137, Figura 138 y Figura 139 se podrá visualizar las pruebas a los endpoints generados de la tabla “persona”.

Figura 136. Prueba de EndPoint “Create” de la tabla “persona”

```
POST http://localhost:3123/api/public/persona/create

{
  "nombre": "Santiago",
  "apellido": "Vela",
  "cedula": "54321678901"
}

Status: 200 OK Time: 75 ms Size: 384 B

{
  "message": "Persona created",
  "persona": {
    "nombre": "Santiago",
    "apellido": "Vela",
    "cedula": "54321678901",
    "idpersona": 5
  }
}
```

Fuente: Autores

Figura 137. Prueba de EndPoint “List” de la tabla “persona”

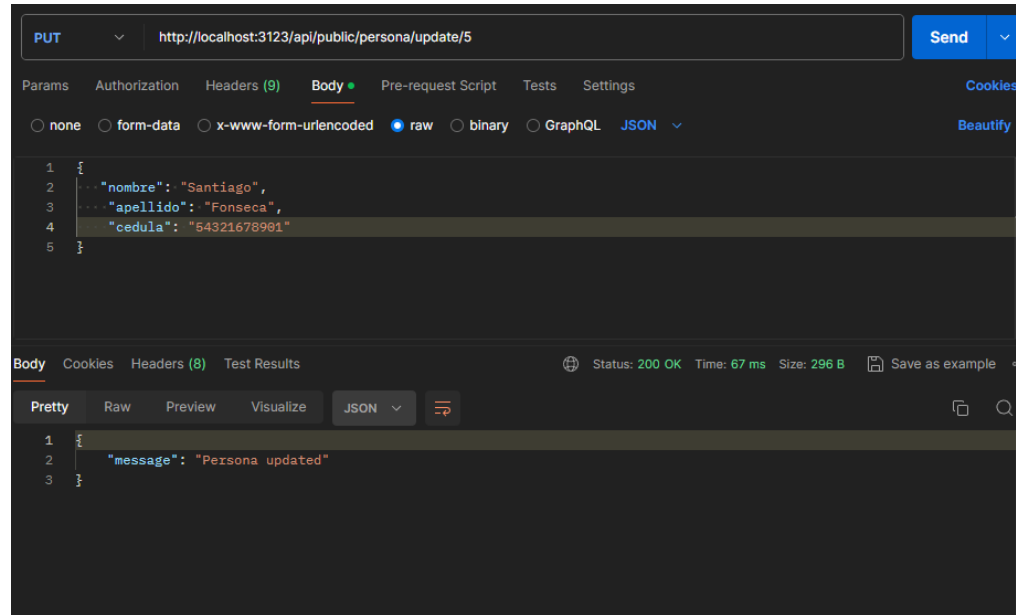
```
GET http://localhost:3123/api/public/persona/list

Status: 200 OK Time: 42 ms Size: 693 B

{
  "message": "List Personas",
  "personas": [
    {
      "idpersona": 1,
      "nombre": "Juan",
      "apellido": "Pérez",
      "cedula": "12345678901"
    }
  ]
}
```

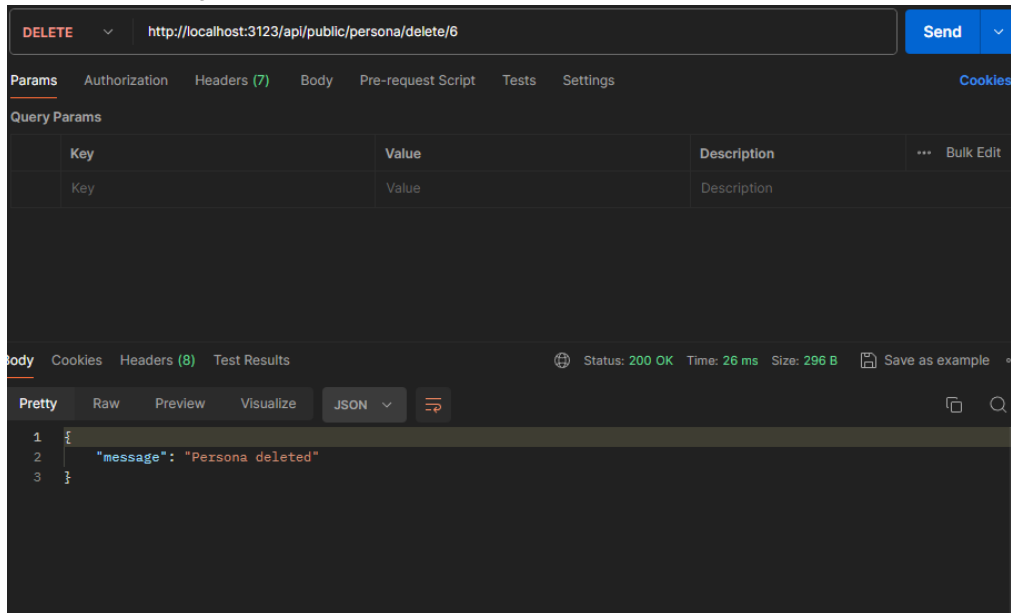
Fuente: Autores

Figura 138. Prueba de EndPoint “Update” de la tabla “persona”



Fuente: Autores

Figura 139. Prueba de EndPoint “Delete” de la tabla “persona”

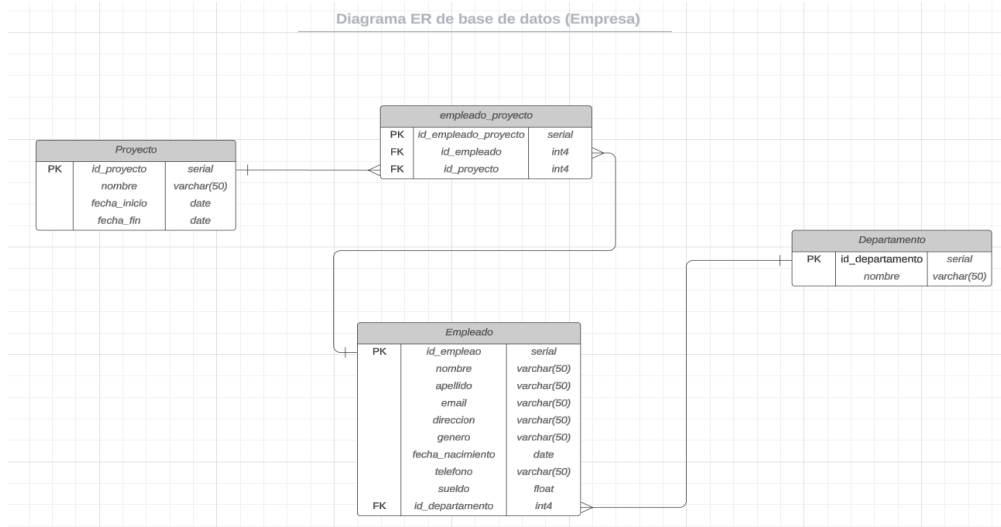


Fuente: Autores

### 10.18. Anexo: Plan de pruebas Base de Datos (“Empresa”)

En la Figura 140 se encuentra el diagrama Entidad-Relación (ER) de la base de datos "Empresa", la cual será utilizada como el enfoque principal para las pruebas del generador de backend. Este diagrama representa la estructura conceptual de la base de datos, mostrando las entidades principales, sus atributos y las relaciones entre ellas.

Figura 140. Diagrama Entidad-Relación (ER) de la base de datos “Empresa”



Fuente: Autores

Se diligencia los campos del formulario de conexión con las credenciales de conexión a la base de datos con la cual queremos generar el backend, como se muestra en la Figura 141 y Figura 114.

Figura 141. Formulario de conexión a base de datos

Database connection

postgres

localhost

Empresa

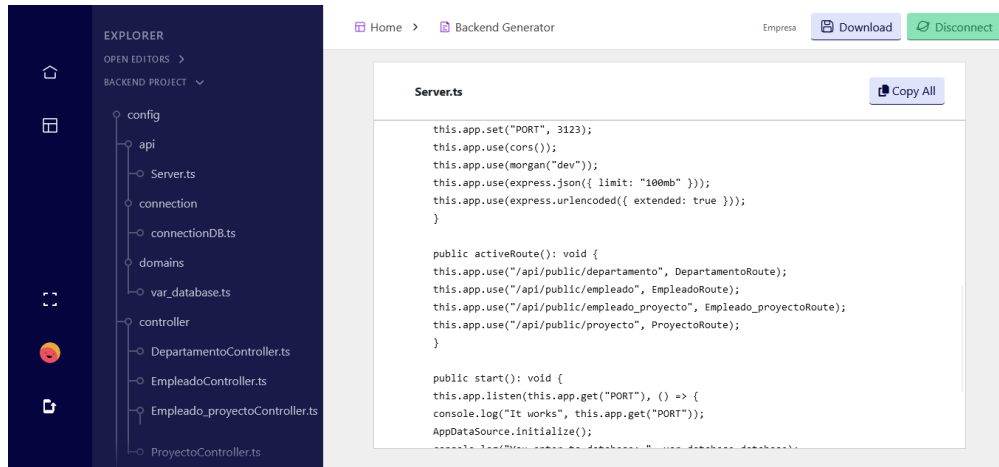
..... 5432

Connect

Fuente: Autores

Lograr establecer con éxito la conexión a nuestra base de datos principal. Esto es fundamental para avanzar en nuestro proyecto de desarrollo de software. En la Figura 142, se puede observar el proceso de conexión a la base de datos y la generación automática de código basada en su estructura.

Figura 142. Generación automática de código



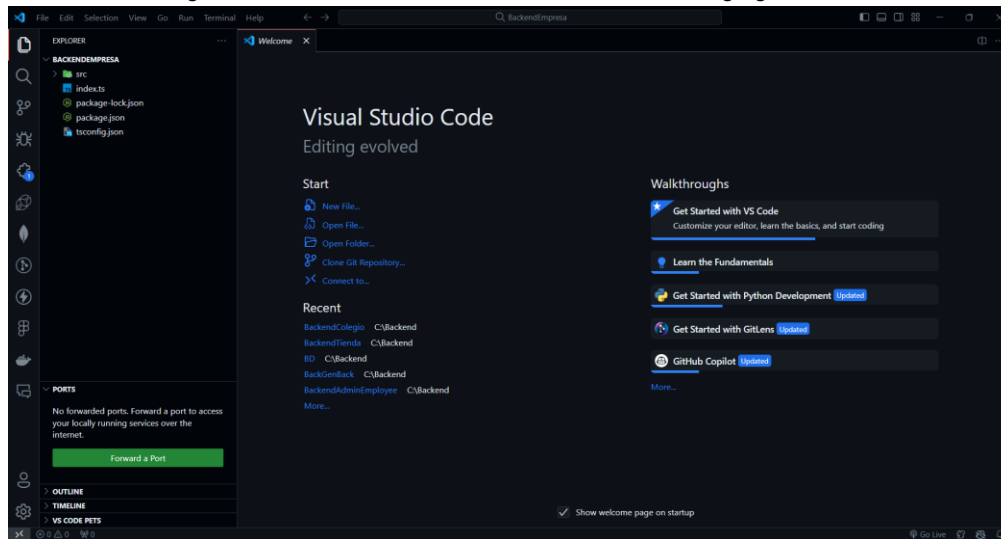
Fuente: Autores

La descarga del proyecto que contiene el código backend necesario para nuestro desarrollo. Este paso es fundamental para comenzar a trabajar en la implementación de las funcionalidades principales.

En la Figura 143, se muestra la interfaz de Visual Studio Code (VS Code), nuestra herramienta principal para el desarrollo de software. Esta interfaz proporciona un entorno integrado y altamente personalizable que facilita la navegación y modificación del código fuente.

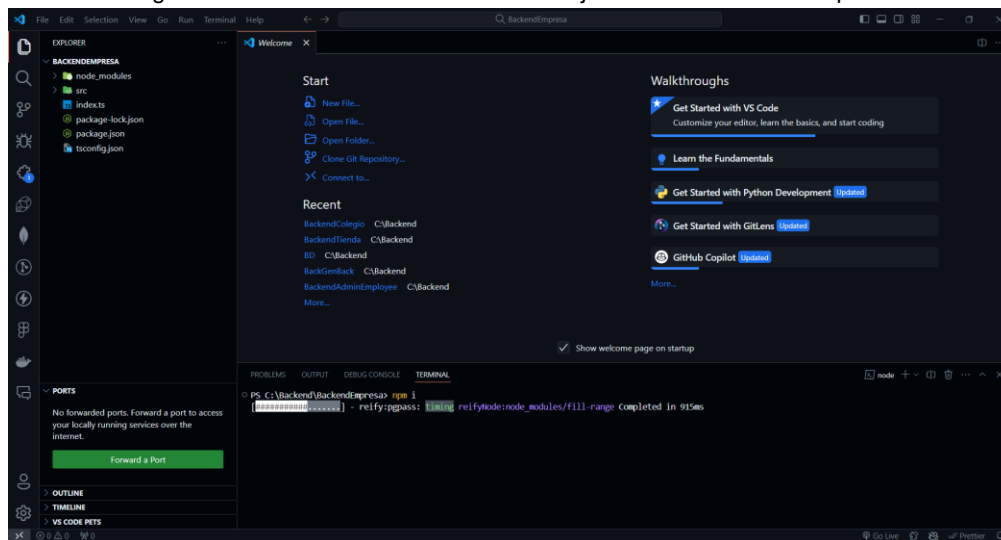
A la vez, en la Figura 144, se muestra la ejecución del comando `npm i` en la terminal integrada de VS Code. Este comando, abreviatura de `npm install`, se utiliza para instalar todas las dependencias del proyecto especificadas en el archivo `package.json`.

Figura 143. Interfaz de Visual Studio Code con código generado



Fuente: Autores

Figura 144. Interfaz de Visual Code con ejecución de comando “npm i”



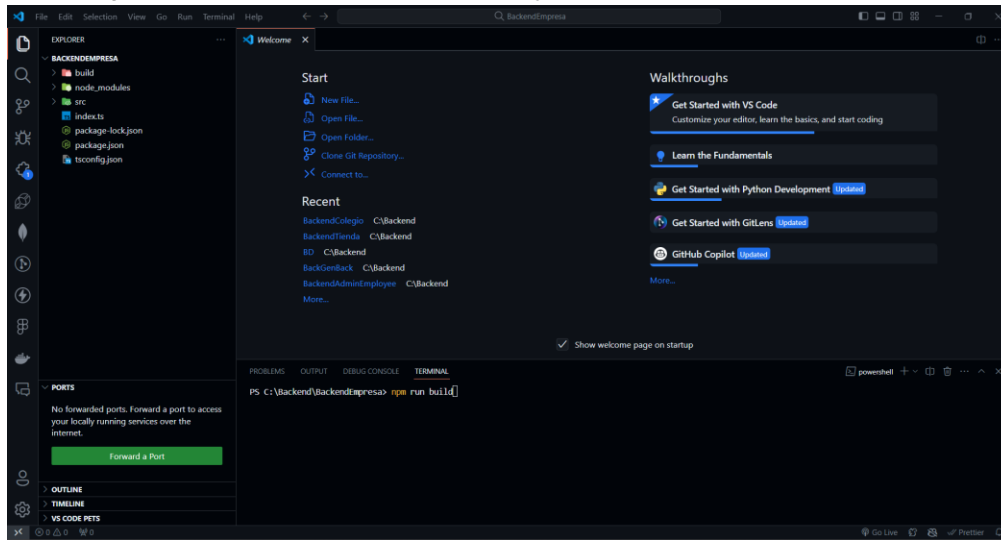
Fuente: Autores

- **Puesta en marcha:**

En la Figura 145, se puede observar la ejecución del comando `npm run build` en la terminal. Este comando se utiliza para compilar y construir los archivos necesarios para la aplicación, como transpilar código TypeScript a JavaScript o generar archivos estáticos.

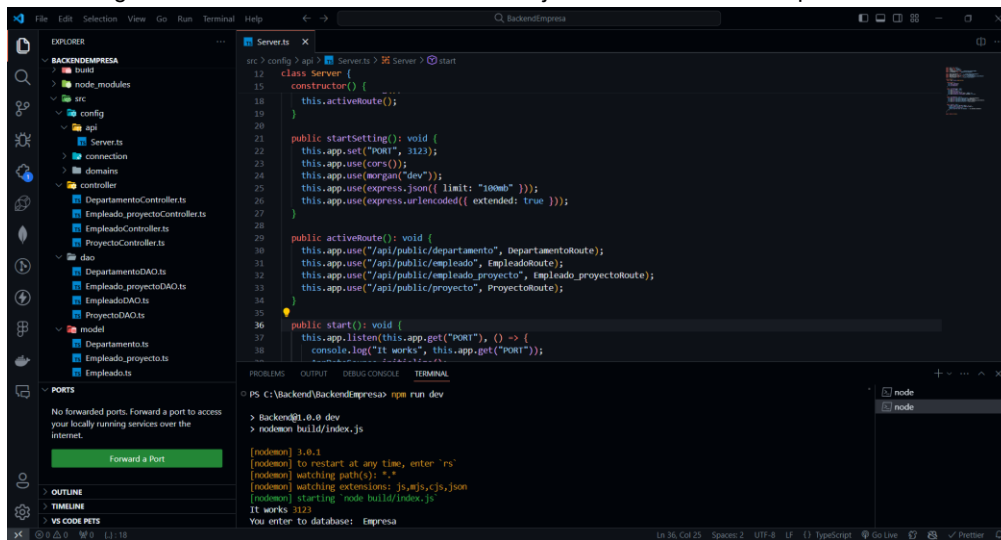
A la vez, en la Figura 146, se muestra la ejecución del comando `npm run dev` en una nueva terminal. Este comando inicia el servidor de desarrollo del backend, permitiendo que nuestra aplicación se ejecute localmente con capacidades de recarga automática.

Figura 145. Interfaz de Visual Code con ejecución de comando “npm run build”



Fuente: Autores

Figura 146. Interfaz de Visual Code con ejecución de comando “npm run dev”



Fuente: Autores

- **Pruebas a los EndPoint generados:**

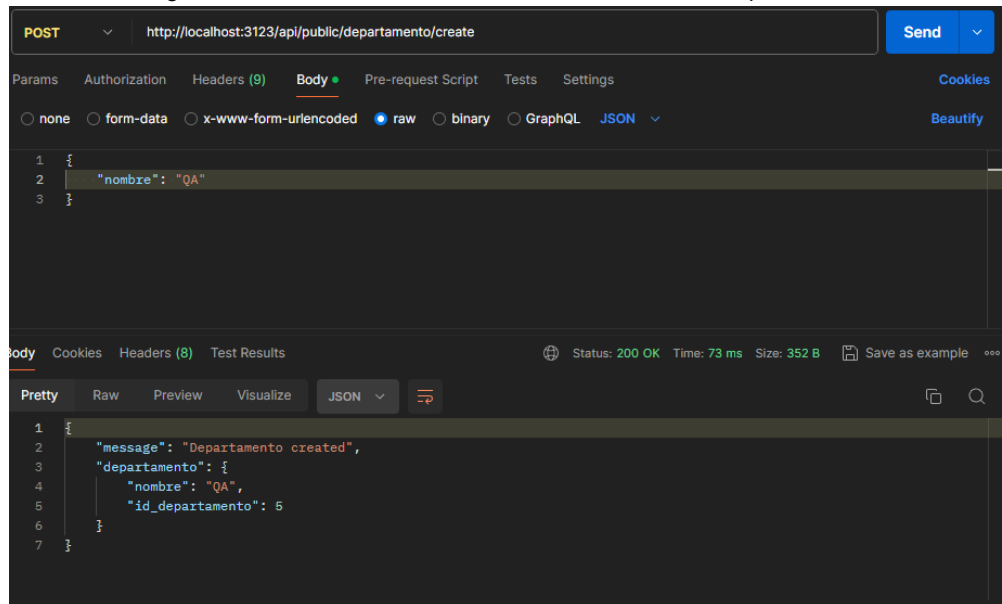
Estas pruebas son cruciales para asegurar el correcto funcionamiento de nuestras API y garantizar la integridad de las operaciones CRUD en relación con las tablas de nuestra base de datos de conexión.

Cada una de las tablas en la base de datos ha sido sometida a pruebas específicas para validar las siguientes operaciones:

- **Create (Crear):** Utilizando solicitudes POST, se crearon nuevos registros en cada tabla de la base de datos. Estas pruebas verifican que los datos sean correctamente persistidos y que se generen identificadores únicos para cada nuevo registro.
- **Read (Leer):** Empleando solicitudes GET, se recuperaron registros existentes de cada tabla. Estas pruebas validan que los datos se devuelvan correctamente.
- **Update (Actualizar):** A través de solicitudes PUT, se modificaron registros existentes en las tablas. Estas pruebas garantizan que los cambios en los datos se reflejen de manera adecuada y que las actualizaciones sean precisas y consistentes.
- **Delete (Eliminar):** Mediante solicitudes DELETE, se han eliminado registros seleccionados en cada tabla. Estas pruebas verifican que los datos sean eliminados correctamente.

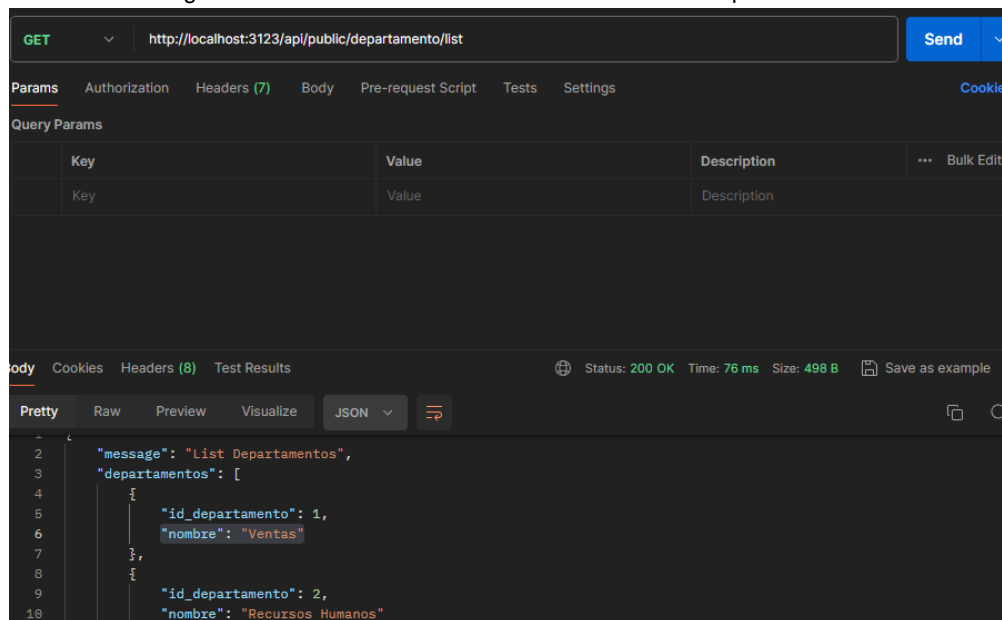
A continuación, en la Figura 147, Figura 148, Figura 149 y Figura 150 se podrá visualizar las pruebas a los endpoints generados de la tabla "Departamento".

Figura 147. Prueba de EndPoint “Create” de la tabla “departamento”



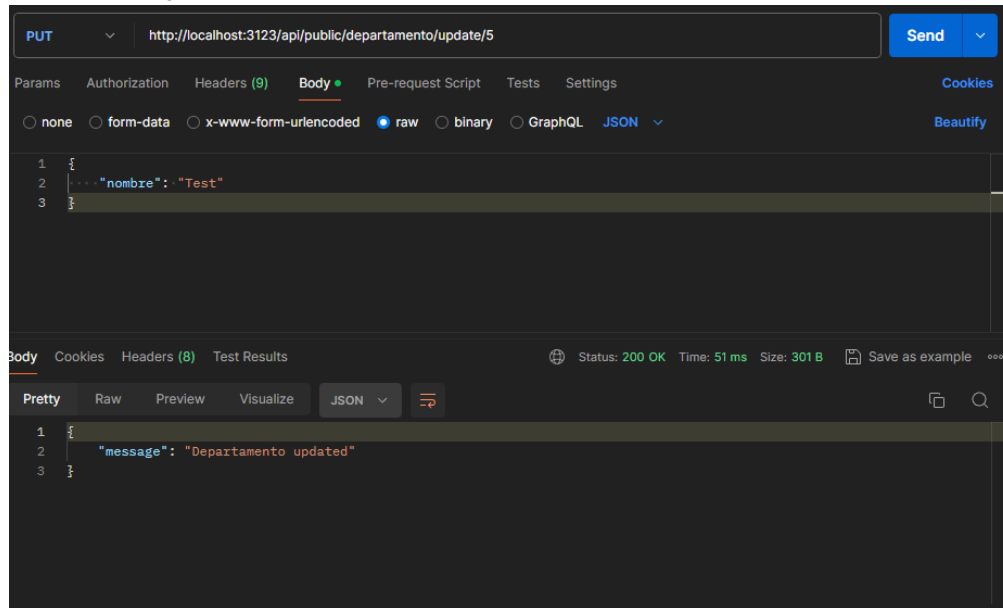
Fuente: Autores

Figura 148. Prueba de EndPoint “List” de la tabla “departamento”



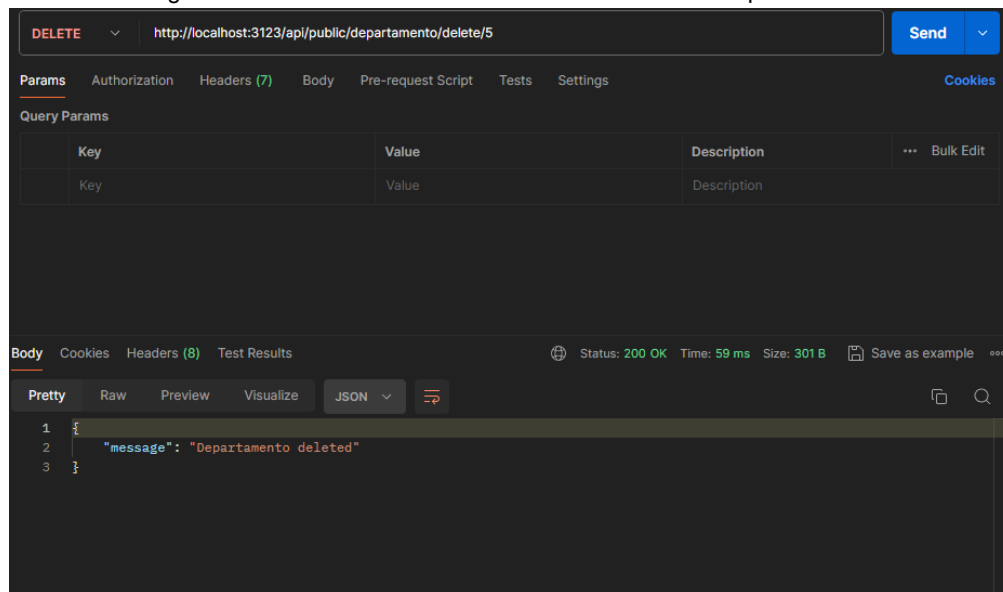
Fuente: Autores

Figura 149. Prueba de EndPoint “Update” de la tabla “departamento”



Fuente: Autores

Figura 150. Prueba de EndPoint “Delete” de la tabla “departamento”



Fuente: Autores

A continuación, en la Figura 151, Figura 152, Figura 153 y Figura 154 se podrá visualizar las pruebas a los endpoints generados de la tabla “empleado”.

Figura 151. Prueba de EndPoint “Create” de la tabla “empleado”

```
POST http://localhost:3123/api/public/empleado/create

{
  "nombre": "Felipe",
  "apellido": "Cruz",
  "email": "felipeCruz@example.com",
  "direccion": "Calle Principal 785",
  "genero": "Masculino",
  "fecha_nacimiento": "1998-05-15",
  "telefono": "315697541",
  "sueldo": 3000,
  "id_departamento": 3
}

{
  "message": "Empleado created",
  "empleado": {
    "nombre": "Felipe",
    "apellido": "Cruz",
```

Fuente: Autores

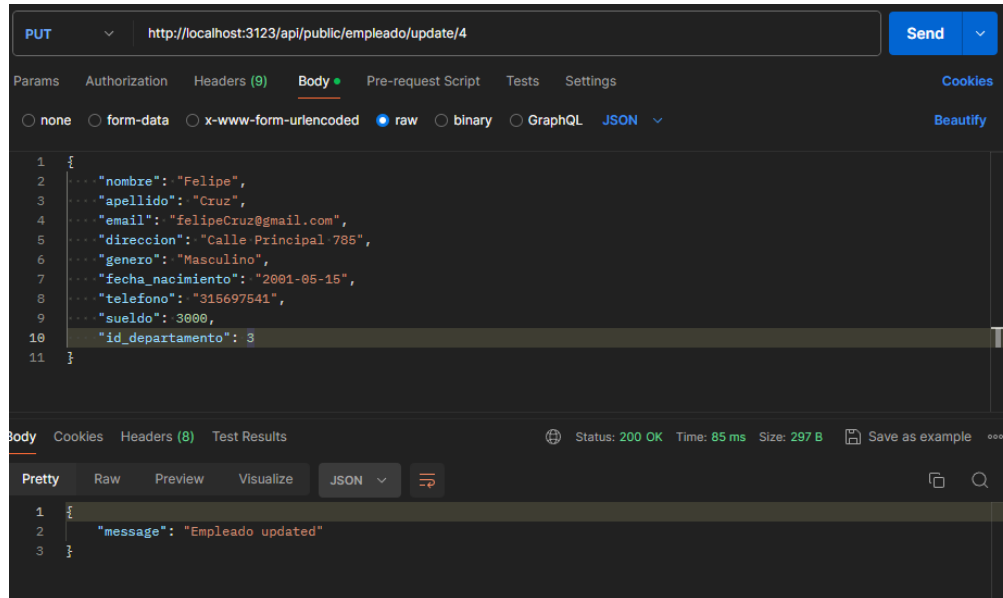
Figura 152. Prueba de EndPoint “List” de la tabla “empleado”

```
GET http://localhost:3123/api/public/empleado/list

{
  "message": "List Empleados",
  "empleados": [
    {
      "id_empleado": 1,
      "nombre": "Juan",
      "apellido": "Perez",
      "email": "juan@example.com",
      "direccion": "Calle Principal 123",
      "genero": "Masculino",
      "fecha_nacimiento": "1990-05-15",
      "telefono": "123456789",
      "sueldo": 3000,
      "id_departamento": 3
    },
    {
      "id_empleado": 2,
      "nombre": "Maria",
      "apellido": "López",
      "email": "maria@example.com",
      "direccion": "Avenida Central 456",
```

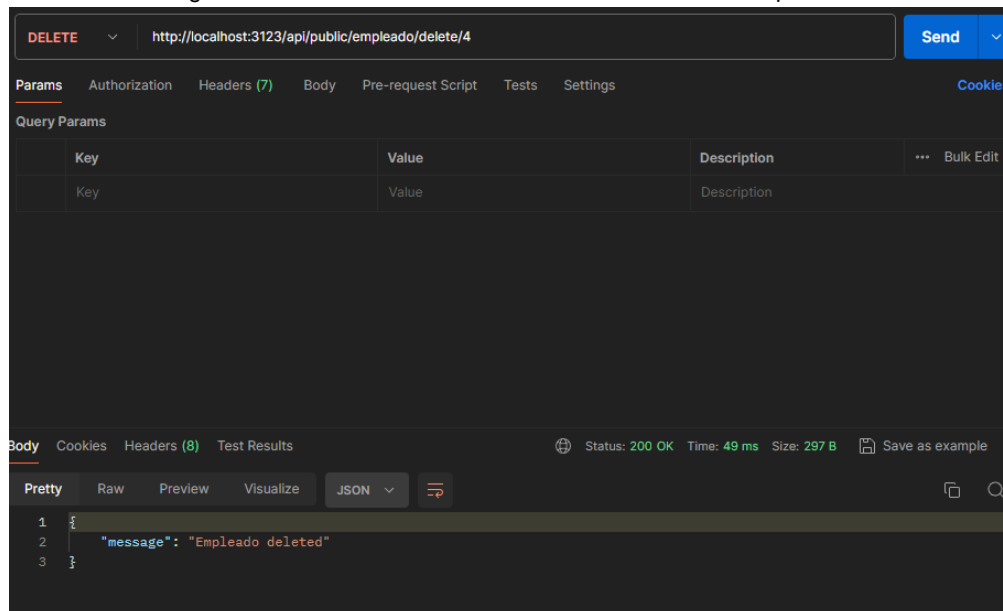
Fuente: Autores

Figura 153. Prueba de EndPoint "Update" de la tabla "empleado"



Fuente: Autores

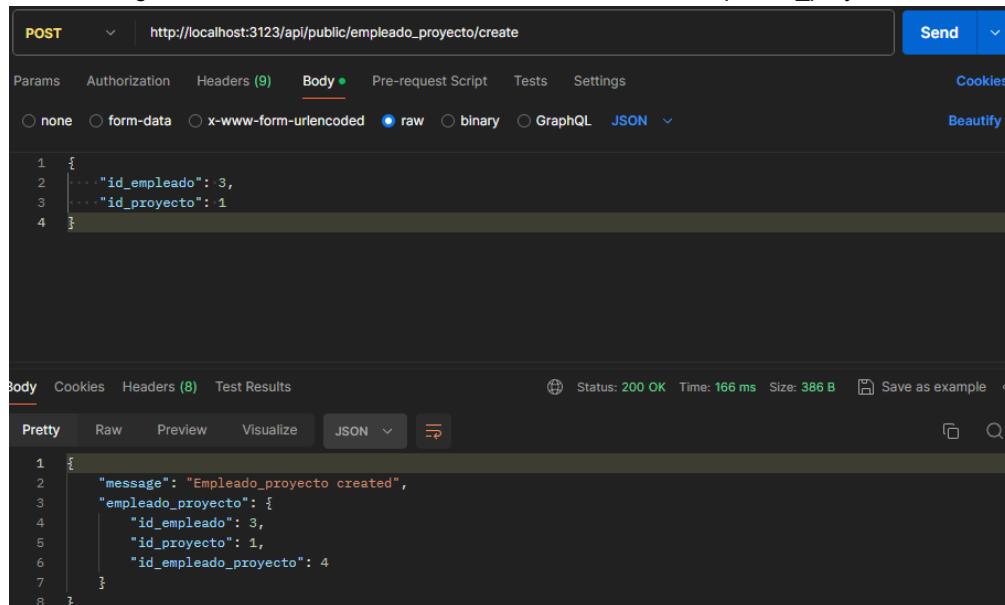
Figura 154. Prueba de EndPoint "Delete" de la tabla "empleado"



Fuente: Autores

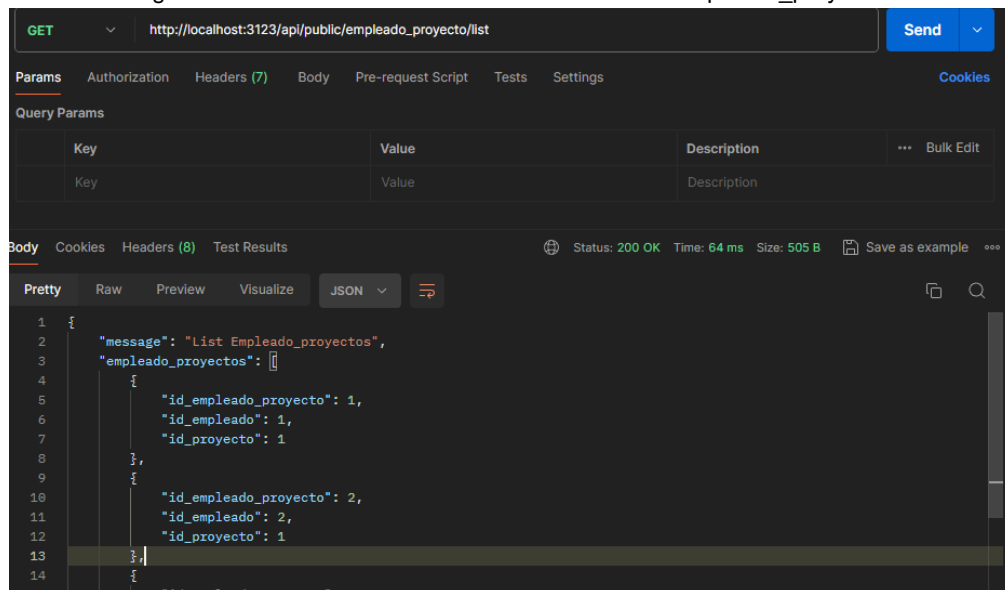
A continuación, en la Figura 155, Figura 156, Figura 157 y Figura 158 se podrá visualizar las pruebas a los endpoints generados de la tabla “empleado\_proyecto”.

Figura 155. Prueba de EndPoint “Create” de la tabla “empleado\_proyecto”



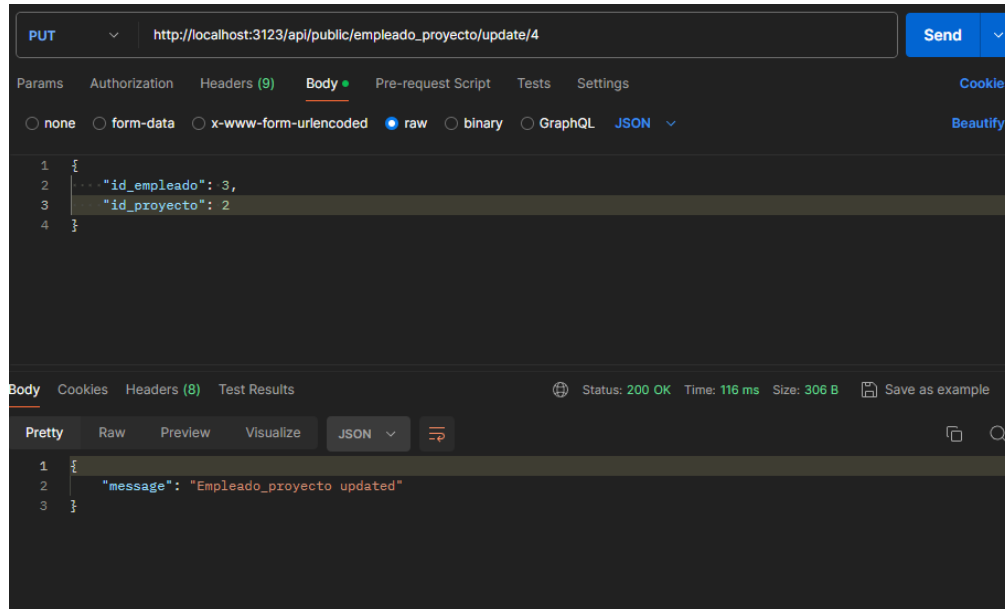
Fuente: Autores

Figura 156. Prueba de EndPoint “List” de la tabla “empleado\_proyecto”



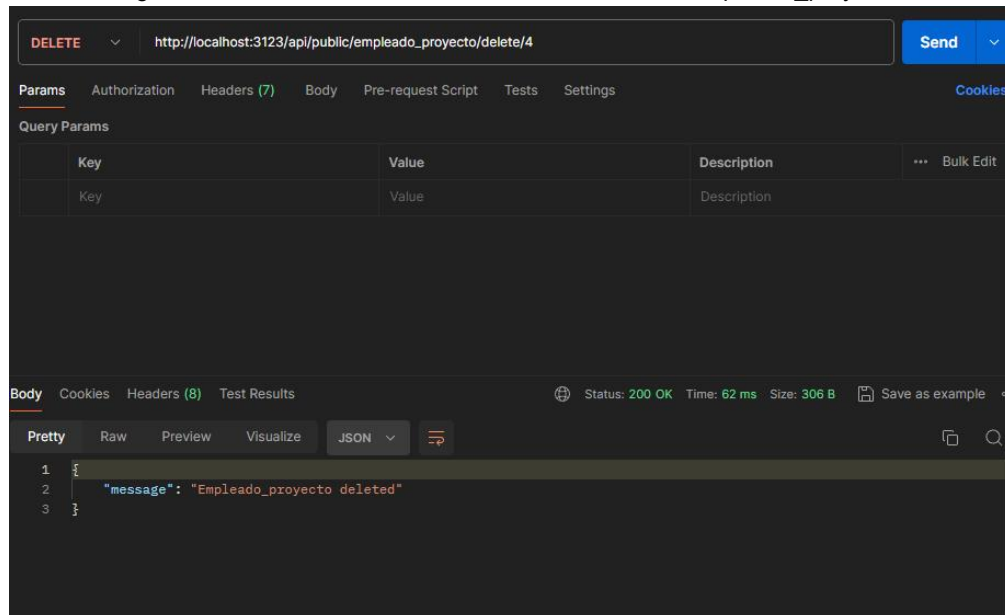
Fuente: Autores

Figura 157. Prueba de EndPoint “Update” de la tabla “empleado\_proyecto”



Fuente: Autores

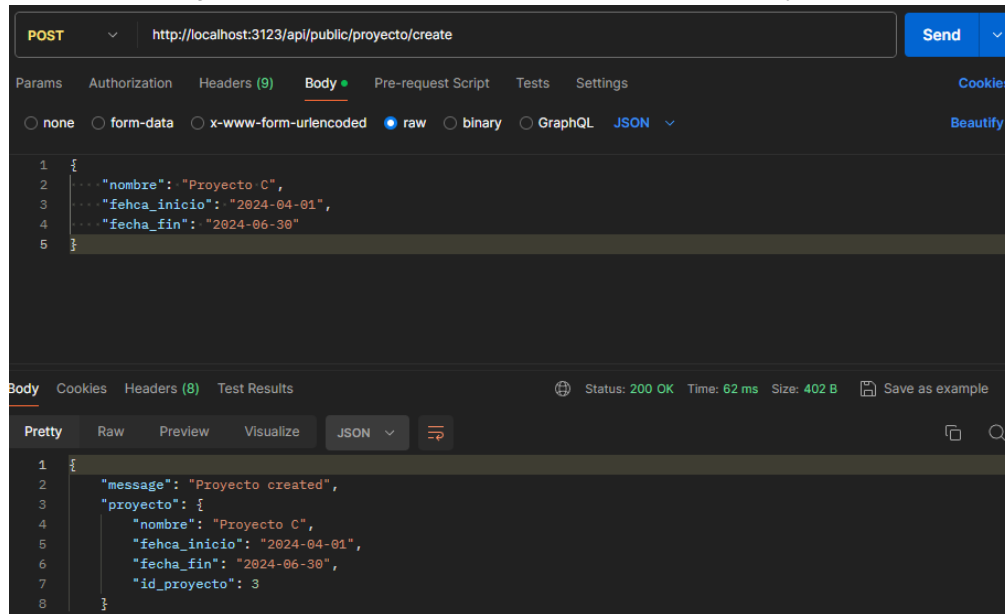
Figura 158. Prueba de EndPoint “Delete” de la tabla “empleado\_proyecto”



Fuente: Autores

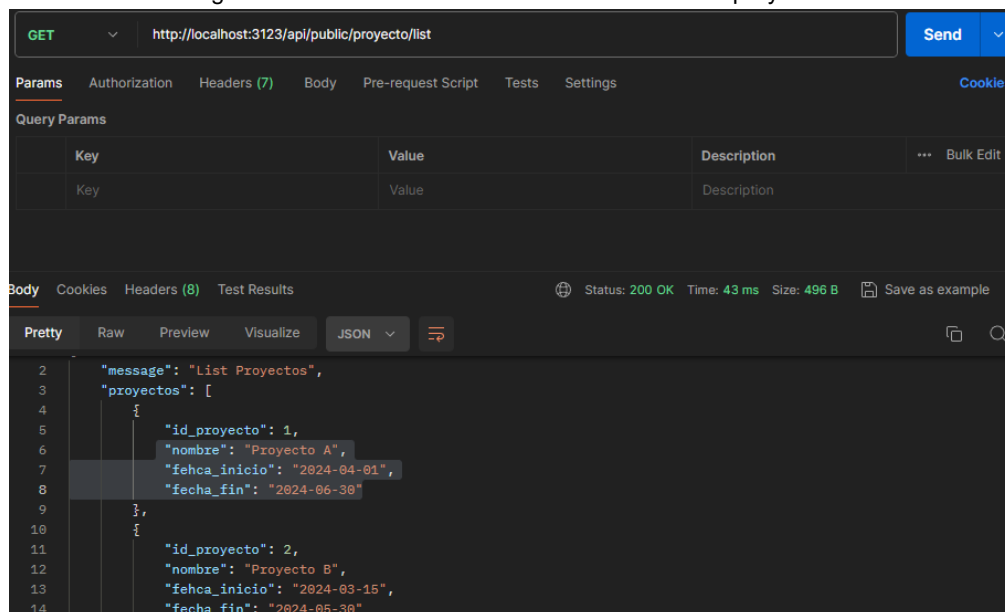
A continuación, en la Figura 159, Figura 160, Figura 161 y Figura 162 se podrá visualizar las pruebas a los endpoints generados de la tabla “proyecto”.

Figura 159. Prueba de EndPoint “Create” de la tabla “proyecto”



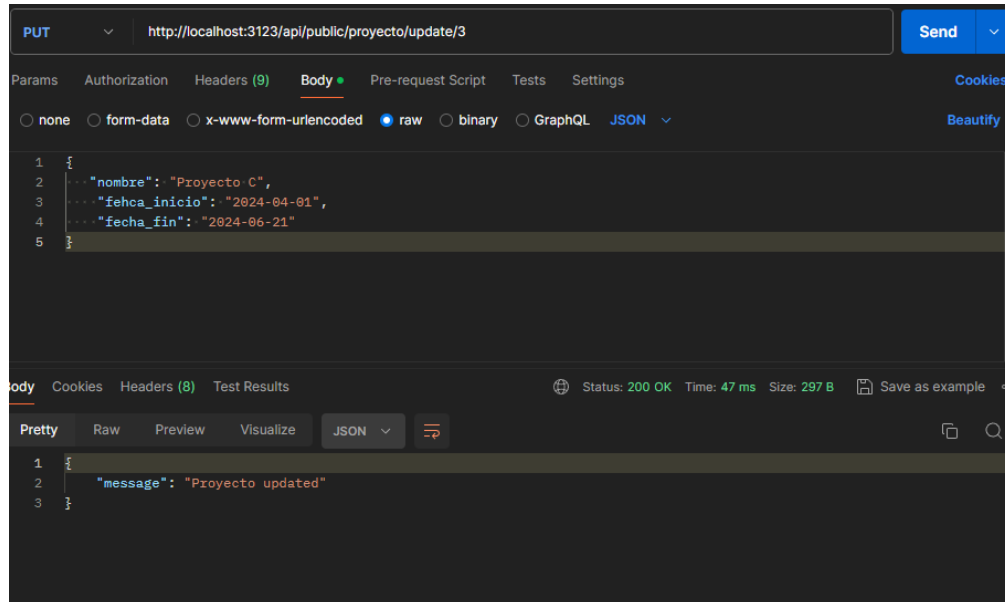
Fuente: Autores

Figura 160. Prueba de EndPoint “List” de la tabla “proyecto”



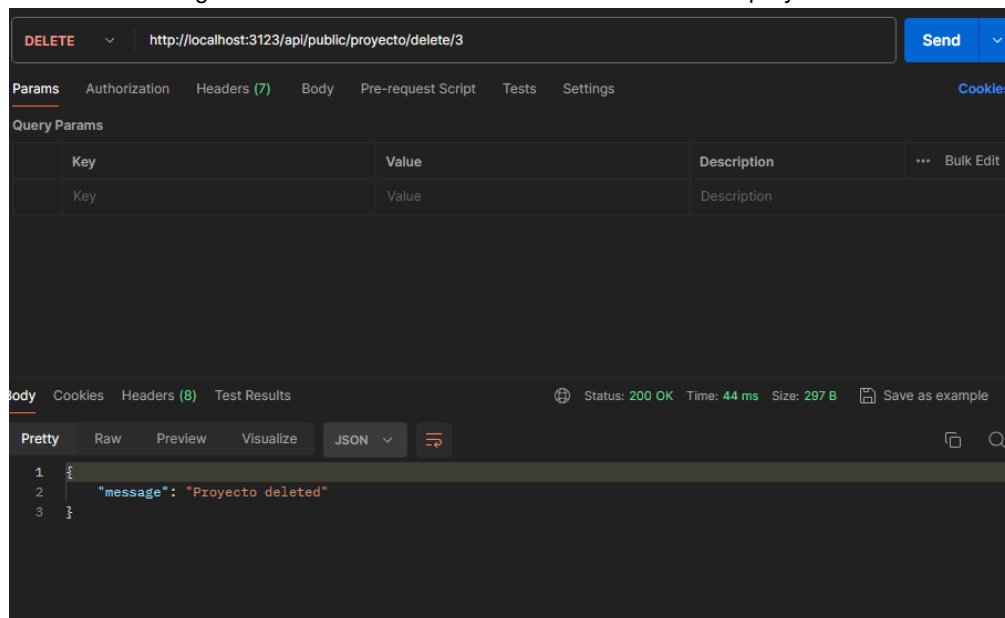
Fuente: Autores

Figura 161. Prueba de EndPoint "Update" de la tabla "proyecto"



Fuente: Autores

Figura 162. Prueba de EndPoint "Delete" de la tabla "proyecto"

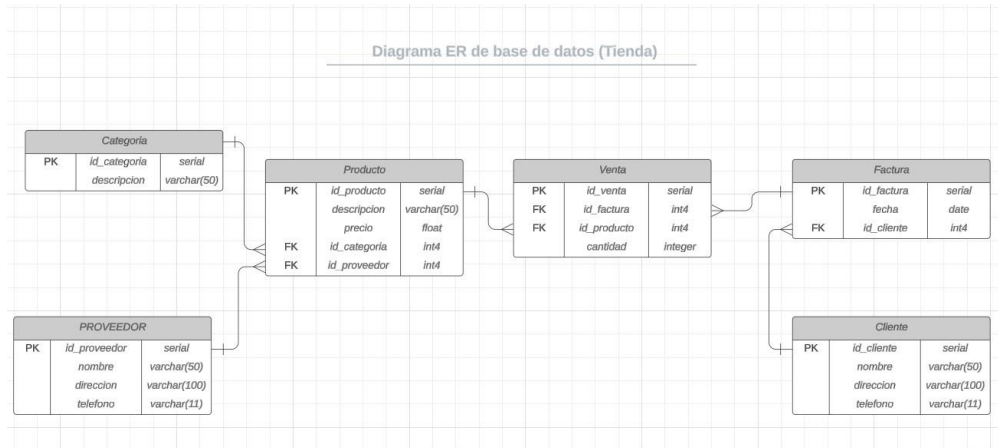


Fuente: Autores

### 10.19. Anexo: Plan de pruebas Base de Datos (“Tienda”)

En la Figura 163 se encuentra el diagrama Entidad-Relación (ER) de la base de datos "Tienda", la cual será utilizada como el enfoque principal para las pruebas del generador de backend. Este diagrama representa la estructura conceptual de la base de datos, mostrando las entidades principales, sus atributos y las relaciones entre ellas.

Figura 163. Diagrama Entidad-Relación (ER) de la base de datos “Tienda”



Fuente: Autores

Se diligencia los campos del formulario de conexión con las credenciales de conexión a la base de datos con la cual queremos generar el backend, como se muestra en la Figura 164.

Figura 164. Formulario de conexión a base de datos

Database connection

postgres

localhost

Tienda

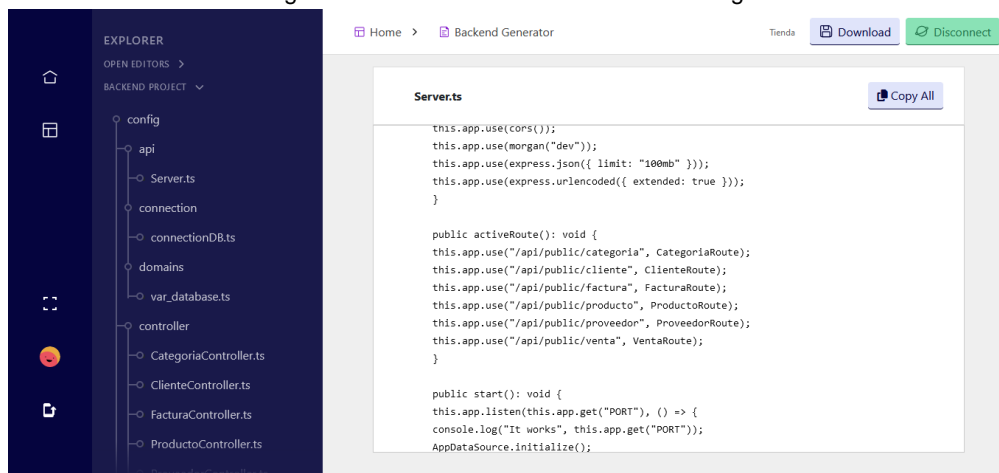
..... 5432

Connect

Fuente: Autores

Lograr establecer con éxito la conexión a nuestra base de datos principal. Esto es fundamental para avanzar en nuestro proyecto de desarrollo de software. En la Figura 165Figura 142, se puede observar el proceso de conexión a la base de datos y la generación automática de código basada en su estructura.

Figura 165. Generación automática de código



Fuente: Autores

La descarga del proyecto que contiene el código backend necesario para nuestro desarrollo. Este paso es fundamental para comenzar a trabajar en la implementación de las funcionalidades principales.

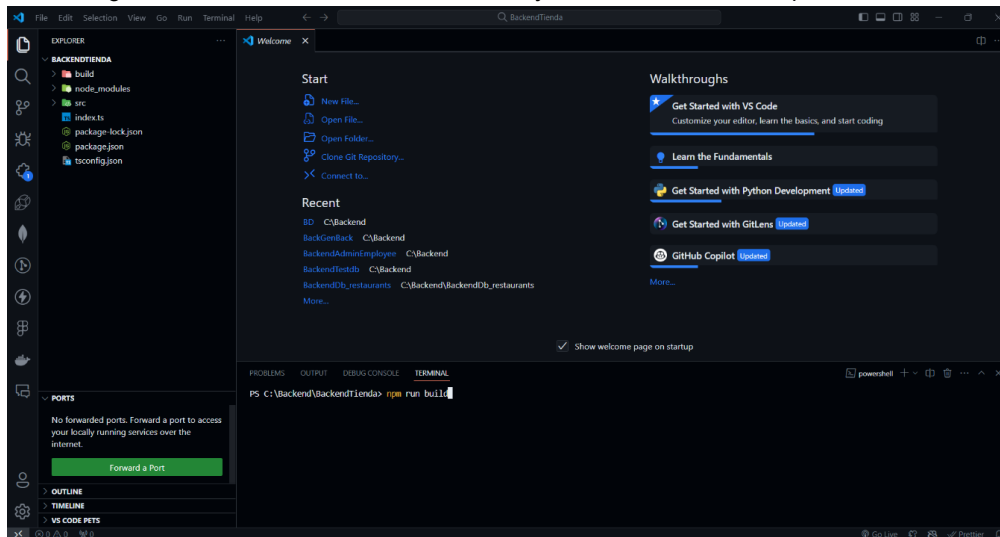
En la Figura 166 y Figura 143, se muestra la interfaz de Visual Studio Code (VS Code), nuestra herramienta principal para el desarrollo de software. Esta interfaz proporciona un entorno integrado y altamente personalizable que facilita la navegación y modificación del código fuente.

A la vez, en la Figura 167, se muestra la ejecución del comando npm i en la terminal integrada de VS Code. Este comando, abreviatura de npm install, se utiliza para instalar todas las dependencias del proyecto especificadas en el archivo package.json.



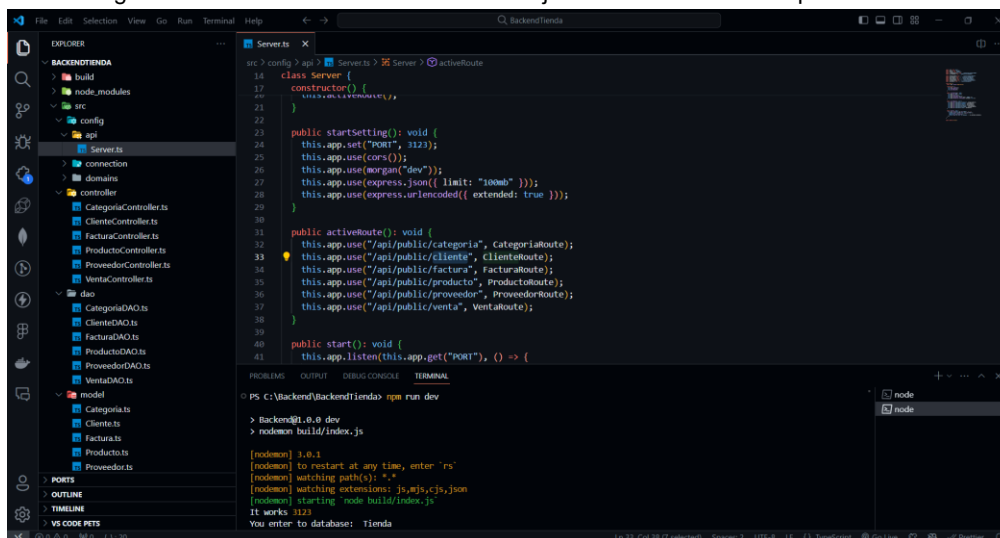
A la vez, en la Figura 169, se muestra la ejecución del comando `npm run dev` en una nueva terminal. Este comando inicia el servidor de desarrollo del backend, permitiendo que nuestra aplicación se ejecute localmente con capacidades de recarga automática.

Figura 168. Interfaz de Visual Code con ejecución de comando “npm run build”



Fuente: Autores

Figura 169. Interfaz de Visual Code con ejecución de comando “npm run dev”



Fuente: Autores

- **Pruebas a los EndPoint generados:**

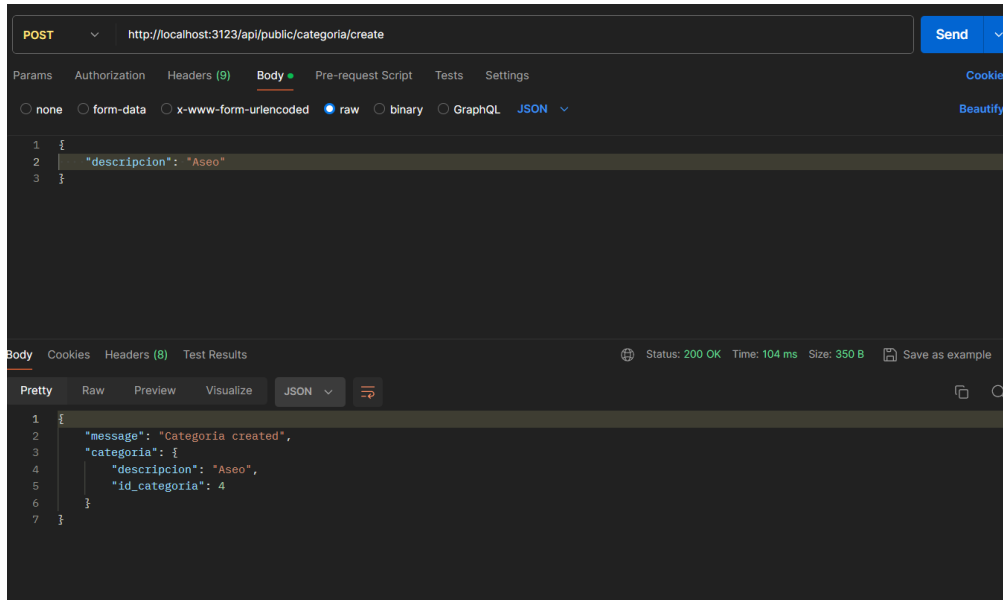
Estas pruebas son cruciales para asegurar el correcto funcionamiento de nuestras API y garantizar la integridad de las operaciones CRUD en relación con las tablas de nuestra base de datos de conexión.

Cada una de las tablas en la base de datos ha sido sometida a pruebas específicas para validar las siguientes operaciones:

- **Create (Crear):** Utilizando solicitudes POST, se crearon nuevos registros en cada tabla de la base de datos. Estas pruebas verifican que los datos sean correctamente persistidos y que se generen identificadores únicos para cada nuevo registro.
- **Read (Leer):** Empleando solicitudes GET, se recuperaron registros existentes de cada tabla. Estas pruebas validan que los datos se devuelvan correctamente.
- **Update (Actualizar):** A través de solicitudes PUT, se modificaron registros existentes en las tablas. Estas pruebas garantizan que los cambios en los datos se reflejen de manera adecuada y que las actualizaciones sean precisas y consistentes.
- **Delete (Eliminar):** Mediante solicitudes DELETE, se han eliminado registros seleccionados en cada tabla. Estas pruebas verifican que los datos sean eliminados correctamente.

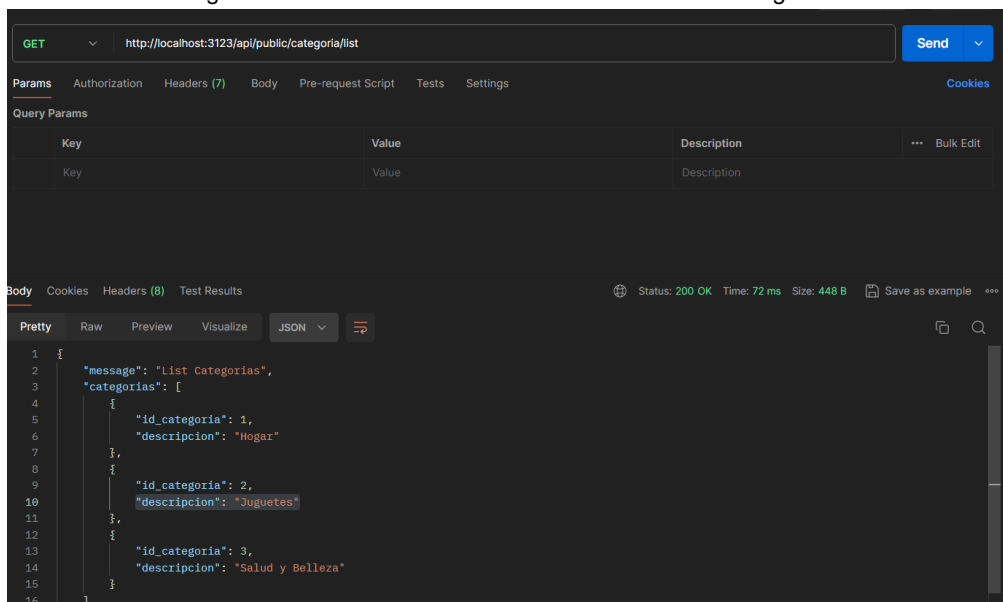
A continuación, en la Figura 170, Figura 171, Figura 172 y Figura 173 se podrá visualizar las pruebas a los endpoints generados de la tabla "categoria".

Figura 170. Prueba de EndPoint "Create" de la tabla "categoria"



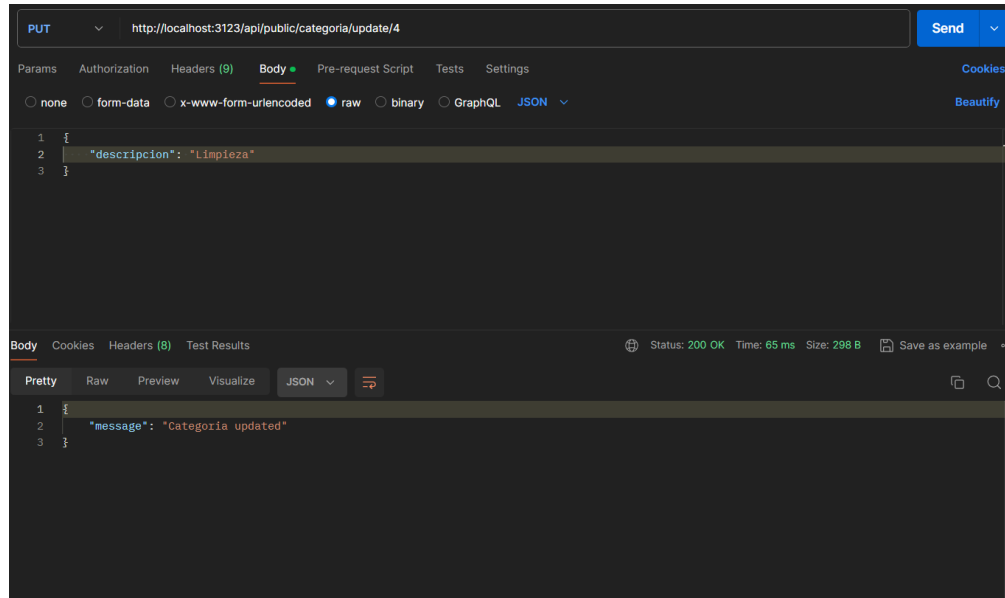
Fuente: Autores

Figura 171. Prueba de EndPoint "List" de la tabla "categoria"



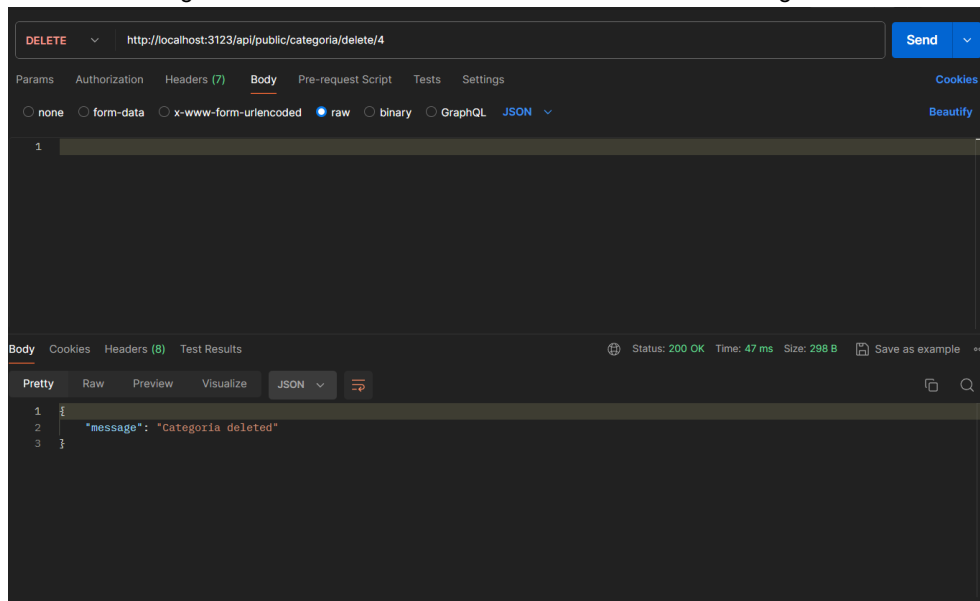
Fuente: Autores

Figura 172. Prueba de EndPoint “Update” de la tabla “categoria”



Fuente: Autores

Figura 173. Prueba de EndPoint “Delete” de la tabla “categoria”



Fuente: Autores

A continuación, en la Figura 174, Figura 175, Figura 176 y Figura 177 se podrá visualizar las pruebas a los endpoints generados de la tabla “cliente”.

Figura 174. Prueba de EndPoint "Create" de la tabla "cliente"

```
POST http://localhost:3123/api/public/cliente/create

{
  "nombre": "Abel Ruiz",
  "direccion": "Avenida Pria, Ciudad C",
  "telefono": "1112223333"
}
```

```
{
  "message": "Cliente created",
  "cliente": {
    "nombre": "Abel Ruiz",
    "direccion": "Avenida Pria, Ciudad C",
    "telefono": "1112223333",
    "id_cliente": 4
  }
}
```

Fuente: Autores

Figura 175. Prueba de EndPoint "List" de la tabla "cliente"

```
GET http://localhost:3123/api/public/cliente/list
```

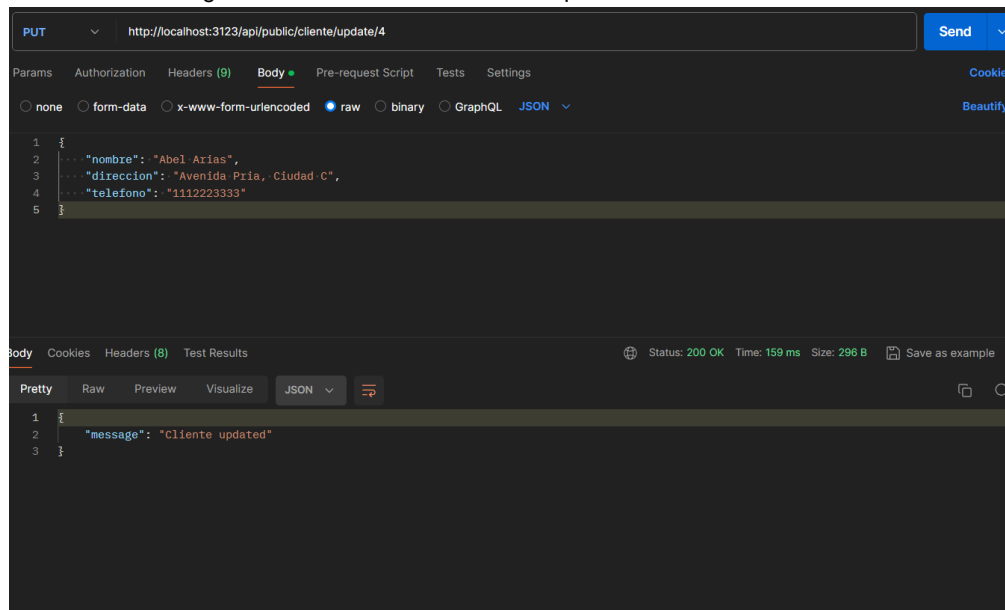
Key	Value	Description
Key	Value	Description

```
{
  "message": "List Clientes",
  "clientes": [
    {
      "id_cliente": 1,
      "nombre": "Ana Martinez",
      "direccion": "Avenida Libertad, Ciudad A",
      "telefono": "1112223333"
    },
    {
      "id_cliente": 2,
      "nombre": "Pedro Garcia",

```

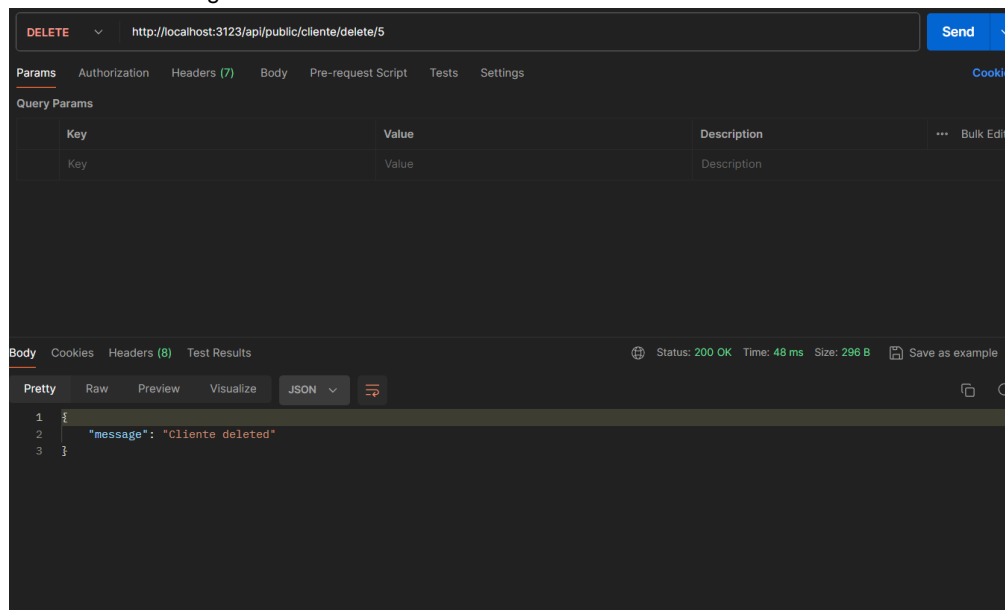
Fuente: Autores

Figura 176. Prueba de EndPoint “Update” de la tabla “cliente”



Fuente: Autores

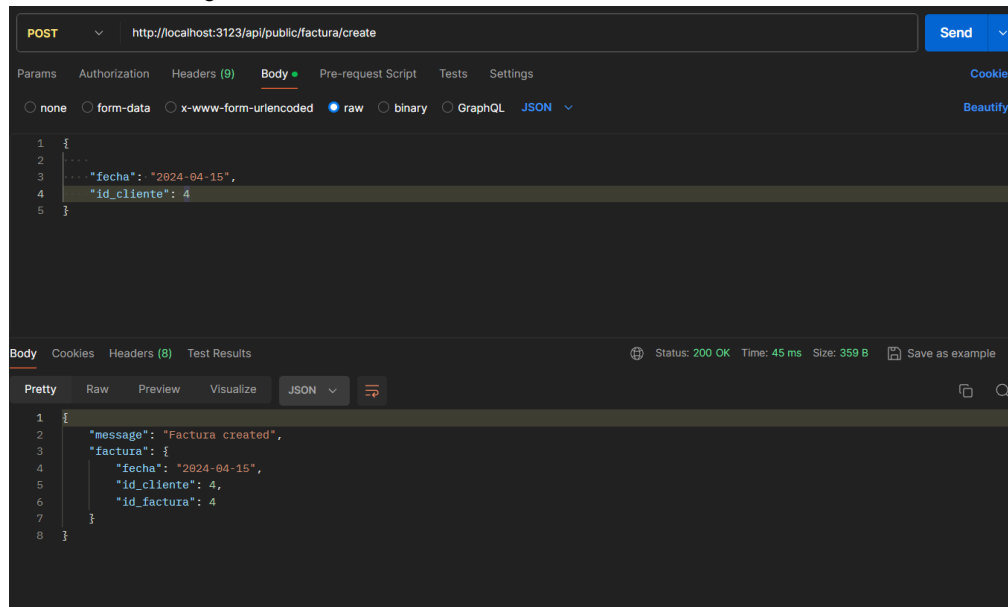
Figura 177. Prueba de EndPoint “Delete” de la tabla “cliente”



Fuente: Autores

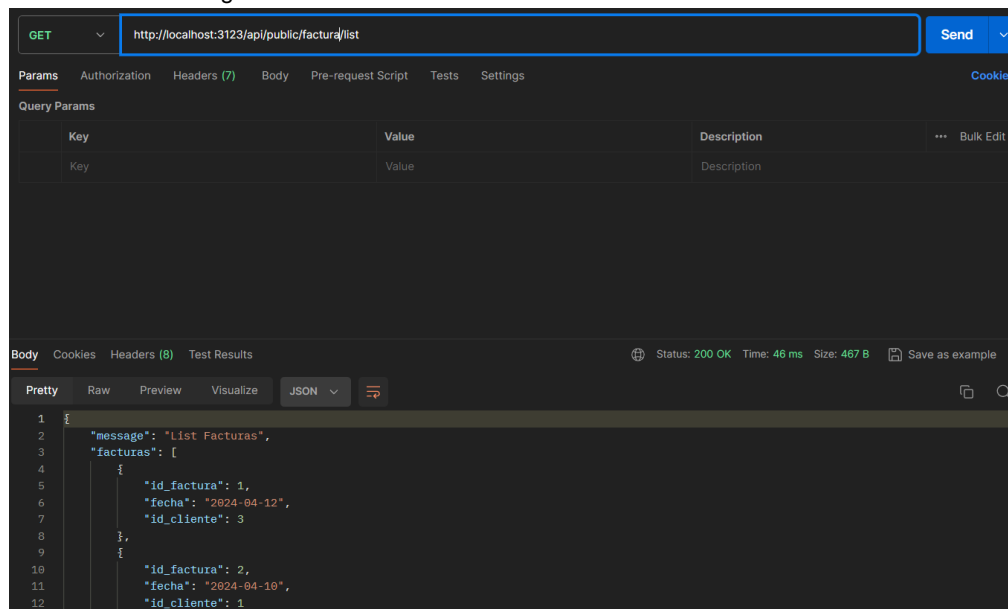
A continuación, en la Figura 178, Figura 179, Figura 180 y Figura 181 se podrá visualizar las pruebas a los endpoints generados de la tabla “factura”.

Figura 178. Prueba de EndPoint "Create" de la tabla "factura"



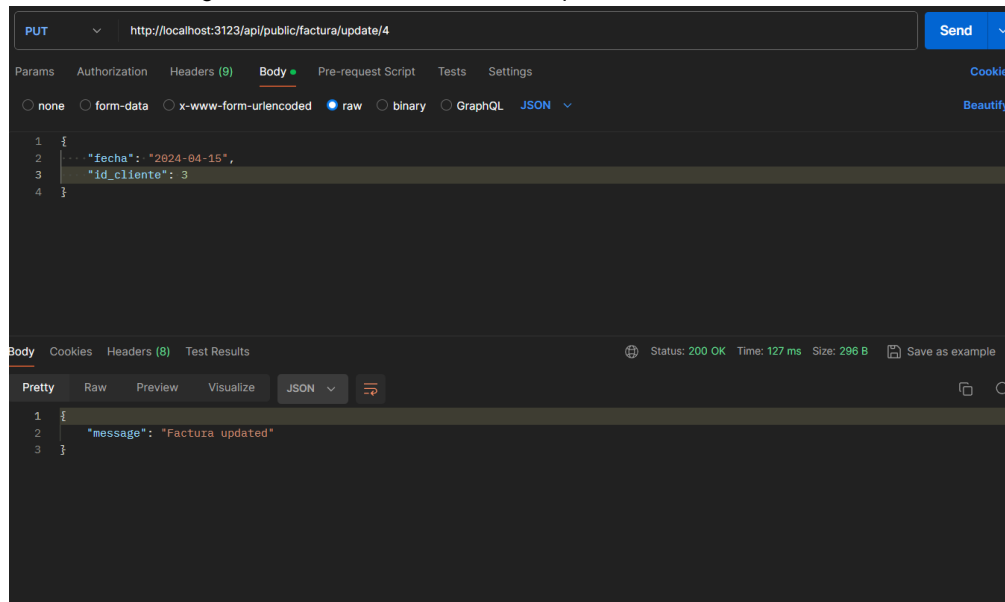
Fuente: Autores

Figura 179. Prueba de EndPoint "List" de la tabla "factura"



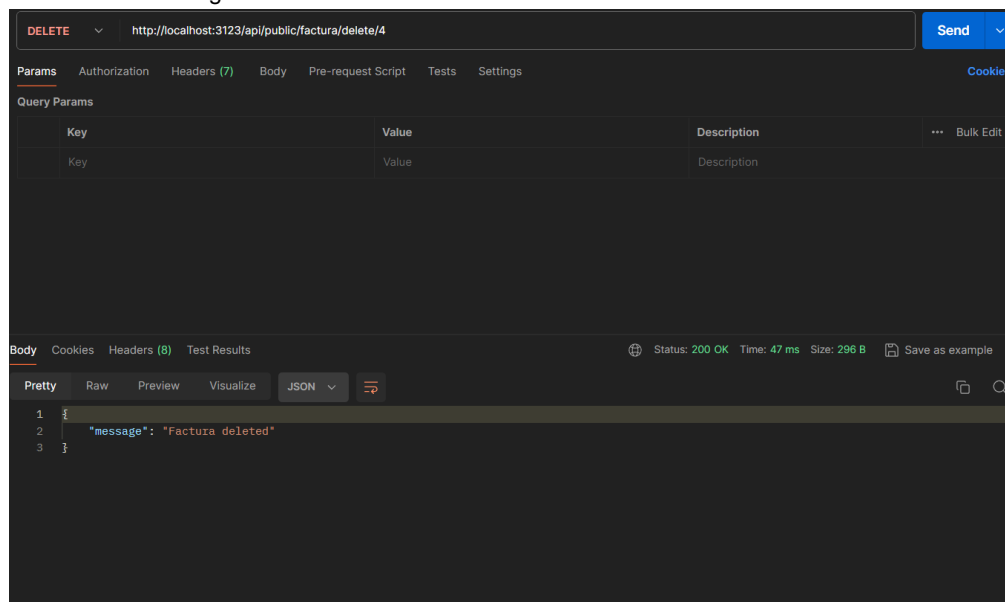
Fuente: Autores

Figura 180. Prueba de EndPoint “Update” de la tabla “factura”



Fuente: Autores

Figura 181. Prueba de EndPoint “Delete” de la tabla “factura”



Fuente: Autores

A continuación, en la Figura 182, Figura 183, Figura 184 y Figura 185 se podrá visualizar las pruebas a los endpoints generados de la tabla “producto”.

Figura 182. Prueba de EndPoint “Create” de la tabla “producto”

```
POST http://localhost:3123/api/public/producto/create

{
  "descripcion": "Carne",
  "precio": 24.99,
  "id_categoria": 1,
  "id_proveedor": 1
}

{
  "message": "Producto created",
  "producto": {
    "descripcion": "Carne",
    "precio": 24.99,
    "id_categoria": 1,
    "id_proveedor": 1,
    "id_producto": 5
  }
}
```

Fuente: Autores

Figura 183. Prueba de EndPoint “List” de la tabla “producto”

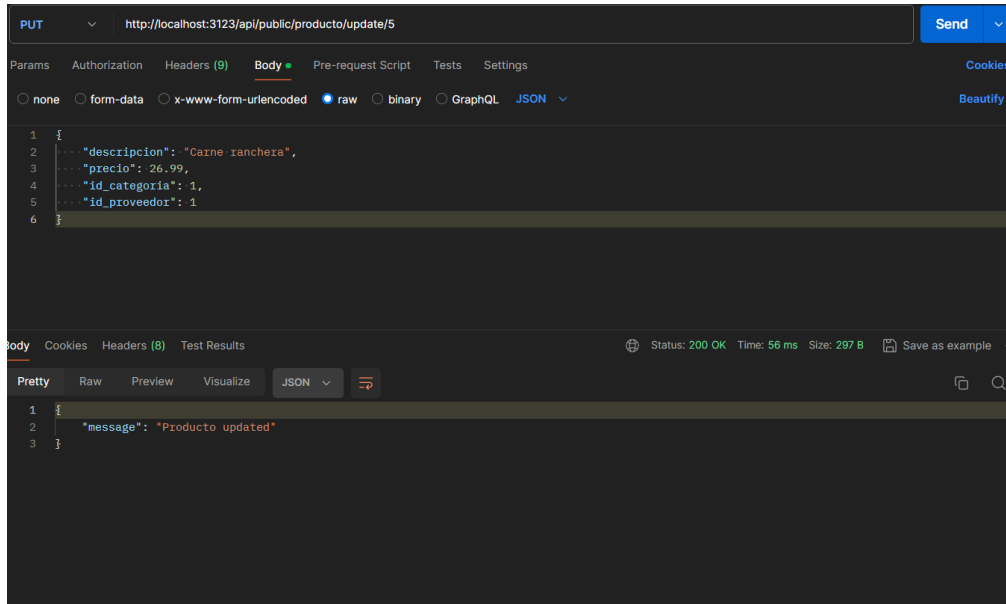
```
GET http://localhost:3123/api/public/producto/list

{
  "message": "List Productos",
  "productos": [
    {
      "id_producto": 1,
      "descripcion": "Lámpara de Mesa",
      "precio": 24.99,
      "id_categoria": 1,
      "id_proveedor": 1
    },
    {
      "id_producto": 2,

```

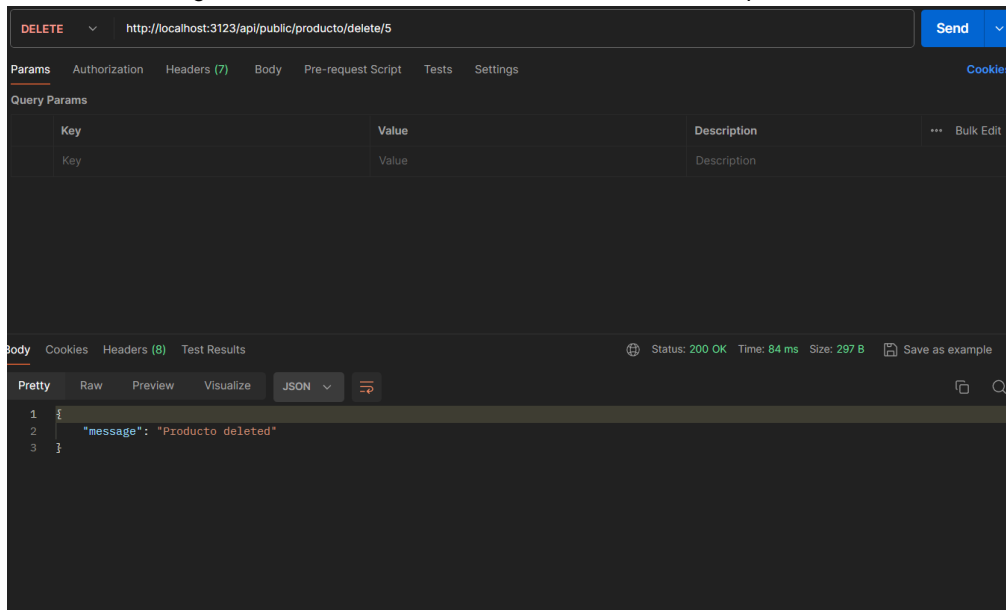
Fuente: Autores

Figura 184. Prueba de EndPoint “Update” de la tabla “producto”



Fuente: Autores

Figura 185. Prueba de EndPoint “Delete” de la tabla “producto”



Fuente: Autores

A continuación, en la Figura 186, Figura 187, Figura 188 y Figura 189 se podrá visualizar las pruebas a los endpoints generados de la tabla “proveedor”.

Figura 186. Prueba de EndPoint “Create” de la tabla “proveedor”

```
POST http://localhost:3123/api/public/proveedor/create

{
  "nombre": "Proveedor D",
  "direccion": "Calle Menor, Ciudad D",
  "telefono": "1239874561"
}

{
  "message": "Proveedor created",
  "proveedor": {
    "nombre": "Proveedor D",
    "direccion": "Calle Menor, Ciudad D",
    "telefono": "1239874561",
    "id_proveedor": 3
  }
}
```

Fuente: Autores

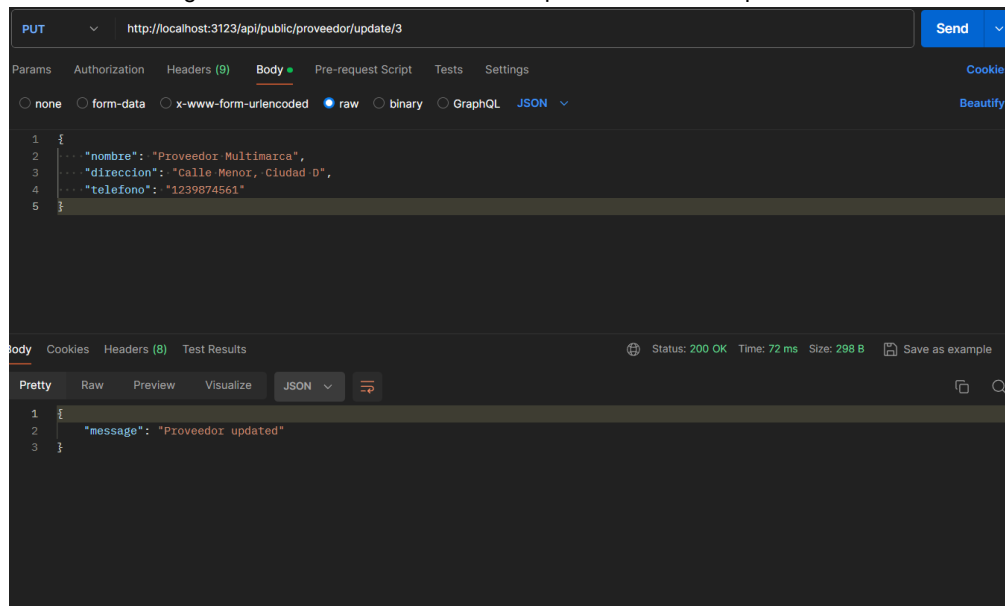
Figura 187. Prueba de EndPoint “List” de la tabla “proveedor”

```
GET http://localhost:3123/api/public/proveedor/list

{
  "message": "List Proveedores",
  "proveedores": [
    {
      "id_proveedor": 1,
      "nombre": "Proveedor C",
      "direccion": "Calle Mayor, Ciudad D",
      "telefono": "1239874561"
    },
    {
      "id_proveedor": 2,
      "nombre": "Proveedor D",
      "direccion": "Calle Menor, Ciudad D",
      "telefono": "1239874561"
    }
  ]
}
```

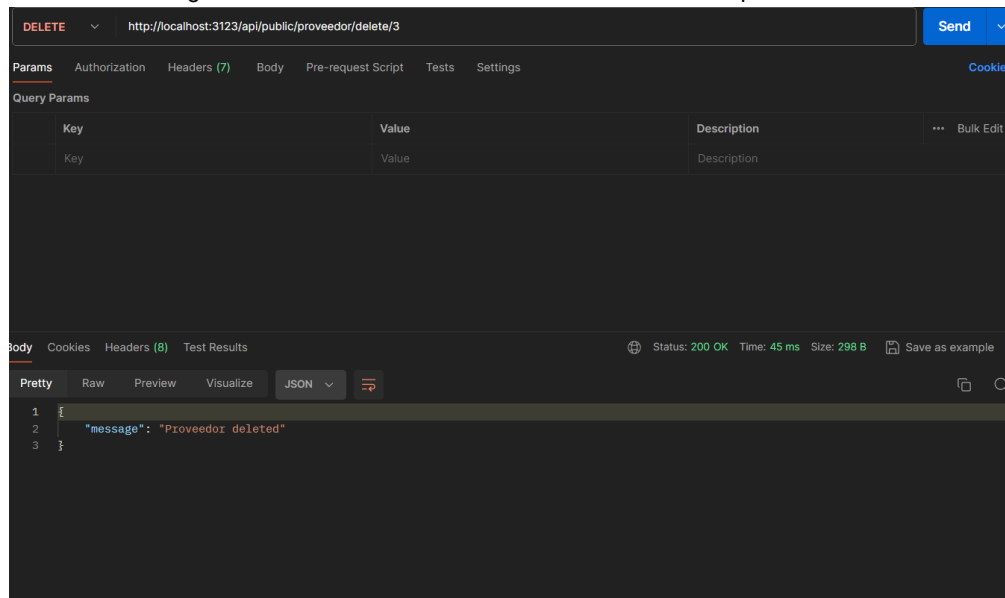
Fuente: Autores

Figura 188. Prueba de EndPoint “Update” de la tabla “proveedor”



Fuente: Autores

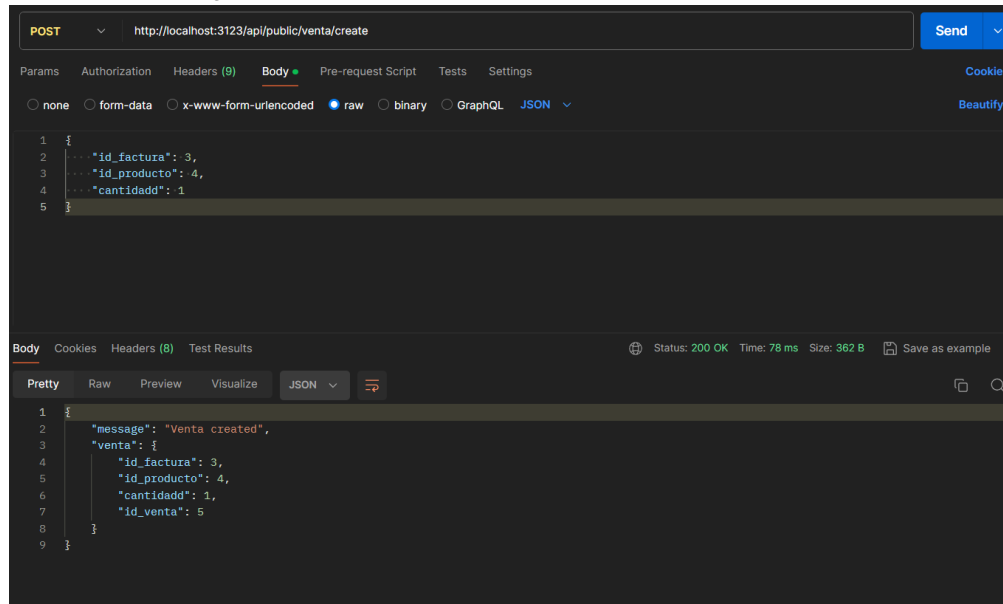
Figura 189. Prueba de EndPoint “Delete” de la tabla “proveedor”



Fuente: Autores

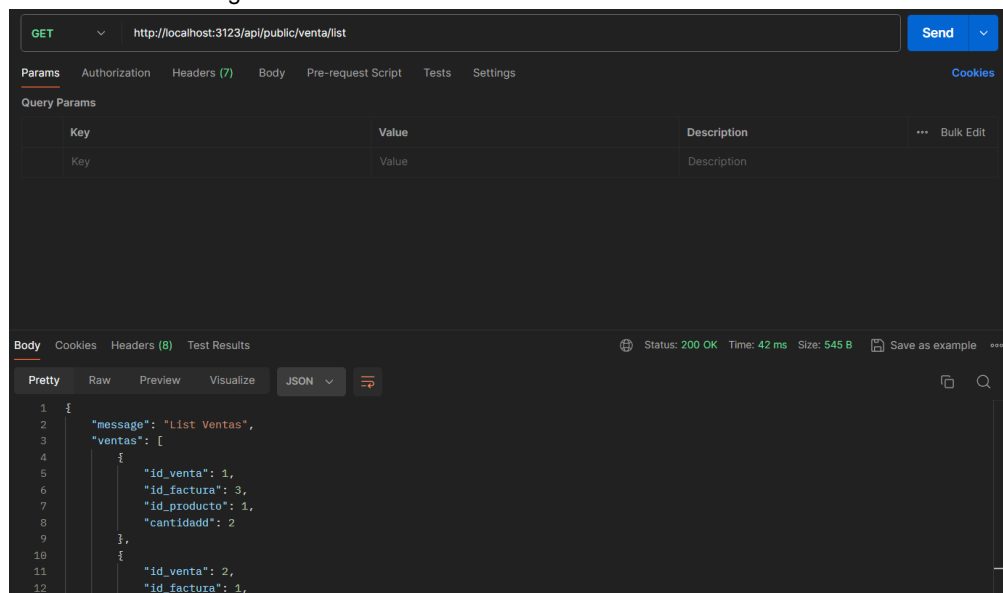
A continuación, en la Figura 182, Figura 183, Figura 184 y Figura 185 se podrá visualizar las pruebas a los endpoints generados de la tabla “venta”.

Figura 190. Prueba de EndPoint “Create” de la tabla “venta”



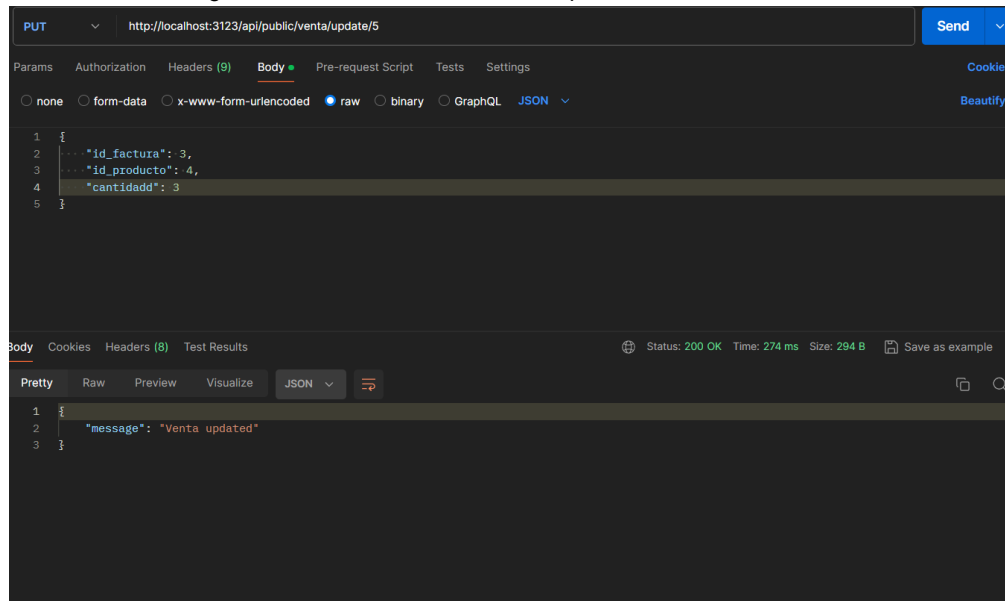
Fuente: Autores

Figura 191. Prueba de EndPoint “List” de la tabla “venta”



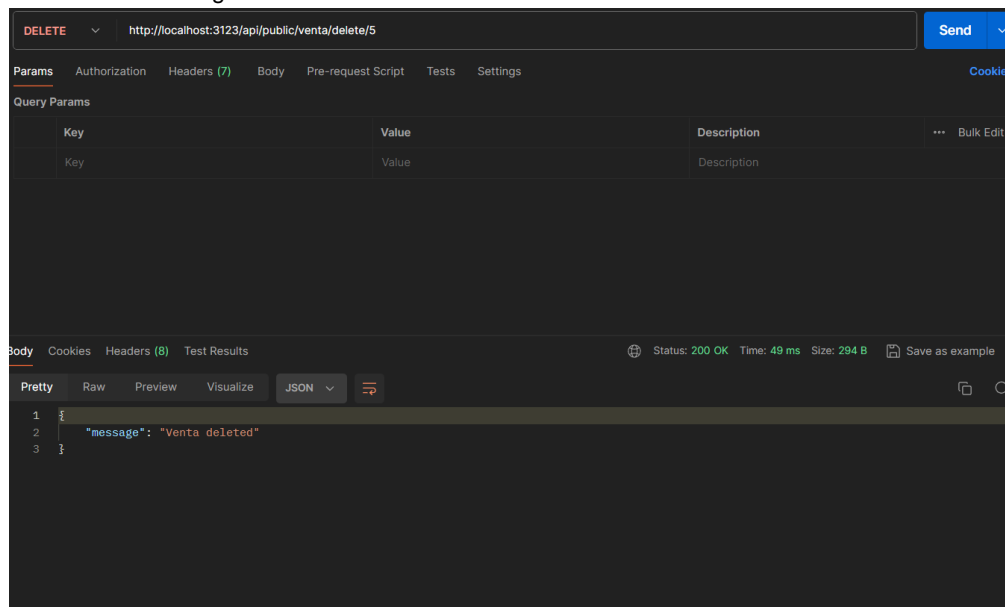
Fuente: Autores

Figura 192. Prueba de EndPoint "Update" de la tabla "venta"



Fuente: Autores

Figura 193. Prueba de EndPoint "Delete" de la tabla "venta"



Fuente: Autores

## 10.20. Anexo: Manual de usuario

Este anexo presenta el manual de usuario de la meta herramienta implementada. A continuación, se proporciona un paso a paso detallado sobre cómo utilizar la herramienta, así como los requisitos necesarios para su funcionamiento adecuado.

- **Requisitos del sistema**

Antes de comenzar a utilizar la herramienta, asegúrate de cumplir con los siguientes requisitos del sistema:

- ✓ **Sistema operativo:** Compatible con Windows, macOS y Linux.
- ✓ **Navegador web:** Se recomienda utilizar Google Chrome, Opera, Mozilla Firefox o Microsoft Edge.

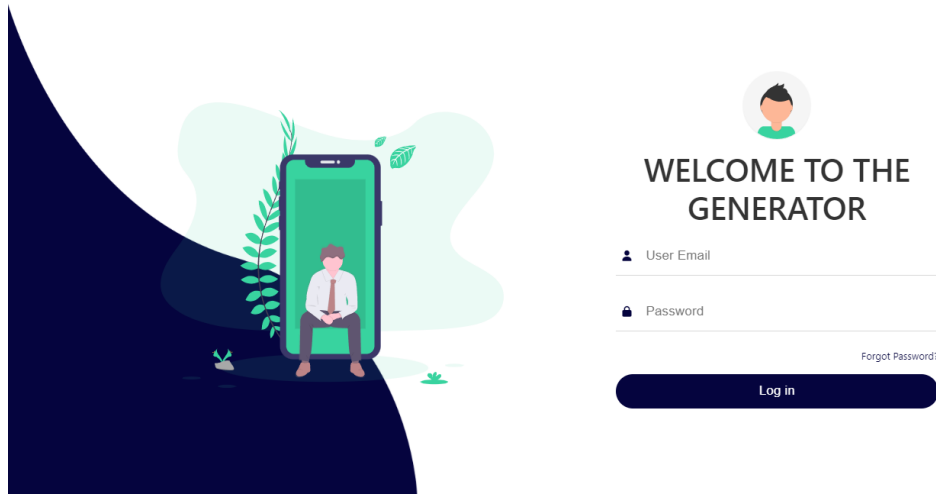
- **Manual de usuario**

<b>INICIO DE SESIÓN .....</b>	<b>170</b>
<b>MENÚ PRINCIPAL.....</b>	<b>170</b>
<b>FORMULARIO DE CONEXIÓN A LA BASE DE DATOS .....</b>	<b>171</b>
<b>VISTA DEL PROYECTO.....</b>	<b>172</b>
<b>DESCARGA O COPIA DEL PROYECTO .....</b>	<b>173</b>
<b>PUESTA EN MARCHA DEL PROYECTO DESCARGADO.....</b>	<b>174</b>
<b>DESCONEXIÓN DE LA BASE DE DATOS .....</b>	<b>177</b>
<b>CONFIGURACIÓN Y PERFIL.....</b>	<b>177</b>
<b>CERRAR SESIÓN.....</b>	<b>180</b>

### ✓ Inicio de sesión

Abre tu navegador web y accede a la URL proporcionada para la herramienta. Ingresa tus credenciales de inicio de sesión y haz clic en el botón "Iniciar Sesión".

Figura 194. Vista del inicio de sesión



Fuente: Autores

### ✓ Menú principal

Después de iniciar sesión, será dirigido al menú principal, donde encontrará un encabezado lateral con las siguientes opciones de navegación:

- **Menú:** Redirecciona a la página principal del sistema.
- **Formulario de conexión:** Acceda al formulario de conexión para la generación de código.
- **Pantalla completa:** Permite expandir la pantalla a pantalla completa.
- **Perfil del usuario:** Acceda a su perfil y visualiza su información.
- **Salir:** Cierra sesión y sale del sistema.

Figura 195. Vista del menú principal

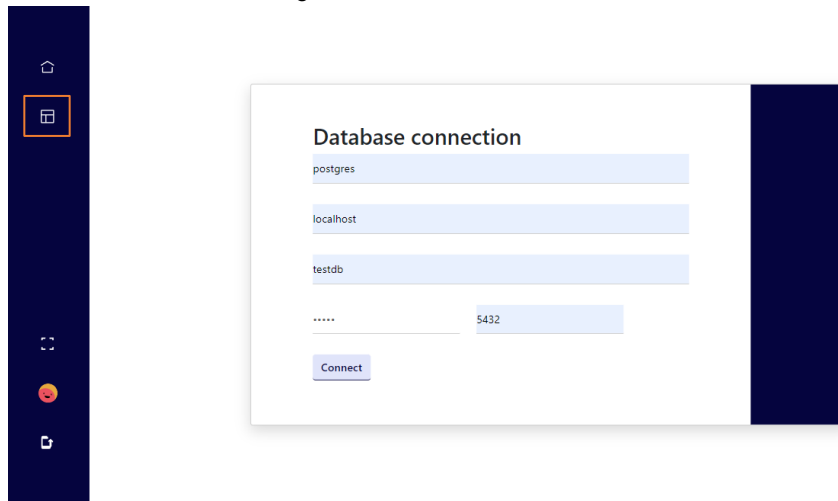


Fuente: Autores

✓ **Formulario de conexión a la base de datos:**

- En el menú principal, encontrará un menú lateral con varios iconos, incluyendo el menú principal, un formulario de conexión a la base de datos relacional, pantalla completa, perfil de usuario y cerrar sesión.
- Al hacer clic en el icono de formulario de conexión, el usuario accederá al formulario mostrado en la Figura 196. Aquí el usuario ingresará la información necesaria para la conexión a su base de datos relacional.
- Complete los campos requeridos, como la dirección del servidor, el nombre de usuario, la contraseña, el puerto y el nombre de la base de datos.
- Haga clic en el botón "Conectar" para establecer la conexión con la base de datos.

Figura 196. Vista de formulario de conexión



The screenshot shows a 'Database connection' form with the following fields and values:

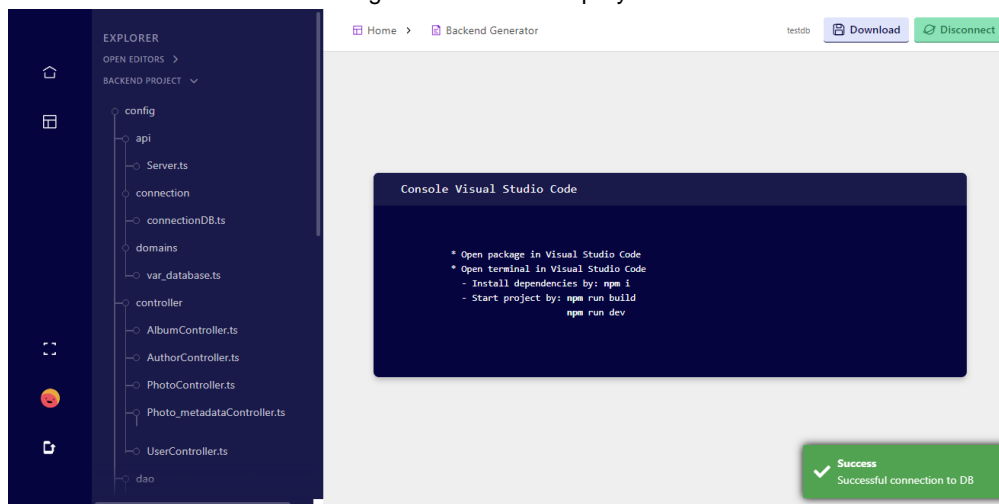
- Database name: postgres
- Host: localhost
- Database name: testdb
- Port: 5432
- Connect button

Fuente: Autores

✓ **Vista del proyecto:**

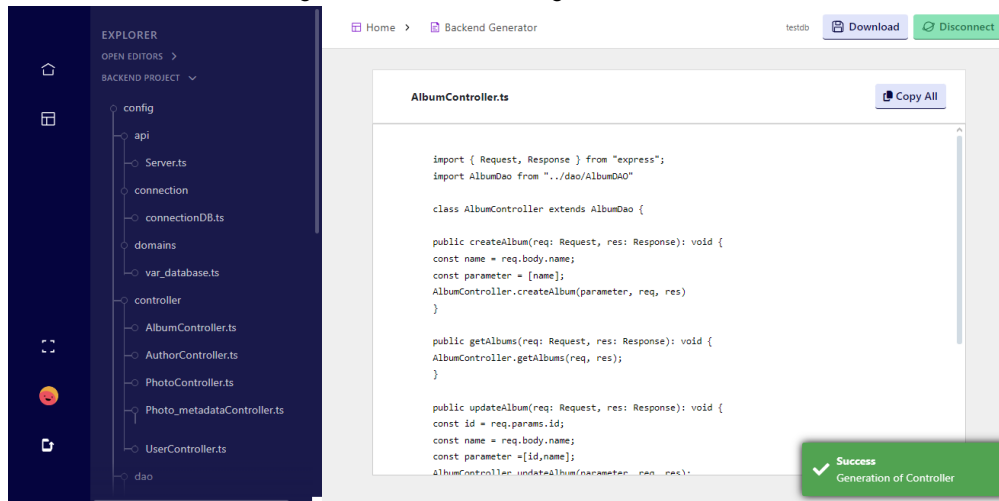
- Una vez que la conexión se haya establecido correctamente, accederá a la vista del proyecto. Aquí el usuario encontrará el árbol del proyecto y los comandos necesarios para iniciar tu proyecto después de descargarlo.
- Explore las diversas opciones disponibles como se muestra en la Figura 197, Figura 198 y en la Figura 199, que incluyen la visualización de código, la capacidad de copiar líneas o todo el código, y la descarga de este.

Figura 197. Vista del proyecto



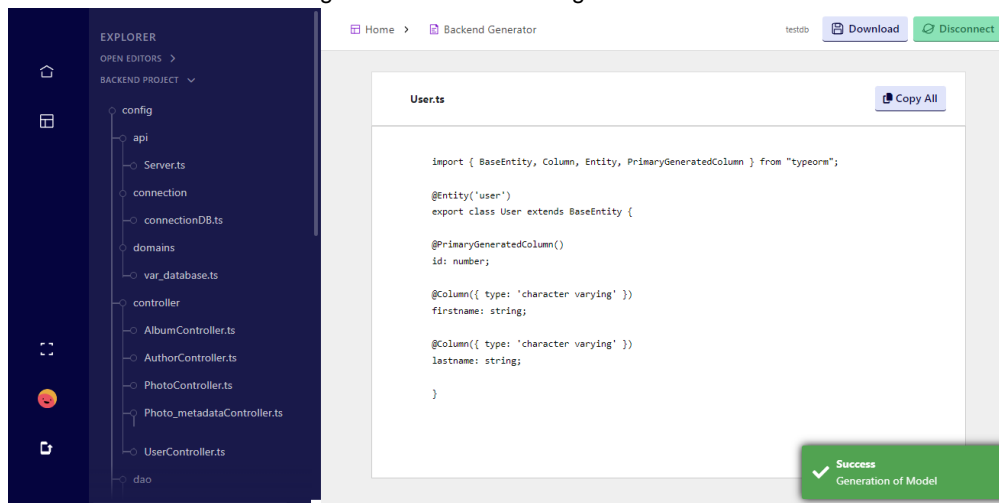
Fuente: Autores

Figura 198. Vista de código del controlador



Fuente: Autores

Figura 199. Vista de código del modelo

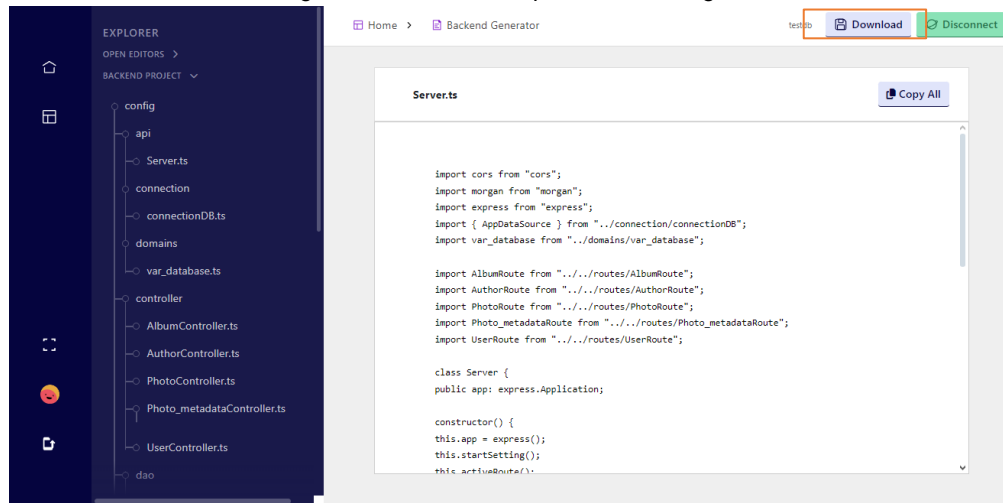


Fuente: Autores

✓ **Descarga o copia del proyecto:**

- Después de la revisión necesaria del proyecto, tendrás la opción de descargar el proyecto completo en un archivo ZIP como se muestra en la Figura 200.

Figura 200. Vista con opción de descargar

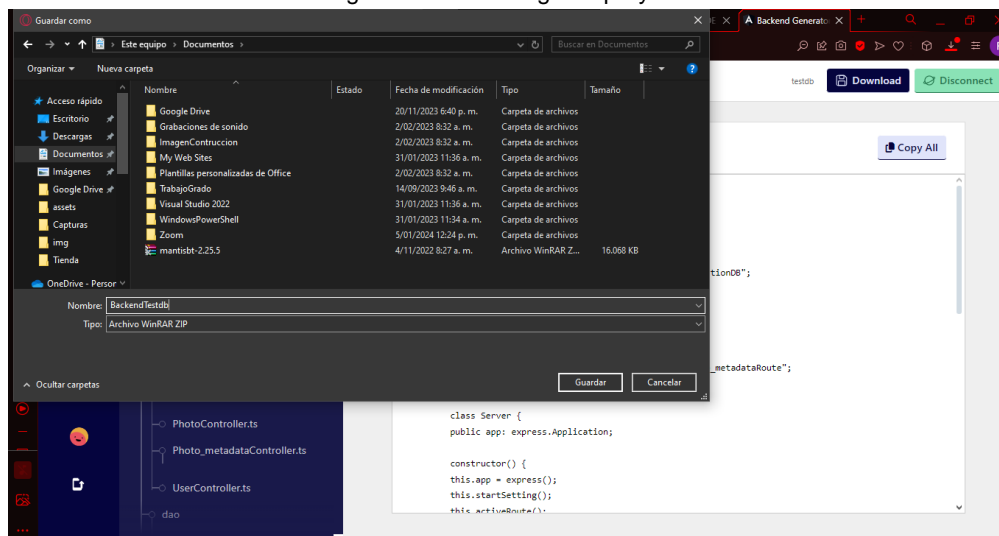


Fuente: Autores

✓ **Puesta en marcha del proyecto descargado:**

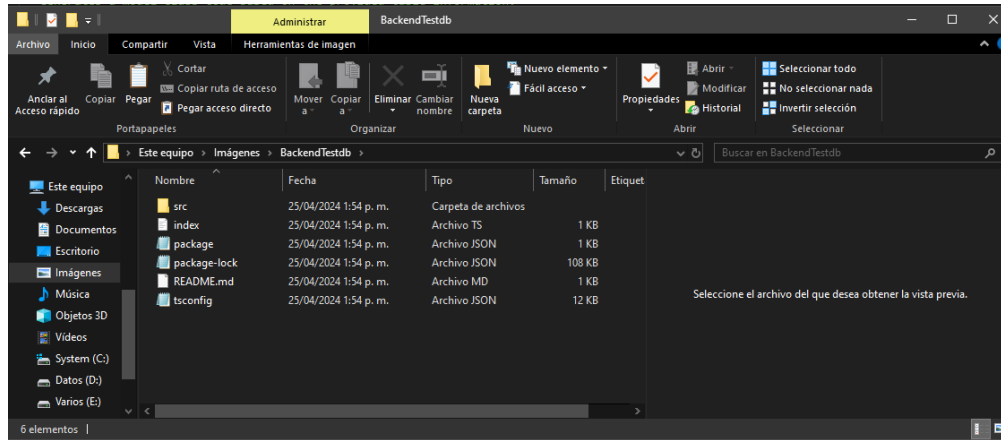
Una vez descargado el proyecto, descomprima el archivo ZIP en su sistema local como se muestra en la Figura 201, Figura 202 y en la Figura 203.

Figura 201. Descarga de proyecto



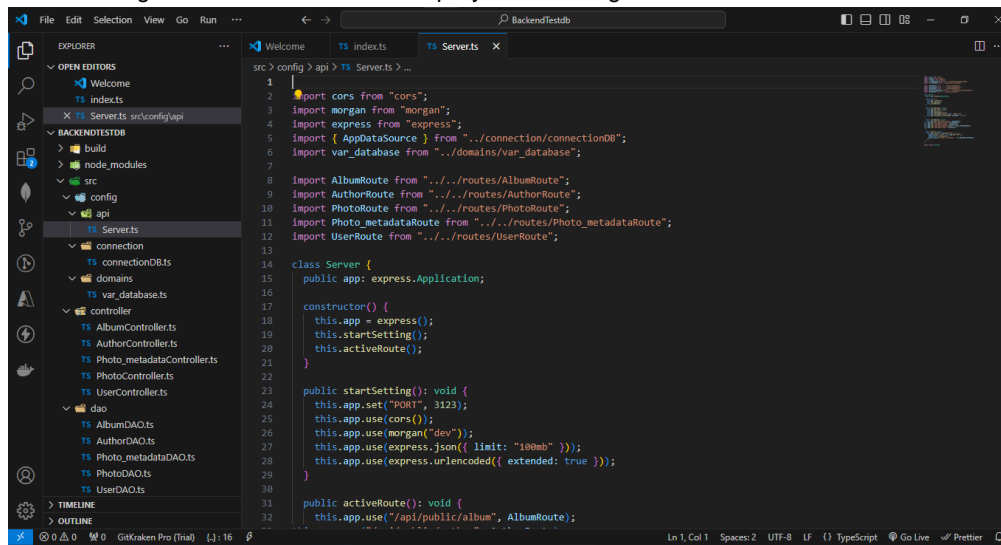
Fuente: Autores

Figura 202. Visualización del proyecto descomprimido



Fuente: Autores

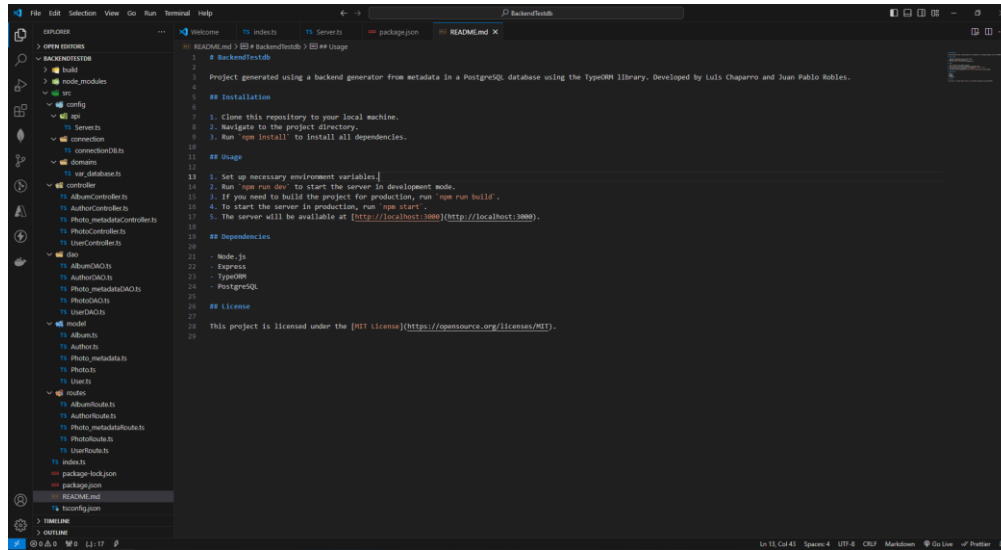
Figura 203. Visualización del proyecto descargado en Visual Studio Code



Fuente: Autores

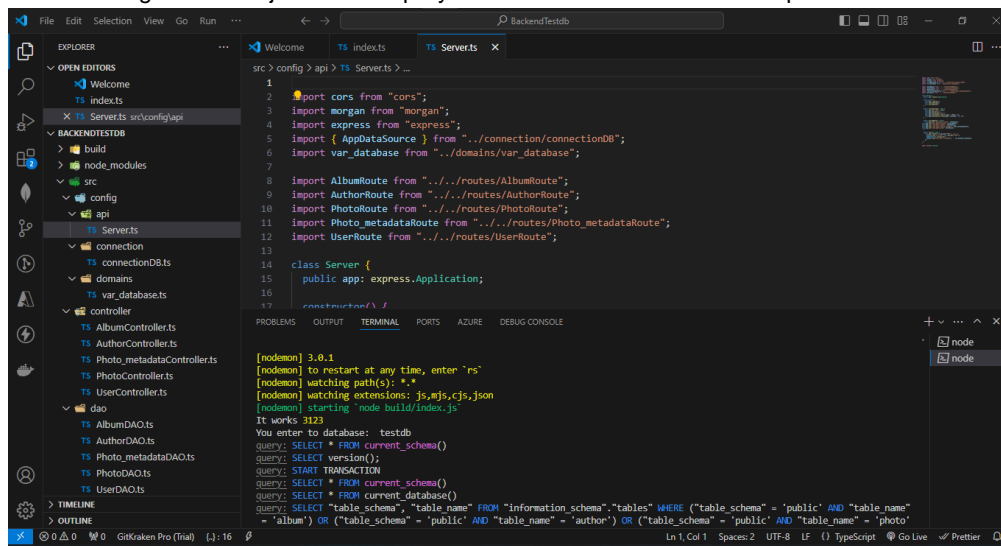
Siga las instrucciones proporcionadas en el archivo README que se muestra en la Figura 204 incluido para configurar y ejecutar el proyecto en el servidor de producción que se muestra en la Figura 205.

Figura 204. Archivo README.md



Fuente: Autores

Figura 205. Ejecución del proyecto Backend en el servidor de producción

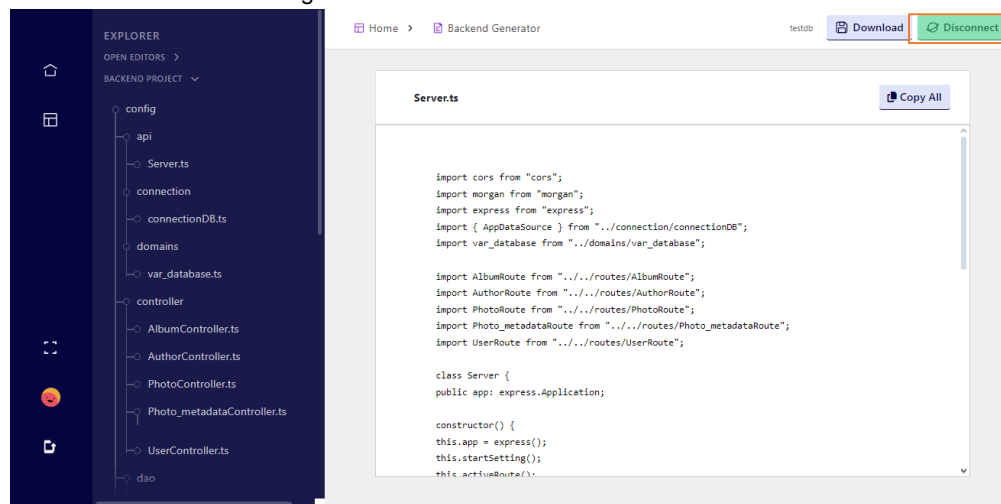


Fuente: Autores

### ✓ Desconexión de la base de datos:

En la Figura 206 se muestra la opción para desconectar la base de datos. Una vez desconectada, el usuario puede seguir probando conexiones con otras bases de datos relacionales de PostgreSQL, extrayendo los backends necesarios.

Figura 206. Desconexión de base de datos

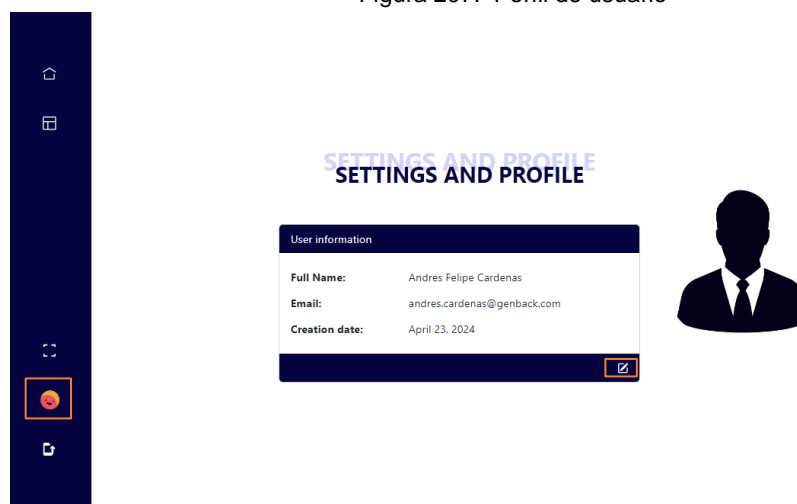


Fuente: Autores

### ✓ Configuración y perfil:

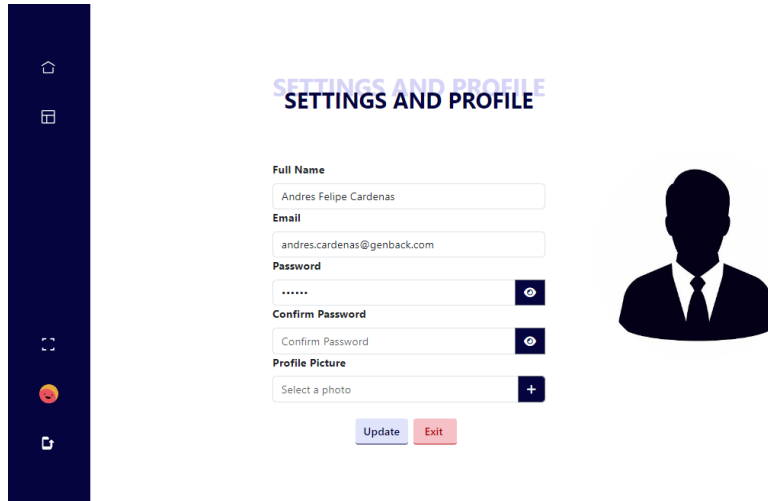
En la Figura 207 se presenta la sección de perfil, donde se encuentra la información del usuario y la posibilidad de actualizarla como se muestra en la Figura 208, donde se muestra los detalles del usuario.

Figura 207. Perfil de usuario



Fuente: Autores

Figura 208. Actualizar usuario

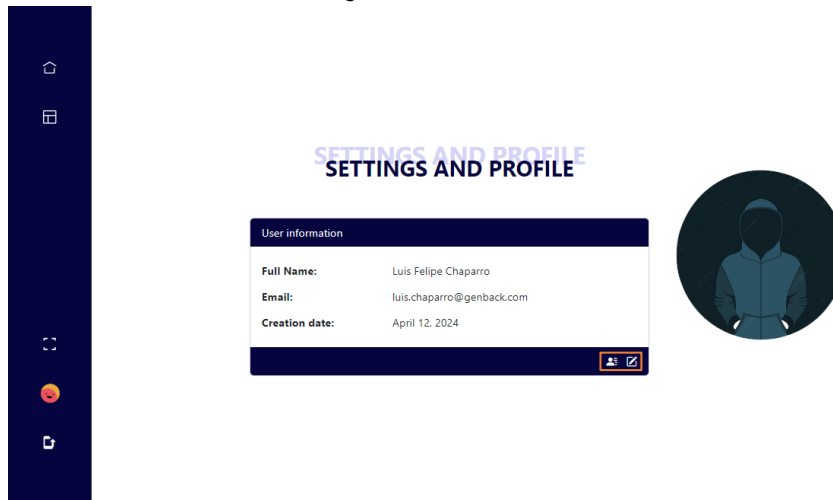


The screenshot shows a user interface for updating a user profile. On the left is a dark blue vertical sidebar with navigation icons: a home icon, a grid icon, a double-dash icon, a red smiley face icon, and a document icon. The main content area is titled "SETTINGS AND PROFILE" in blue and black text. Below the title are several form fields: "Full Name" with the value "Andres Felipe Cardenas", "Email" with "andres.cardenas@genback.com", "Password" with masked characters ".....", and "Confirm Password" with "Confirm Password". There are eye icons to toggle password visibility. Below these is a "Profile Picture" section with a "Select a photo" button and a plus icon. At the bottom are "Update" and "Exit" buttons. To the right of the form is a silhouette of a person in a suit.

Fuente: Autores

Los administradores, como se ilustra en la Figura 209, pueden acceder a la sección de listado de usuarios, mostrada en la Figura 210. Aquí tienen la capacidad de crear tanto usuarios como administradores, como se muestra en la Figura 211, y actualizarlos según sea necesario, tal como se ejemplifica en la Figura 212.

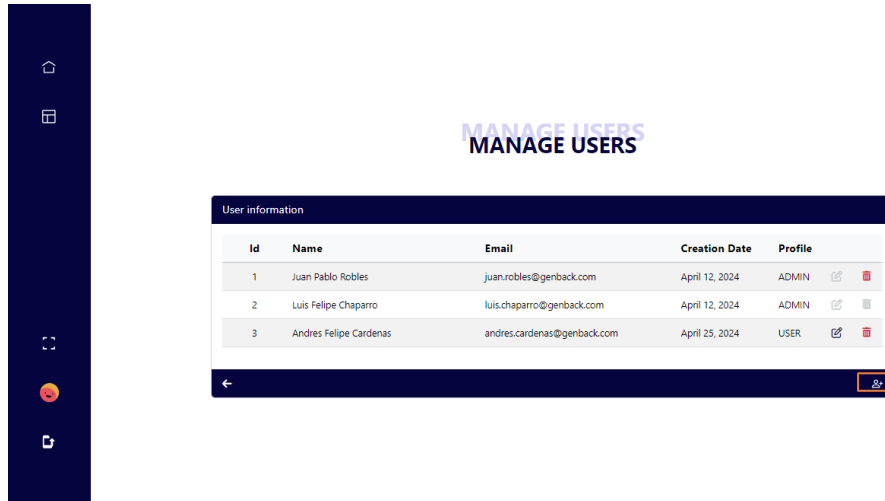
Figura 209. Perfil de administrador



The screenshot shows an administrator profile page. It features the same dark blue sidebar as Figure 208. The main content area is titled "SETTINGS AND PROFILE" in blue and black text. Below the title is a "User information" section with a dark blue header. The information displayed is: "Full Name: Luis Felipe Chaparro", "Email: luis.chaparro@genback.com", and "Creation date: April 12, 2024". At the bottom right of this section are icons for a person and a checkmark. To the right of the information box is a circular profile picture of a person in a blue hoodie.

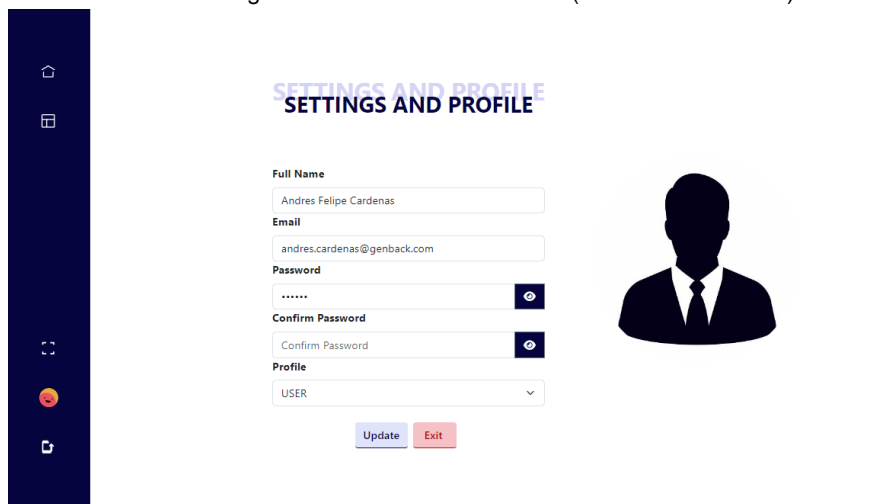
Fuente: Autores

Figura 210. Lista de usuarios



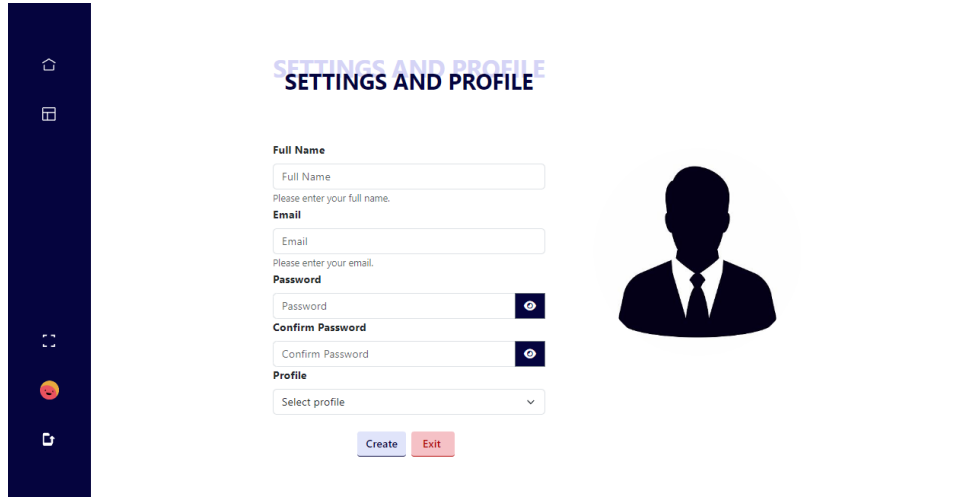
Fuente: Autores

Figura 211. Actualizar usuarios (Vista Administrador)



Fuente: Autores

Figura 212. Crear usuarios Actualizar usuarios (Vista Administrador)



**SETTINGS AND PROFILE**

**Full Name**  
Full Name  
Please enter your full name.

**Email**  
Email  
Please enter your email.

**Password**  
Password

**Confirm Password**  
Confirm Password

**Profile**  
Select profile

Create Exit

Fuente: Autores

✓ **Cerrar sesión:**

Figura 213. Cerrar sesión



Fuente: Autores

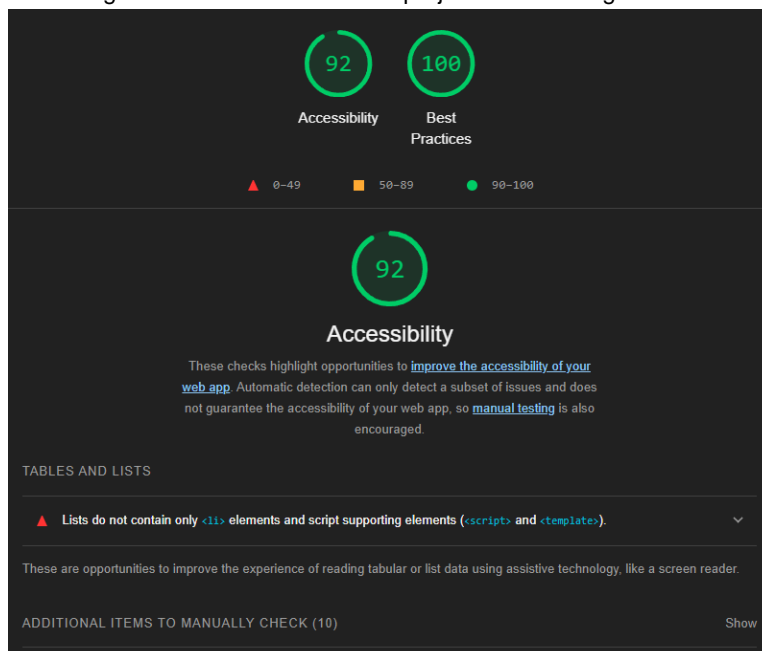
## 10.21. Anexo: Mantenimiento y correcciones de la meta herramienta

En el presente anexo muestra las evidencias de mantenimiento y las correcciones de la meta herramienta, realizadas de manera colaborativa a través de repositorios virtuales en GitHub. Este enfoque permitió un trabajo en equipo eficiente, lo que facilitó los ajustes y avances en el desarrollo tanto del backend como del frontend.

Después de completar las pruebas de frontend utilizando Lighthouse, se identificaron problemas con etiquetas <a>, botones, encabezados y otras etiquetas HTML que no cumplían con las pautas de accesibilidad.

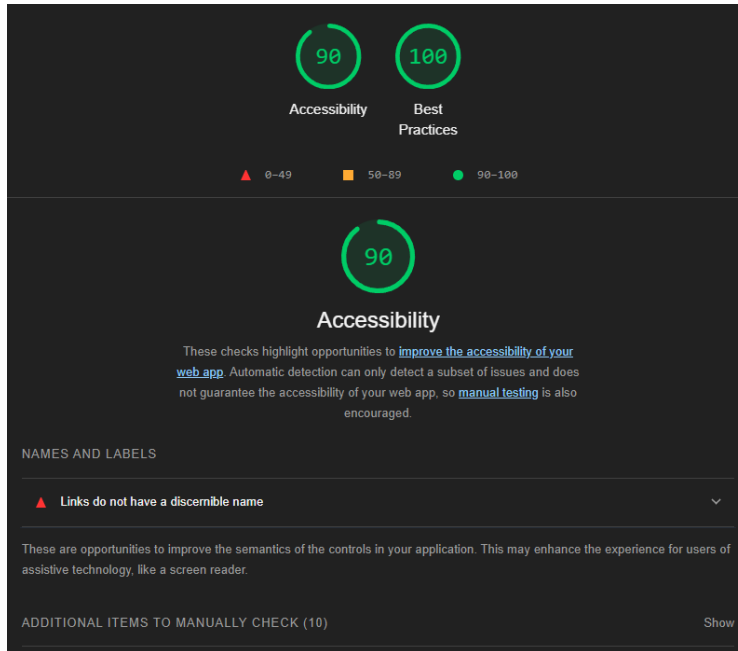
Para abordar estos problemas, implementamos correcciones, asegurándonos de que nuestras soluciones cumplieran con las normativas de accesibilidad. Las cuales se muestran de la Figura 214 a la Figura 219. Después de aplicar las correcciones, volvimos a ejecutar las pruebas de Lighthouse para evaluar el impacto de los cambios.

Figura 214. Prueba inicial a “/projectMenu” en Lighthouse



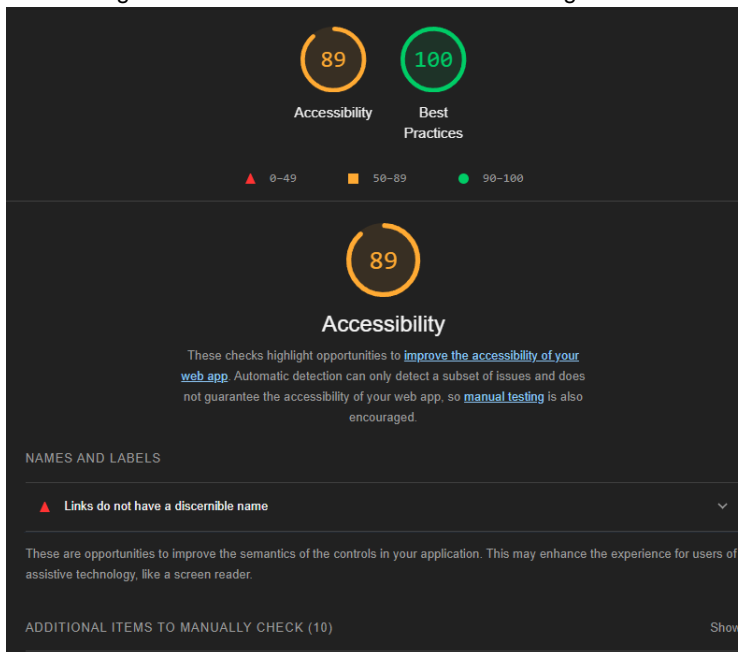
Fuente: Lighthouse

Figura 215. Prueba inicial a “/profile” en Lighthouse



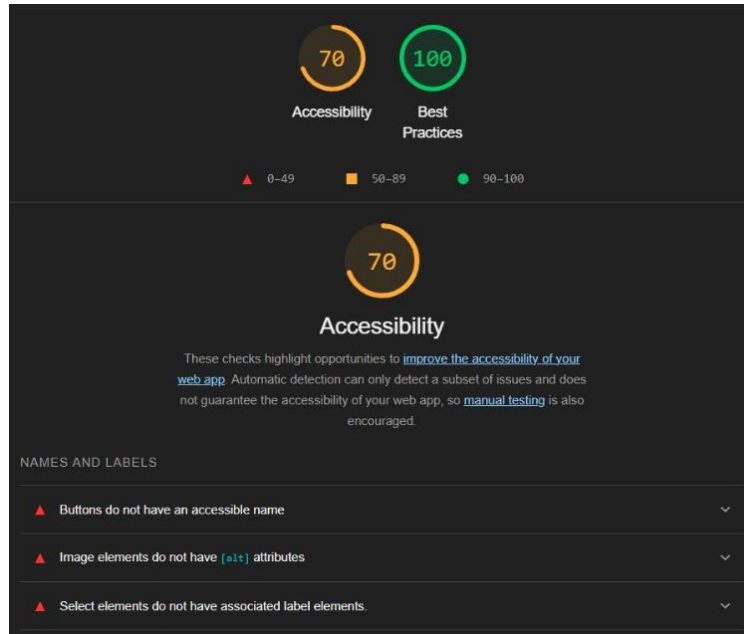
Fuente: Lighthouse

Figura 216. Prueba inicial a “/listUser” en Lighthouse



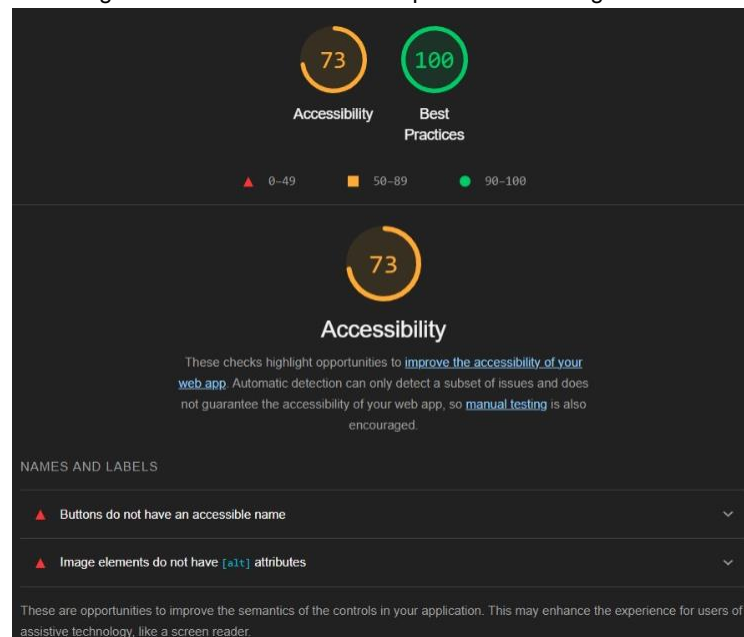
Fuente: Lighthouse

Figura 217. Prueba inicial a “/createUser” en Lighthouse



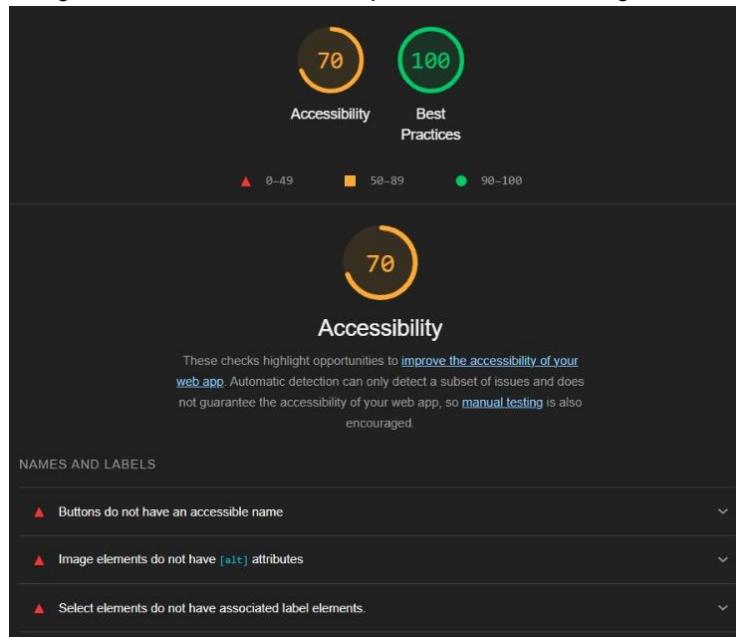
Fuente: Lighthouse

Figura 218. Prueba inicial a “/updateUser” en Lighthouse



Fuente: Lighthouse

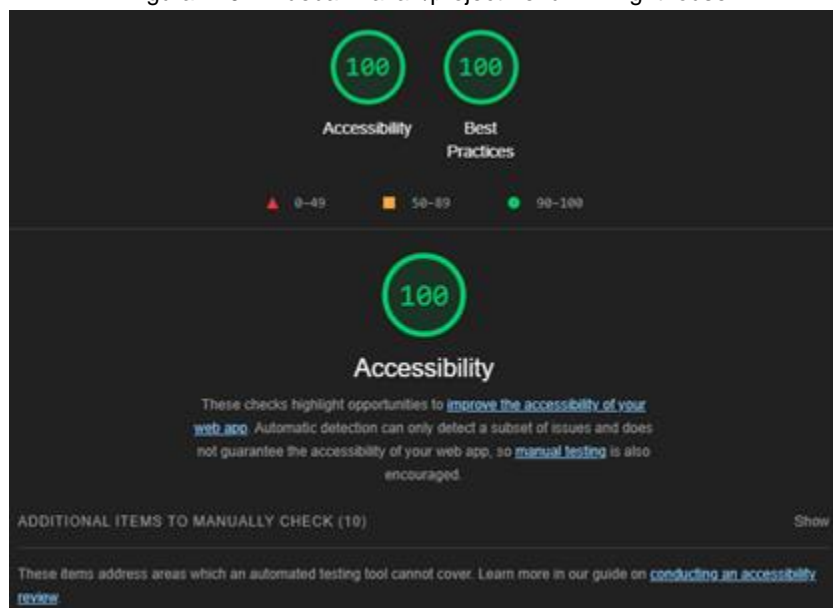
Figura 219. Prueba inicial a “/updateUserAdmin” en Lighthouse



Fuente: Lighthouse

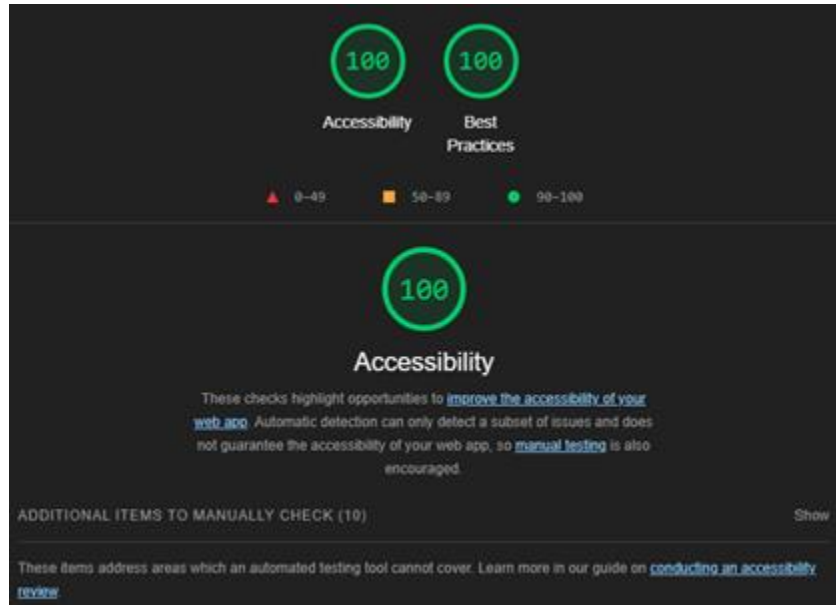
Gracias a estas mejoras, la accesibilidad del sitio web ha mejorado significativamente como se muestra en la Figura 220 a la Figura 225, asegurando que todos los usuarios puedan acceder y utilizar los servicios de manera efectiva.

Figura 220. Prueba final a “/projectMenu” en Lighthouse



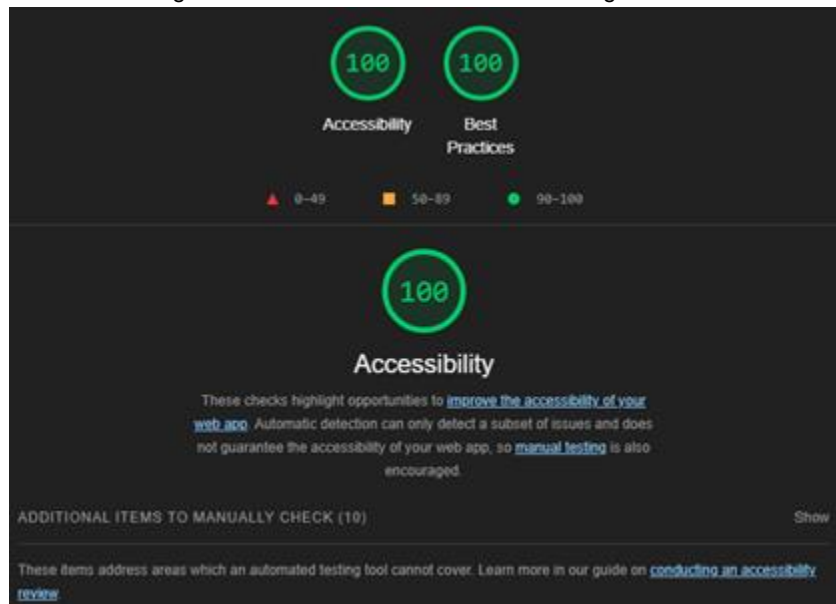
Fuente: Lighthouse

Figura 221. Prueba final a "/profile" en Lighthouse



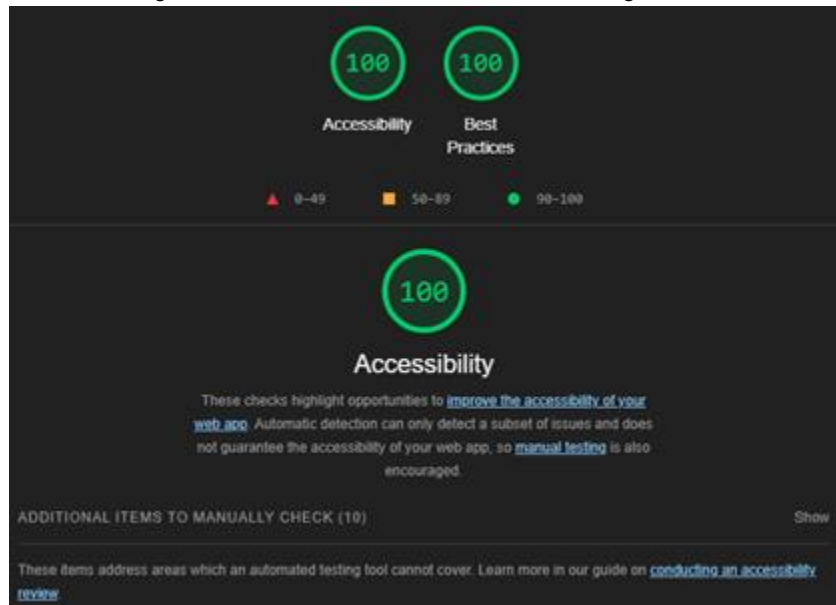
Fuente: Lighthouse

Figura 222. Prueba final a "/listUser" en Lighthouse



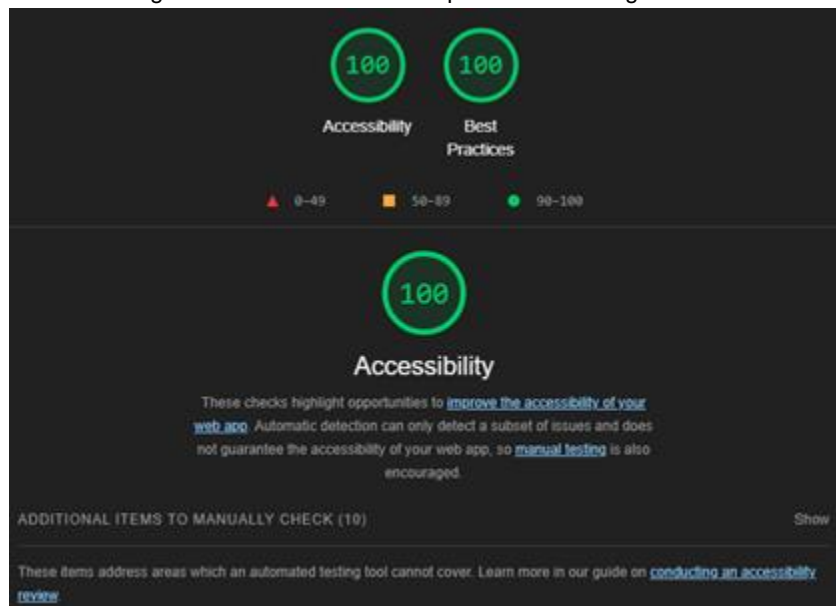
Fuente: Lighthouse

Figura 223. Prueba final a “/createUser” en Lighthouse



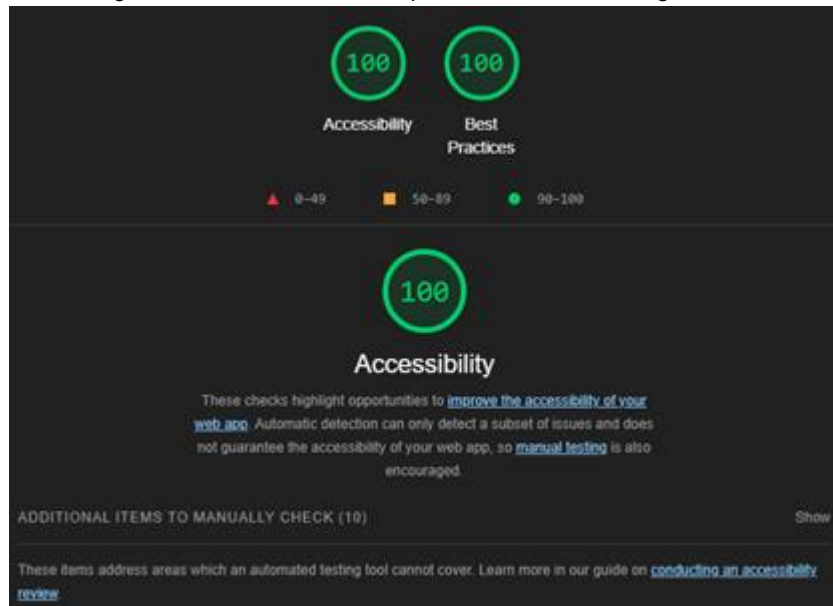
Fuente: Lighthouse

Figura 224. Prueba final a “/updateUser” en Lighthouse



Fuente: Lighthouse

Figura 225. Prueba final a “/updateUserAdmin” en Lighthouse



Fuente: Lighthouse

En la Figura 226 y Figura 227 muestra el proceso de mantenimiento, donde se identificaron y solucionaron problemas relacionados con la funcionalidad, accesibilidad y la interfaz gráfica del sistema.

Figura 226. Árbol de commits Frontend

Graph	Description	Date	Author	Commit
male   origin   origin/HEAD	Mantenimiento pruebas de accesibilidad	25 Apr 2024 18:10	Felipe	7561c7a
	Mantenimiento	23 Apr 2024 19:11	Felipe	b3d8ebd
	Correcciones terminadas	19 Apr 2024 15:12	Felipe	91880932
	Conflicto en menu, solucionado	19 Apr 2024 11:40	Felipe	4380c2d7
	Cambios manual de usuario	19 Apr 2024 11:38	Felipe	08a2c201
origin/development/juan	Update files manual user	19 Apr 2024 11:11	JuanPablo	4b006748
	Update admin settings	19 Apr 2024 10:39	JuanPablo	cb3396d1
	Change correct of main	14 Apr 2024 12:39	JuanPablo	e96419d7
	Update dashboard admin	14 Apr 2024 10:22	JuanPablo	25e1e421
	Subiendo codigo documentado	13 Apr 2024 20:08	Felipe	82d55998
LasChaparro   origin	Ultimos detalles	12 Apr 2024 21:21	Felipe	1c74287
	Cambio administrador usuarios	12 Apr 2024 13:53	Felipe	317c1647
	Traduciendo	12 Apr 2024 13:47	Felipe	f1618b30
	Security views admin and user	10 Apr 2024 10:04	JuanPablo	2d059515
	Admin functions frontend	10 Apr 2024 00:29	JuanPablo	e889856a
	Boton copiar	8 Apr 2024 21:32	Felipe	f99ea375
	Cambios configuracion y perfil	19 Mar 2024 23:47	Felipe	a3a6ffc1
	Cambios en el arbol	12 Mar 2024 15:54	Felipe	f292edbb
	Change whit app and security routes private	9 Mar 2024 16:57	JuanPablo	1109b202
	Update changes Luis and func toast	26 Feb 2024 11:44	JuanPablo	2b4eff69
	luis	26 Feb 2024 11:11	JuanPablo	90591fa9
	Trayecto cambios	26 Feb 2024 11:05	JuanPablo	67420a79
	Entrega Luis	25 Feb 2024 22:11	Felipe	8f3bec39
	Services backend	15 Nov 2023 17:36	JuanPablo	ba5e7420
	Test ModelGenerator with new component and id table	14 Nov 2023 11:30	JuanPablo	4c33489f
	Test of Backend the generate Models whit connection DB	1 Nov 2023 19:44	JuanPablo	8d9610dc
	Trayecto formulario	26 Oct 2023 18:28	Felipe	91497f05
	Cambios	26 Oct 2023 18:27	Felipe	33ae55d7
	Test connection db	26 Oct 2023 09:57	JuanPablo	1d31f6d7
	Test commit project	24 Oct 2023 19:22	JuanPablo	d484f64
	initial commit	24 Oct 2023 12:16	JuanPablo	3064c5dc

Fuente: Autores

Figura 227. Árbol de commits Backend

Graph	Description	Date	Author	Commit
origin/HEAD	Mantenimiento subiendo readme	25 Apr 2024 18:59	Felipe	4581da79
origin/development_juan	Unit test building project and download, create endpoint readme	25 Apr 2024 12:03	JuanPablo	2478e96a
origin/main	Merge branch 'main' into development_juan	25 Apr 2024 11:46	JuanPablo	7121189c
origin/main	Test build project and download	25 Apr 2024 11:44	JuanPablo	8daa3280
origin/main	Correcciones terminadas	19 Apr 2024 15:13	Felipe	e4013f5d
origin/main	Merge branch 'development_juan'	16 Apr 2024 10:32	JuanPablo	ca9602f
origin/main	Unit test backend whit test	19 Apr 2024 10:31	JuanPablo	0920193
origin/main	Codigo documentado	15 Apr 2024 16:53	Felipe	09d8bc93
origin/main	Update service generecode with tables Uppercase	14 Apr 2024 12:41	JuanPablo	99008c3
origin/main	Cambios LuisChaparro, descarga	12 Apr 2024 21:26	Felipe	95389c0a
origin/main	Merge branch 'main' of https://github.com/juanrobles21/BackendGenBack	12 Apr 2024 13:46	Felipe	c48c5386
origin/main	Cambios orden de codigo generado	12 Apr 2024 13:45	Felipe	7501020
origin/main	Services package json	12 Apr 2024 13:41	JuanPablo	b3311104
origin/main	Functon backend admin project	10 Apr 2024 00:32	JuanPablo	4946b4bb
origin/main	Cambios visualizacion	8 Apr 2024 21:31	Felipe	5c004a9
origin/main	Update with crypt pw	26 Mar 2024 11:39	JuanPablo	ed2be9ad
origin/main	Cambios en token	19 Mar 2024 23:46	Felipe	a99cd8e3
origin/main	Merge branch 'main' of https://github.com/juanrobles21/BackendGenBack	16 Mar 2024 14:37	Felipe	c7803379
origin/main	Functon edit users and update data jwt	15 Mar 2024 18:45	JuanPablo	8e8d4585
origin/main	Normal	12 Mar 2024 15:54	Felipe	6e96e831
origin/main	Cambios	12 Mar 2024 12:28	Felipe	5ec90754
origin/main	Trayendo correcciones	12 Mar 2024 12:24	Felipe	cfaeeeb3
origin/main	Update structure genereate code typeOrm	12 Mar 2024 11:48	JuanPablo	a419ee16
origin/main	Merge branch 'main' of https://github.com/juanrobles21/BackendGenBack	11 Mar 2024 13:12	Felipe	c2a3448a
origin/main	Error	11 Mar 2024 13:12	Felipe	40e9c573
origin/main	Create seguity api and middleware whit jwt	9 Mar 2024 17:04	JuanPablo	035ca6b3
origin/main	Merge branch 'main' of https://github.com/juanrobles21/BackendGenBack	15 Nov 2023 19:10	Felipe	1800f6f1
origin/main	Cambios espacios	15 Nov 2023 19:10	Felipe	4a220547
origin/main	Generator backend project	15 Nov 2023 17:30	JuanPablo	db3b79f4
origin/main	Cambios caso finales Juan	10 Nov 2023 14:12	Felipe	ba28ce31
origin/main	LuisChaparro Back a Luis	3 Nov 2023 02:13	Felipe	77b5da49
origin/main	Functionality Generate ModelsTypeOrm	1 Nov 2023 19:35	JuanPablo	cd703c3
origin/main	Test connection DB	26 Oct 2023 09:54	JuanPablo	b62c0223
origin/LuisChaparro	First commit project	24 Oct 2023 19:23	JuanPablo	c7261d0e

Fuente: Autores