



SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN Y GENERACIÓN DE  
REPORTES DE LOS SISTEMAS DE GESTIÓN DOCUMENTAL

Danny Jersson Ochoa  
Peralta

Tutora  
Isabel Andrea Mahecha Nieto

Institución Universitaria Politécnico  
Grancolombiano  
Facultad de ingeniería, diseño e innovación  
Ingeniería de Sistemas  
Bogotá D.C.  
2021

## Tabla de contenido

1. RESUMEN DEL PROYECTO .....	6
2. PLANTEAMIENTO DEL PROBLEMA .....	8
3. JUSTIFICACIÓN .....	10
4. ALCANCE .....	11
4.1. Restricciones .....	11
5. OBJETIVOS .....	12
5.1. Objetivo general .....	12
5.2. Objetivos específicos.....	12
6. CARGO Y FUNCIONES .....	13
6.1. Cargo .....	13
6.2. Funciones: .....	13
7. PERFIL DE LA EMPRESA .....	14
7.1. Razón Social de la empresa.....	14
7.2. Objeto social de la empresa.....	14
7.2.1. Misión de la empresa .....	14
7.2.2. Visión de la empresa.....	14
7.2.3. Objetivos de la empresa .....	15
8. MARCO TEÓRICO .....	16
8.1. Documento .....	16
8.2. Gestión documental.....	17
8.3. Reporte .....	18
8.4. Tabla de retención documental .....	18
8.5. Arquitectura de software .....	19
8.6. Modelo Vista Controlador.....	19
8.7. Front-end: .....	19
8.8. Back-end.....	21
8.9. JavaScript .....	21
8.10. Node.js .....	22
8.11. Control de acceso HTTP (CORS) .....	22
8.12. JSON.....	24
8.13. Base de datos .....	24
8.14. Bases de datos relacionales.....	24

8.15.	PostgreSQL.....	25
8.16.	Scrum.....	25
9.	ESTADO DEL ARTE .....	27
10.	METODOLOGÍA .....	29
10.1.	¿Qué es metodología agile?.....	29
10.2.	Analizar la manera en los que se generar reportes.....	29
10.3.	Metodología ágil elegida .....	29
10.4.	Levantamiento de requerimientos .....	30
11.	IMPLEMENTACIÓN.....	31
11.1.	Tecnologías asignadas por el Scrum master para el Desarrollo .....	31
11.1.1.	Reuniones .....	31
11.1.2.	Diseño.....	32
11.1.3.	Resultados de la metodología Scrum.....	33
12.	ARQUITECTURA DE SOFTWARE.....	38
12.1.	Diseño de la arquitectura .....	38
13.	REQUERIMIENTOS .....	41
13.1.	Requerimientos funcionales .....	41
13.2.	Requerimientos no funcionales .....	42
14.	DIAGRAMAS UML .....	43
14.1.	Diagrama de clases .....	43
14.2.	Diagrama de componentes.....	44
14.3.	Diagrama de despliegue.....	45
14.4.	Diagrama de secuencias.....	46
14.5.	Diagrama casos de uso .....	52
15.	MODELO BASES DE DATOS .....	53
15.1.	Diagrama entidad-relación .....	53
16.	RESULTADOS.....	54
16.1.	Back-end.....	54
16.2.	Front-end .....	59
17.	CRONOGRAMA.....	68
18.	CONCLUSIONES .....	69
19.	BIBLIOGRAFÍA .....	71

## Tabla de ilustraciones y tablas

Ilustración 1 ciclo de vida gestión documental, adaptado Check out this amazing template to make your presentations look awesome (s. f. Slideteam) 2020.....	17
Ilustración 2 Gráfico API, elaboración propia, adaptado A. (2020b, diciembre 3).....	23
Ilustración 3 Diagrama de actividades de Scrum A. (2021a, septiembre 8).....	26
Ilustración 4 Tabla de sprints.....	33
Ilustración 5 Crear sprint .....	34
Ilustración 6 Progreso de un sprint .....	35
Ilustración 7 Repositorio en BitBucket.....	35
Ilustración 8 Commits en repositorio.....	36
Ilustración 9 Desarrollo de una épica.....	36
Ilustración 10 Backlog .....	37
Ilustración 11 Arquitectura web service .....	39
Ilustración 12 Patrón MVC.....	40
Ilustración 13 Diagrama de clases .....	43
Ilustración 14 Diagrama de componentes.....	44
Ilustración 15 Diagrama de despliegue.....	45
Ilustración 16 Diagrama de secuencia consultar usuario.....	46
Ilustración 17 Diagrama de secuencia editar usuario .....	47
Ilustración 18 Diagrama de secuencia consultar reportes.....	48
Ilustración 19 Diagrama de secuencia login .....	49
Ilustración 20 Diagrama de secuencia eliminar usuario .....	50
Ilustración 21 Diagrama de secuencia registro usuario .....	51
Ilustración 22 Diagrama de casos de uso.....	52
Ilustración 23 Diagrama entidad-relación.....	53
Ilustración 24 Diagrama generación de reportes .....	53
Ilustración 25 Dependencias backend.....	54
Ilustración 26 POST login .....	55
Ilustración 27 Respuesta POST login .....	55
Ilustración 28 GET usuarios .....	56
Ilustración 29 Respuesta GET usuarios .....	56
Ilustración 30 PUT usuario .....	57
Ilustración 31 Respuesta PUT usuario elaboración propia.....	57
Ilustración 32 DELETE usuario.....	57
Ilustración 33 Respuesta DELETE usuario .....	58
Ilustración 34 Vista login.....	59
Ilustración 35 Vista login fallido .....	60
Ilustración 36 Vista admin.....	61
Ilustración 37 Vista ver usuarios.....	62
Ilustración 38 Vista agregar usuarios.....	63
Ilustración 39 Editar usuario.....	64

Ilustración 40 Vista dashboard.....	65
Ilustración 41 Vista reportes .....	66
Ilustración 42 Vista usuario .....	67
Ilustración 43 Diagrama de Gantt.....	68

## 1. RESUMEN DEL PROYECTO

Este proyecto se encuentra encaminado en desarrollar un sistema de información que cuenta con una interfaz amigable y sencilla para el usuario, enfocado en la generación de reportes basados en gestión documental que se encuentran almacenados en distintas bases de datos de la empresa. El sistema de información genera los reportes en archivos en formato de hoja cálculo con la información correspondiente, donde se suplió la necesidad de realizar los reportes manualmente, con esto disminuye considerablemente el tiempo en el que se generan.

Las personas que son usuarios tienen acceso a los reportes correspondientes y si tienen el rol de administrador gestiona los usuarios que tienen acceso al sistema. Para el desarrollo de esta aplicación se usó la metodología SCRUM para poder implementar de manera adecuada el proyecto.

Se cumplió la necesidad de generar reportes sobre la gestión documental, por lo cual se desarrolló el sistema con sus respectivas pruebas para detallar el correcto funcionamiento en la etapa de producción del sistema y a la hora de hacer el despliegue, se enfocó en los objetivos propuestos, para satisfacer estas necesidades se hizo una previa investigación de los temas referentes al proyecto como al desarrollo web, gestión documental y sistema de información.

Para desarrollar este proyecto fue necesario el uso de material referente a la gestión documental, por esta razón se utilizaron libros como “Gestión Documental” de PR Gallo, ya que desarrolla la idea y la necesidad de la gestión documental. Este libro contiene elementos referentes a la gestión documental, por tanto, se usó como una excelente guía para entender el funcionamiento y los requerimientos necesarios para desarrollar la

aplicación. Tanto los componentes del backend, del frontend y de la base de datos trabajen de manera colaborativa con su respectiva documentación del software.

**Palabras Claves:** Gestión documental, generación de reportes, tablas, bases de datos, información, Sistema de información, interfaz, backend, frontend, API, documentos, gestión de usuarios, Scrum, metodología ágil.

## 2. PLANTEAMIENTO DEL PROBLEMA

Manejo Técnico de Información SA (MTI), es una empresa reconocida por sus sistema de administración de información, sus fortalezas es permitir acceso rápido y eficiente. Sabiendo que la gestión documental es distinta para cada organización, el servicio que ofrece MTI es personalizado para cada una. Las principales funciones de la empresa es diseñar e implementar la gestión documental de cada organización. El uso de herramientas informáticas permite la ayuda para la gestión documental, MTI permite un buen uso de esas herramientas.

Como definió Mijailov, Chernii y Guiliarevskii por documento es entendido como “cualquier objeto material que registra o fija algún conocimiento y puede ser incluido en una recopilación. En su acepción más amplia, conviene no solo a las inscripciones, manuscritos y ediciones impresas sino también a las obras de arte, monedas y piezas de museo [1, p. 79]”.

Para la empresa MTI, la importancia de la gestión documental crece con las mismas organizaciones, al administrar esta información se busca tener un acceso rápido y eficiente, basándose principalmente en la eficiencia y la protección de la información, al manejar todas estas bases de datos de gestión documental.

Los documentos que se manejan en la empresa son de carácter confidencial, ya que las organizaciones tienen temas importantes que deben ser reservados, solo las personas designadas pueden tener acceso a estos documentos. Por esta razón se debe tener como eje principal la seguridad, por este motivo en la solución se priorizó esto.



La generación de reportes se usa en gestiones tecnológicas aprovechando todas las herramientas del área del desarrollo con lo cual se construyó un producto que supla las necesidades y originando la documentación requerida. Los reportes son estructurados con la información requerida a la hora de hacer las consultas a los sistemas de información.

### 3. JUSTIFICACIÓN

Al ofrecer una solución a la problemática presentada por la empresa, el desarrollo de una aplicación web, que les permite acceder de una manera óptima teniendo control de los reportes y la administración de los documentos que allí se requieran. Esto genera un gran impacto en el nivel de productividad, ya que repercute en el tiempo que se requiere al generar los reportes de manera manual. Lo otro que se puede gestionar son los documentos físicos, ya que este tipo de documentos se almacenan una gran cantidad, con la aplicación estos se pueden ordenar de manera electrónica, obviamente respetando su confidencialidad.

La importancia de esta herramienta es bastante grande porque gestionar los documentos son cosas muy relevantes, entonces por cuestiones de facilidad esta herramienta es la más indicada, ya que contara con todas las funcionalidades necesarias para un buen desempeño y compatibilidad. Este proyecto se debe llevar a cabo por el planteamiento del problema.

Al enfocarse en la parte de seguridad de la aplicación, de esta forma los datos que la empresa maneja van a mantener de una manera segura y podrán acceder las personas que estén autorizadas, por esta razón se manejara una autenticación y solo podrán acceder los que tengan ciertas credenciales, los que van a ser los administradores podrán gestionar los usuarios que tengan acceso a los informes y documentación.

#### 4. ALCANCE

El sistema tiene el alcance del departamento de tecnología, con el software se puede optimizar de una manera más exacta el uso de reportes y el manejo de estos, como la gestión de la información que se está trabajando.

Todos los documentos están asociados a los datos de otras empresas, con la aplicación estos se pueden acceder e identificarlos de una manera más fácil, los encargados del manejo de reportes para que puedan acceder a estos, ya desde el aplicativo se podrán gestionar con una mayor facilidad.

Se pensó en eventos futuros desde la plataforma tecnológica se puede acceder mediante la conectividad al servidor vía Internet, ya desde la aplicación se tendrá acceso a la gestión de reportes.

##### 4.1. Restricciones

- Por políticas de privacidad y confidencialidad, mucho de los datos manejados en los sistemas de información no pueden ser compartidos por ningún motivo.
- El software solo está disponible para el personal que tenga autorización por parte de la compañía.

## 5. OBJETIVOS

### 5.1.Objetivo general

Desarrollar un sistema de información enfocado a la generación de reportes basado en gestión documental, abarcando la información suministrada en las bases de datos de la empresa de sus distintos clientes.

### 5.2.Objetivos específicos

- Desarrollar una base de datos relacional, que integre otras bases de datos, relacionales y no relacionales.
- Desarrollar un backend, que conecta con distintos servicios a la base de datos, para interactuar con la información allí almacenada.
- Implementar la conexión entre el frontend y el backend, para acceder a las características de la aplicación de manera correcta.

## 6. CARGO Y FUNCIONES

### 6.1.Cargo

Practicante en el área de desarrollo de software

### 6.2.Funciones:

- Brindar soporte y mantenimiento a las aplicaciones de la compañía con soluciones enfocadas a la gestión documental.
- Apoyar en la elaboración de la documentación de los sistemas de información de la empresa, con datos obtenidos de los distintos medios que maneja la empresa.
- Realizar procesos de calidad en pruebas unitarias de los sistemas de información, con esto se comprueba las funcionalidades correctas de cada componente de las aplicaciones.

## 7. PERFIL DE LA EMPRESA

### 7.1. Razón Social de la empresa

- Manejo técnico De La Información S.A
- NIT: 900011545-4
- Régimen Común

### 7.2. Objeto social de la empresa

“El objetivo social principal de la empresa manejo técnico de información es la custodia, el almacenar y movilizar archivos de distinto indoles como lo son archivos magnéticos, archivos físicos, etc.” (Información Empresarial Manejo Técnico de Información, 2021)

#### 7.2.1. Misión de la empresa

“La principal misión de es dar soluciones de seguridad integral y eficientes basadas en la innovación y tecnología”. (Información Empresarial Manejo Técnico de Información, 2021)

#### 7.2.2. Visión de la empresa

“La visión de Manejo Técnico de Información es ser líderes en soluciones integrales de seguridad que harán de las vidas de las personas más seguras y tranquilas”. (Información Empresarial Manejo Técnico de Información, 2021)

### 7.2.3. Objetivos de la empresa

7.2.3.1. “Implementar modelos de gestión que sean adecuados a las empresas, mediante la creación de TRD Y TVD” implementando procesos en aplicaciones propias”.

(Información Empresarial Manejo Técnico de Información, 2021)

7.2.3.2. “Prestar el servicio de gestión documental con personas que sean especialistas en el manejo de las aplicaciones que sirven para administrar información, servicio de mensajería y correspondencia.”

## 8. MARCO TEÓRICO

El marco teórico que se va a describir a continuación hace referencia a la conceptualización que se va a encontrar en el presente documento acerca del desarrollo del proyecto. Se va a definir la Gestión Documental, reportes, información que contendrá el sistema, flujos de información e información sobre todo el desarrollo.

### 8.1. Documento

La palabra “documento” lo podemos denotar como un registro textual de información (Department of Defence, 2015). Un “documento” debe incluir lo siguiente:

- a. Cualquiera de los siguientes elementos:
  - Cualquier papel u otro material donde pueda contener escritura.
  - Mapas, planos, dibujos o fotografías.
  - Cualquier escrito en el cual se haya almacenado o registrado información, ya sea de forma física o electrónica.
  - Cualquier otro registro de información.
- b. Cualquier copia, reproducción o duplicado de las cosas antes nombradas.
- c. Cualquier parte de dicha copia, reproducción o duplicado.

Según el artículo 251 del Código de Procedimiento Civil “Son documentos los escritos, impresos, planos, dibujos, cuadros fotografías, cintas cinematográficas, discos, grabaciones magnetofónicas, radiografías, talones, contraseñas, cupones, etiquetas, sellos



y, en general, todo objeto mueble que tenga carácter representativo o declarativo, y las inscripciones en lápidas, monumentos, edificios o similares (Código de Procedimiento Civil).

El Archivo General de la Nación define el documento como: “Información registrada, cualquiera sea su forma o el medio utilizado.”

## 8.2. Gestión documental

“La gestión documental es un conjunto de actividades las cuales permiten controlar y coordinar los puntos relacionados con la generación, recepción, organización, almacenamiento, perseveración, acceso y difusión de documentos”. (Patricia, 2021)

De otra forma la ISO define la gestión documental como la disciplina encargada del control eficiente y sistemático de la creación, recepción, mantenimiento, uso y eliminación de registros, se incluye el proceso de captura de la información. Por ejemplo, en la ilustración 1 se muestra el ciclo de vida de la gestión documental.



*Ilustración 1 ciclo de vida gestión documental, adaptado Check out this amazing template to make your presentations look awesome (s. f. Slideteam) 2020.*

### 8.3.Reporte

“Un reporte es una forma específica en el cual se organiza en torno a eventos o hallazgos que hayan ocurridos en eventos dentro de las organizaciones.” (Massey University, 2019)

“En las bases de datos se pueden generar reportes, estos resultan para presentar información de las bases de datos. Tiene los siguientes objetivos: Mostrar o distribuir un resumen de los datos”. (Microsoft, 2021)

Teniendo en cuenta esto, en las bases de datos se pueden almacenar bastante información, para esto se requiere la generación de reportes, de esta forma poder analizar la información es muy necesario.

### 8.4.Tabla de retención documental

“Esto se define como el listado de series, con los correspondientes tipos de documentos, en estos se le asignan el tiempo del ciclo de vida de estos documentos, con esto se puede buscar cuales documentos son de que entidad, su necesidad e importancia revisando el tiempo de conservación.” (Archivo General de la Nación, 2021)

Estas tablas contienen datos muy relevantes sobre clientes, a la hora de manipularla se debe tener el conocimiento suficiente para que la información allí almacenada sea administrada de manera correcta. Además, se debe tener muy en cuenta la seguridad de estos archivos debido a que contienen información sensible sobre clientes.

### 8.5.Arquitectura de software

La arquitectura que se definió para el proceso de las decisiones de las etapas a la hora del diseño del sistema.

“Estas decisiones tempranas llevan un peso importante con respecto al desarrollo del resto de un sistema, ya que condicionan otras decisiones que siguen, y en el caso de cambios, implican ramificaciones posteriores.” (M. E., 2017).

Para un buen desarrollo de software se tuvo en cuenta el diseño en el que tiene pensado el cliente, con la infraestructura de las TI pensando en los principales objetivos de la problemática.

### 8.6.Modelo Vista Controlador

El modelo vista controlador (MVC) es un concepto introducido por el fundador de Smalltalk (Trygve Reenskaug) para convertir un tipo de paquete de datos de red en otro tipo de datos junto con el procesamiento (modelo), desde el proceso de manipulación (controlador) y la vista (vista) para presentarse en una interfaz de usuarios. (Deacon, 2009)

### 8.7.Front-end:

“El desarrollo Font-end está pensado principalmente en los usuarios. Es un desarrollo que se centra en la codificación y creación de elementos de un sitio web en el cual el usuario interactuara”. (Goins, 2021)

Para el desarrollo del front-end, se pueden usar herramientas, lenguajes y framework para un desarrollo completo de esta parte del proyecto, para el desarrollo de este se usaron las siguientes:

- **AngularJS**

AngularJS es de código abierto, que permite la facilidad de creación de páginas web. Su principal meta es proporcionar todas las herramientas necesarias al desarrollo de una aplicación robusta y organizada, además por contar con el lenguaje de texto HTML, la hoja de estilos CSS y el lenguaje JavaScript se puede desarrollar todos los componentes necesarios. Las características principales de AngularJS son:

**Directivas:** Las directivas son para que el sistema HTML se comporte de una manera determinada.

**Enlace de datos:** Permite crear un enlace entre el modelo y la vista.

**Filtros:** Los filtros son herramientas de angular que permiten declarar la transformación de datos, para mostrárselos a los usuarios en las plantillas generados.

- **Angular Material**

“Angular Material es una librería de componentes web basado en el diseño de Material design. Desde un tiempo para acá se ha puesto bastante de moda este estilo de material design, y han salido muchas librerías que lo implementan.” (Potions, 2019)

Para usar las librerías de Angular Material se añade al proyecto de angular se agrega con el sistema de gestión de paquetes “npm”. Ya con esto se importan los objetos de material.

## 8.8.Back-end

“El desarrollo Back-end esta principalmente enfocado en el lado del sitio web que los usuarios no pueden ver. Esto hace que el sitio sea interactivo. Se sabe que los usuarios no interactúan directamente con el Back-end de un sitio web, pero si lo hacen de manera indirecta a través de los elementos del Front- end ya que este consume de la aplicación del Back-end”. (Goins, 2021)

Debido a que el back-end es de donde se consume los servicios, la mayor parte de la lógica a la que se accede a la información de las bases de datos se encuentra en esta parte que es del lado del servidor.

Tanto como el Front-end, el Back-end también tiene sus propias herramientas para el desarrollo de un sitio web, las cuales son las siguientes:

## 8.9.JavaScript

“JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. [...] Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos.” (Pérez, J. E., 2019)

Este lenguaje se puede usar tanto como en el back-end y front-end, para este caso se usó en Back debido a que se usa con el entorno Node.js, ya que fue escrito en JavaScript, por esta razón el manejo de este entorno la mejor opción es JavaScript.

## 8.10. Node.js

“Node.js es un entorno de ejecución orientado a eventos asíncronos de JavaScript diseñado para construir aplicaciones de red escalables.” (Node.js, 2018)

A la hora de crear un servidor se debe pensar de una manera distinta de cómo se está acostumbrado ya que su meta principal es permitirle al desarrollador desarrollar aplicaciones escalables, sin preocuparse de las conexiones simultáneas sin ningún límite. Usa una programación orientada a eventos orientadas a situaciones del lado del servidor y no del cliente.

Node.js utiliza la programación asíncrona, usando un archivo en el servidor y devolviendo el contenido al cliente, con la herramienta Express de este framework nos permite el intercambio de información por middlewares.

## 8.11. Control de acceso HTTP (CORS)

“El Intercambio de Recursos de Origen Cruzado (CORS (en-US)) es un mecanismo que utiliza cabeceras HTTP adicionales para permitir que un usuario obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio) al que pertenece.” (MDN, 2021)

Hay herramientas que permite comprobar la funcionalidad de estas APIS, para este proyecto se usó Postman.

Los métodos que se usan principalmente a la hora de hacer peticiones son las siguientes:

- GET: El método GET obtiene datos de la base de datos de sitios específicos, con este método se puede obtener todos los datos de una base de datos o simplemente un solo dato traído desde el ID.
- POST: El método POST se usar para enviar solicitudes con recursos que se desean llenar, este método inserta datos, algunos cambios se causan en el servidor, pero estos son mínimos.
- PUT: El método PUT reemplaza datos desde una petición de registros ya almacenados en una base de datos.
- DELETE: El método DELETE elimina un registro específico en la base de datos.

Para entender el funcionamiento del control de acceso HTTP entre la base de datos y la aplicación con los distintos métodos de acceso, esto se puede observar en la ilustración 2.

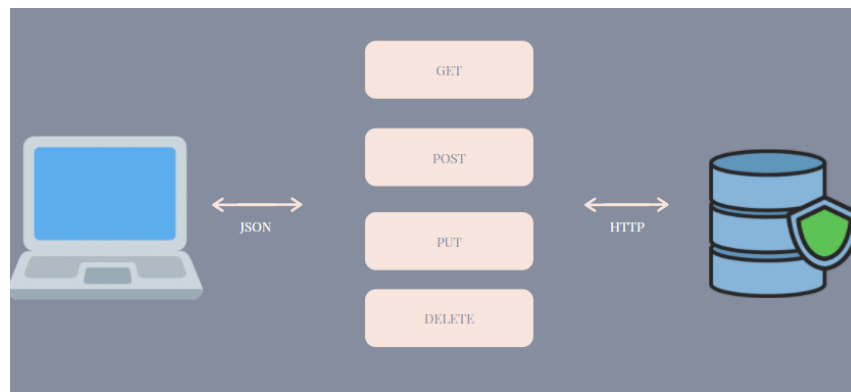


Ilustración 2 Gráfico API, adaptado A. (2020b, diciembre 3)

## 8.12. JSON

“JSON, es el formato de datos más popular para enviar solicitudes y respuestas API [...] los desarrolladores pueden especificar la estructura de los documentos JSON. La sintaxis y semántica para los esquemas JSON, pueden facilitar a la obtener de obtener datos.” (Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., & Vrgoč, D. 2016, April). Para las peticiones este esquema es bastante completo, además, de que se pueden modificar como realmente deseemos obtener las peticiones extraídas de las bases de datos, esto se generan desde el back-end y se operan los datos, el ejemplo de un JSON es el siguiente:

```
{“name”: “Danny”}
```

## 8.13. Base de datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. (2021, Oracle)

## 8.14. Bases de datos relacionales

Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de bases de datos relacionales proporciona la forma más eficiente y flexible de acceder a información estructurada. (Oracle, 2021)



### 8.15. PostgreSQL

“PostgreSQL es un poderoso sistema de base de datos relacional de objetos de código abierto que usa y extiende el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas.”

Este sistema al ser tan completo facilita bastante la creación de bases de datos, también como su administración, entonces es un sistema ideal a la hora de gestionar los datos en un sistema de información, Además, que otorga la facilidad de conectarse a otras bases de datos a la hora de hacer consultas. (PostgreSQL, 2021)

### 8.16. Scrum

“Scrum es una metodología ágil para desarrollar productos, servicios y proyectos innovadores [...] tiene un enfoque de desarrollo simple, común y ágil, tiene una lista de características otras capacidades para el desarrollo de un producto exitoso.” (S., R. K., 2012)

Para la metodología SCRUM tiene 3 principales actividades que son las siguientes:

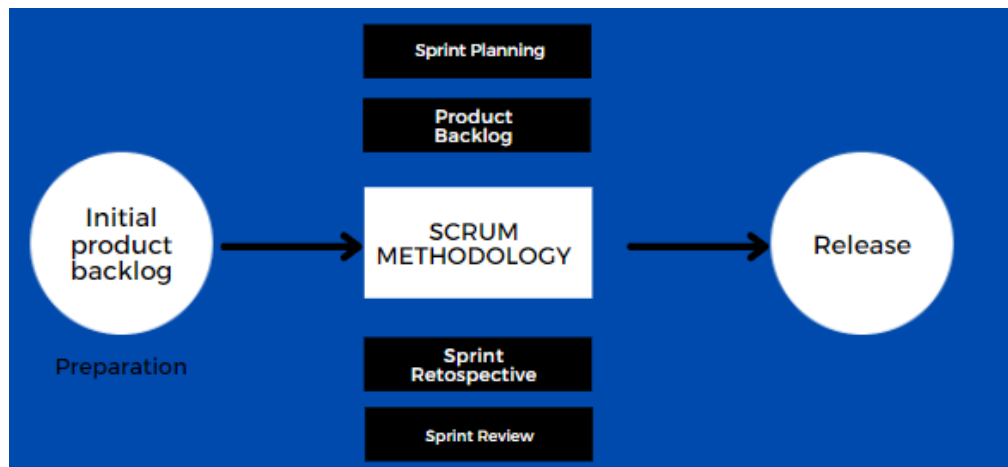
- Sprint: “Es El núcleo central de la metodología de trabajo ‘scrum’ es el ‘sprint’. Se trata de un mini proyecto de no más de un mes (ciclos de ejecución muy cortos -entre una y cuatro semanas), cuyo objetivo es conseguir un incremento de valor en el producto que estamos construyendo. Todo ‘sprint’ cuenta con una definición y una planificación que ayudará a lograr las metas marcadas.” (BBVA., 2021, 6 octubre)
- Product backlog: “Una de ellas es el ‘product backlog’, que podría definirse como un listado ordenado y priorizado de los requisitos necesarios para la implementación de un proyecto. Debe contener descripciones de las funcionalidades y características que

se quieran en el producto, ordenadas en función de su prioridad.” (Cañete, I., 2019, 11 marzo).

Sprint backlog: “Subconjunto de objetivos/requisitos del Product Backlog seleccionado para la iteración actual y su plan de tareas de desarrollo.”

- La metodología SCRUM tiene los siguientes beneficios:
  - Flexibilidad a cambios
  - Mayor productividad
  - Predicciones de tiempo
  - Reducción de contingencias

En la ilustración 3 se puede ver todas las actividades que se completaron en todo el entorno de trabajo del proyecto.



*Ilustración 3 Diagrama de actividades adaptado de Scrum A. (2021a, septiembre 8)*

## 9. ESTADO DEL ARTE

Esta última década se ha enfocado mucho la gestión documental, entonces a base de esto los sistemas de bases de datos están llenos de estos documentos, por esta razón se debe pensar la generación de reportes. Según (Correa, J. A., Vásquez, L., Soto, Y. M., & Hernández, M. C. 2016) “Los reportes integrados satisfacen la actual demanda de información de los grupos de interés que necesitan tomar mejores decisiones y observar en ellos la generación de valor de las empresas [...] para evaluar el cumplimiento de su revelación según algunas guías internacionales, teniendo como resultado que las empresas dan la suficiente importancia a la revelación de información integrada”.

Los datos son de gran valor dependiendo del contexto en el que se usen, en este caso son de gran relevancia debido a que ser documentos de personas, son de gran valor y confidencialidad según (marketing, 2021, 13 enero) “Además, tener el control de los datos de la empresa es esencial para que las otras partes que necesitan de esos datos, los tengan disponibles para ejecutar acciones.” Estos datos deben ser controlados y manipulados de manera correcta para tener un buen planteamiento a la hora de generar los reportes y que estos cumplan los estándares requeridos.

La generación de reportes es de gran utilidad para las personas encargadas de gestionar las bases de datos con documentos, ya que, al tener gran cantidad de datos, con los reportes se facilita saber el comportamiento de las bases. Al tener un gran control de

esta información se puede gestionar para filtrar la información de los datos que se deseen obtener basado en las bases de datos.

Las ventajas de los reportes son bastante importantes y que las empresas deberían adoptar al generar estos reportes estos son:

- Adquisición de datos: “La adquisición de datos es lo que hace que los reportes tengan gran importancia para entender la situación actual de todo lo que sucede dentro de la empresa.” (Marketing. 2021b, enero 13)
- Control de procesos: “Además de identificar riesgos y fallas, los reportes ayudan a equipos de IT en cuestiones relacionadas a la manutención de los dispositivos y el control de procesos.”

## 10. METODOLOGÍA

### 10.1. ¿Qué es metodología agile?

“El movimiento de metodología ágil comenzó en la década de los 90, para rechazar métodos aburridos y al parecer lentos. Los métodos antiguos carecían de cumplimiento de tiempos pactados y sobre costos a la hora de calcular presupuestos la metodología agile ofrece mejoras en estos aspectos. La metodología ágil tiene ideas centrales que prescriben métodos iterativos.” (Inflectra, 2020-1 diciembre)

Esta metodología es bastante completa a la hora de desarrollar software ya que al ser una metodología que tiende a la agilidad, genera un entorno de trabajo adecuado para estos desarrollos. Además, que permite que al trabajar de manera colaborativa pone orden en los equipos de trabajo.

### 10.2. Analizar la manera en los que se generar reportes

Para este proyecto se analizó los reportes que se generaban antes de manera manual, de esta forma se buscó la mejor manera en la que se debían originar estos reportes con los requerimientos que se originaron en un principio.

### 10.3. Metodología ágil elegida

El desarrollo del sistema de información se basó en el marco de trabajo para el desarrollo ágil de proyectos de software Scrum, que nos sirvió de apoyo para el ciclo de organización del sistema, esta metodología es perfecta porque brinda facilidad para integrar buenas prácticas a la hora del desarrollo. Este marco de trabajo se usó en el

equipo de trabajo el cual cuenta con un desarrollador junior en este caso yo y un encargado del seguimiento del proyecto llamados Scrum Máster.

Equipo de desarrollo, en este caso como fue un trabajo de una persona, esta se encargó de realizar el desarrollo y velar por cumplir con todos los sprints propuestos.

El Scrum master, la persona que se encargó de hacer el seguimiento del Proyecto como líder del equipo, esta se encargó de que las iteraciones y de que si realmente fueron útiles.

#### 10.4. Levantamiento de requerimientos

Al analizar los resultados de la anterior metodología ágil, se determinó las necesidades principales para el desarrollo, se tuvo en cuenta los siguientes puntos para definir las soluciones que más se acoplen a los requerimientos. Los requerimientos están basados en el product backlog.

- Requerimientos funcionales.
- Requerimientos no funcionales.
- Diagramas UML.

## 11. IMPLEMENTACIÓN

En esta etapa, se conectó todas las partes del desarrollo, se consumirá el API, y se completó la aplicación donde ya es funcional, esto hace parte de la etapa final del sistema de información.

### 11.1. Tecnologías asignadas por el Scrum master para el Desarrollo

Al empezar el proyecto se analizó cada una de las tecnologías que se usaron para el desarrollo, entonces se deben investigar de manera completa cada una de estas tecnologías. En esto se buscó las tecnologías que fueran más adecuadas para el desarrollo correcto del proyecto, acá se definió los aspectos a tener, al ser un proyecto que solo trabajo una persona se tuvieron en cuenta los siguientes aspectos:

#### 11.1.1. Reuniones

##### 1. Sprint Planning

En esta reunión se realizó el comienzo de cada uno de los Sprints; esto sirvió para definir el backlog y poder empezar a trabajar en cada uno de los ítems propuestos. Además de poder planificar las estimaciones de tiempo para el desarrollo de cada sprint.

##### 2. Daily Scrums

Se hicieron diferentes reuniones a lo largo del desarrollo, en diferentes días donde se observe el trabajo de cada uno de los Sprints y que se debe mejorar

en las épicas. Estas reuniones se hicieron entre el Scrum master y yo como desarrollador.

### 3. Sprint Review:

Estas reuniones se hicieron para comprobar que se haya cumplido los sprints que se colocaron en la reunión Sprint Planning. Principalmente nos enfocamos en el correcto funcionamiento del desarrollo.

#### 11.1.2. Diseño

Al iniciar el proyecto se debe estructurar el back-end y la base de datos, donde se almacenarán y consumirán los datos que se van a usar. Ya para el front-end se diseñaron cada una de las vistas donde se verán reflejados los datos que se van a consumir desde el back-end.



### 11.1.3. Resultados de la metodología Scrum

El Proyecto de trabajo se desarrolla desde la plataforma Jira, se trabajó desde las tres ramas (back-end, front-end y la base de datos) para la evidencia se generó la siguiente grafica de los sprints representada en la ilustración 4:

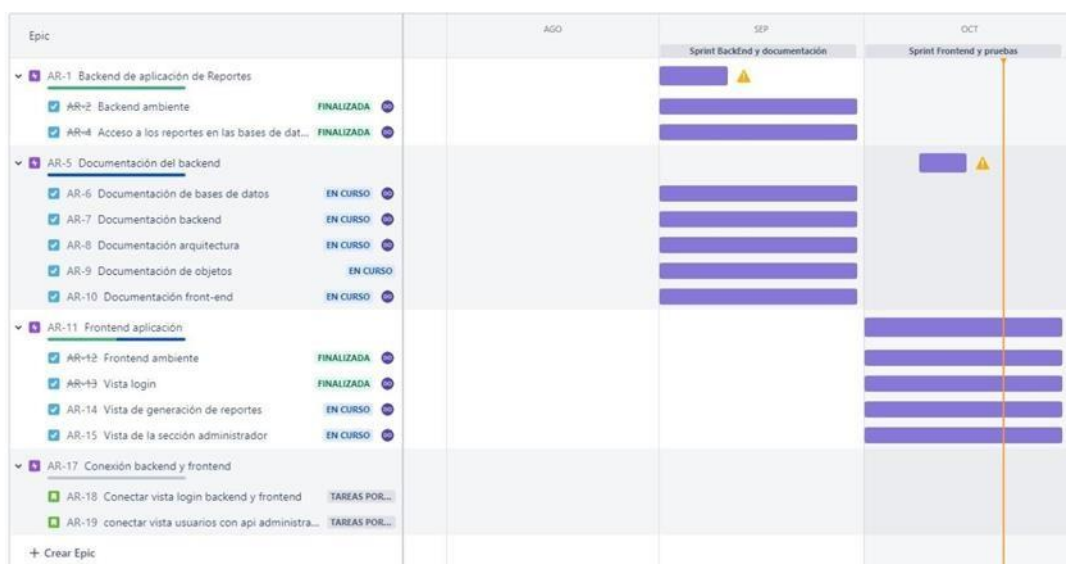


Ilustración 4 Tabla de sprints (elaboración propia)

Para iniciar los sprints se hacen de la siguiente manera, donde definimos el objeto del sprint, la duración, cuando se inicia, cuando finaliza por último el objetivo de cada sprint se hace de la siguiente de la forma:

**Iniciar sprint**

Incidencias que se incluirán en este sprint: **8** incidencias.

Nombre del sprint \*

Sprint Frontend y pruebas

Duración \*

personalizado

Fecha de inicio \*

1/10/2021 10:35

Fecha de inicio prevista: 1/10/2021 0:00  
La fecha de inicio de un sprint afecta a la velocidad y el alcance de los informes. Obtén más información.

Fecha de finalización \*

29/10/2021 10:35

Objetivo del sprint

Se desarrollara la parte gráfica de la aplicación con sus correspondientes vistas y funcionalidades.

**Iniciar** Cancelar

*Ilustración 5 Crear sprint (elaboración propia)*

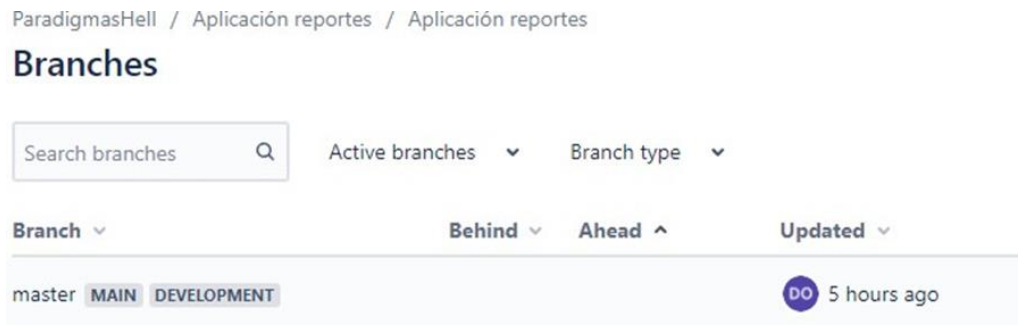
El proyecto se trabajó desde teletrabajo, todas las reuniones que se hicieron para el seguimiento del sistema se hicieron con el apoyo de la herramienta virtual Microsoft Teams, en los canales provistos por la empresa MTI. Para ver el desarrollo y el avance que está teniendo el proyecto se usa el software de Atlassian Jira y para ver código del proyecto se usa el gestor de repositorios BitBucket que esta previamente conectado a Jira.

Para ver el avance de cada uno de los sprints se hace de la siguiente forma de manera grafica.



*Ilustración 6 Progreso de un sprint (elaboración propia)*

El repositorio fue creado en BitBucket, solo se usó un único brazo el cual es el main, ya que fue mejor trabajar desde un solo brazo debido a que solamente hubo un colaborador, en la siguiente imagen se puede ver como se trabajaba.



*Ilustración 7 Repositorio en BitBucket elaboración propia*

Para identificar cada desarrollo que se hacía se utilizaron commits para representar cada uno de los adelantos que se hacían en el brazo principal, en la siguiente imagen se tomó un fragmento desde BitBucket donde se realizan cada uno de los commits. En estos commits se ve el autor, la identificación del commit, el mensaje de cada commit y por último la fecha de cada uno de estos.

Commits

Search commits  All branches ▼

Author	Commit	Message	Date
Danny Ochoa	4459d3e	MERGED Merge branch 'master' of https://bitbucket.org/paradigmashell/apl...	5 hours ago
Danny Ochoa	2593b39	Ajuste rol en backend	5 hours ago
Danny Ochoa	32c2ae0	detalles	3 days ago
Danny Ochoa	4a12602	Arreglo iconos	3 days ago
Danny Ochoa	0f7c3d5	Corrección carga al eliminar	3 days ago
Danny Ochoa	a6574cc	Ajuste	4 days ago
Danny Ochoa	cf73551	Avance	6 days ago
Danny Ochoa	a025ff2	isAdmin en front	2021-11-09
Danny Ochoa	f745793	Borrado login admin	2021-11-09
Danny Ochoa	734b552	Se ajusto rol a la hora de editar usuario	2021-11-09
Danny Ochoa	66fb8ff	Ajuste del back para el rol	2021-11-09
Danny Ochoa	b72fb10	Se le añadió el rol a la hora de agregar usuarios y verlos	2021-11-09
Danny Ochoa	ac1e976	Role añadido en el backend	2021-11-08

*Ilustración 8 Commits en repositorio elaboración propia*

Frontend aplicación /  AR-14

Vista de generación de reportes

Adjuntar  Vincular incidencia ▼

Descripción

Se genera la vista donde se accedera a los sistemas de reportes donde se podran visualizar y descargar los correspondientes informes deseados.

Actividad

Mostrar: Todo Comentarios Historial Más recientes primero ⌵

Añadir un comentario...

Consejo de expertos: pulsa **M** para comentar

En curso ▼

Detalles

Responsable Danny Ochoa

Etiquetas Ninguno

Sprint Sprint Frontend y pruebas

Story point estimate 5

Desarrollo [Crear rama](#)

Informador Danny Ochoa

Creado 9 de octubre de 2021 17:19 ⚙️ Configurar

Actualizado 21 de octubre de 2021 10:33

*Ilustración 9 Desarrollo de una épica elaboración propia*

Para ver los requerimientos y el proceso en el que se lleva se puede definir en tres aspectos, “Tareas por hacer”, “En curso” y “Finalizada”. Esto se puede observar desde el backlog, en la ilustración 10 se puede ver una evidencia de esto.

Proyectos / Aplicación reportes

### Backlog

🔍 DO 👤 Epic Tipo Insights

<input checked="" type="checkbox"/>	AR-7 Documentación backend	DOCUMENTACIÓN DEL BACKEND	3	FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-6 Documentación de bases de datos	DOCUMENTACIÓN DEL BACKEND	3	FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-4 Acceso a los reportes en las bases de datos.	BACKEND DE APLICACIÓN DE REP...		FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-12 Frontend ambiente	FRONTEND APLICACIÓN		FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-15 Vista de la sección administrador	FRONTEND APLICACIÓN	3	FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-13 Vista login	FRONTEND APLICACIÓN		FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-14 Vista de generación de reportes	FRONTEND APLICACIÓN	5	FINALIZADA	DO
<input checked="" type="checkbox"/>	AR-16 Pruebas		5	EN CURSO	DO

*Ilustración 10 Backlog elaboración propia*

## 12. ARQUITECTURA DE SOFTWARE

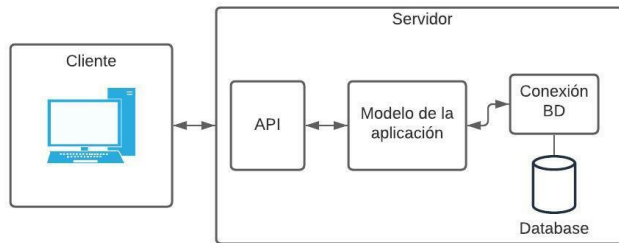
Se definió las especificaciones a la hora de implementar la arquitectura de software pensando en las épicas a cumplir, identificando las necesidades, los requerimientos, la problemática, los objetivos y el modelo.

Para esto evidenciamos cómo funciona esta arquitectura, definimos la problemática que establecimos desde un inicio para abarcar los requerimientos del sistema, pensando principalmente en el software, tanto como evidenciar en tiempo real el uso de la información para los usuarios web.

### 12.1. Diseño de la arquitectura

Esta aplicación se puede describir como un componente de software que está en un servidor de la empresa en el cual se puede acceder a través de internet estando conectado a la red de la empresa, esta arquitectura cuenta con métodos que facilitan la necesidad de acceder a la información pensando en requerir del ciclo de vida, considerando esto la arquitectura que más se acopla a esto es la “web service”.

Pensando en lo anterior se trabajó en a base de un diagrama que explica la funcionalidad de esta arquitectura donde el sistema se conecta al servicio web accediendo a la información de la base de datos esto se evidencia en la ilustración 11.



*Ilustración 11 Arquitectura web service (elaboración propia)*

Se basó en los requerimientos planteados anteriormente definimos e implementamos la fórmula de la arquitectura. Se trabajó en módulos que se acoplan dependiendo del software haciéndolo compatible, esto se define de la siguiente forma:

1. Cliente: Desde esta parte se consumen los servicios API, enviando y recibiendo peticiones definidos anteriormente, esto representa las respuestas que se generan en la salida y la entrada.
2. Modelo de la aplicación: Aquí es donde se generan las peticiones que son procesadas, proviniendo los datos de la base de datos.
3. Conexión base de datos: Este es el que interactúa con las tablas, generando consultas desde la base de datos.

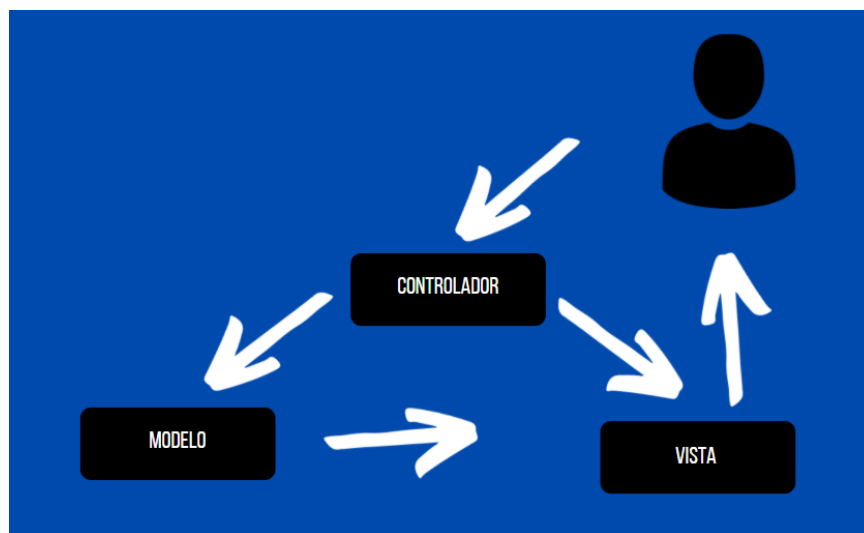
Para el desarrollo de la arquitectura se usó el framework Node.js que permitió acoplar el modelo a los requerimientos, se desarrolló de la siguiente manera:

El modelo permite caracterizar la información con la que debe interactuar el sistemas.

La Vista, se usa para representar el modelo anteriormente descrito en una página web al cual el usuarios va a poder interactuar.

El controlador permite trabajar las acciones del usuario que realice al interactuar con el sistema.

Para representar la arquitectura MVC, se tuvo en cuenta de que esta arquitectura permite separar tanto la lógica como la parte visual de la aplicación, este modelo facilita muchas de las interacciones de las aplicaciones como su mantenimiento, para ver esta arquitectura está representada en la ilustración .



*Ilustración 12 Patrón MVC enfocado en la arquitectura basado en Pantaja, B. E. (s. f.).*



## 13. REQUERIMIENTOS

### 13.1. Requerimientos funcionales

Los requerimientos funcionales de este proyecto se pueden observar en la tabla número 1 donde son basados de las historias de usuarios. Estos requerimientos están enfocados en la parte tanto como de administrador y usuarios de las funcionalidades de la aplicación.

*Tabla 1 Requerimientos funcionales (elaboración propia)*

ID.	Rol	Característica / Funcionalidad	Razón /Resultado	Criterio de aceptación
1	Como usuario	Quiero ingresar con mi correo y contraseña.	Para poder acceder al sistema de información.	Las credenciales serán ingresadas desde la interfaz de la aplicación y serán comprobadas desde el back-end.
2	Como usuario	Quiero ingresar a la vista principal.	Para poder acceder a las tarjetas de las bases de datos que contendran los reportes.	Cada base de datos tendrá una tarjeta con un enlace que mandará a los reportes de la base de datos.
3	Como administrador	Quiero registrar usuarios desde un formulario, donde se almacena el nombre, correo, contraseña y rol.	Para que puedan acceder al sistema de información dependiendo del rol que le administre.	Los datos que se ingresen quedaran almacenados en la base de datos
4	Como administrador	Quiero consultar, editar y eliminar los datos de cada uno de los usuarios con los datos anteriormente almacenados.	Para poder gestionar los datos que allí esten almacenados de cada usuario.	Los datos se consultarán desde la base de datos y se consumira dependiendo de la acción correspondiente.
5	Como administrador/usuario	Quiero consultar todos los reportes de la base de datos seleccionada.	Para poder gestionar los datos que allí esten almacenados de cada usuario.	Los datos se consultarán desde la base de datos y se consumira dependiendo de la acción correspondiente.
6	Como administrador/usuario	Quiero poder salir de la aplicación.	Para poder salir de manera satisfactoria y dejar la sección cerrada.	Al salir si se quiere volver a ingresar a la aplicación se necesita volver a usar las credenciales.

### 13.2. Requerimientos no funcionales

Estos requerimientos no funcionales están basados en las historias de usuarios, estos describen los requisitos de los sistemas especificando que se debe cumplir para que el desarrollo funcione de una manera óptima, estos requerimientos los podemos observar en la tabla número 2.

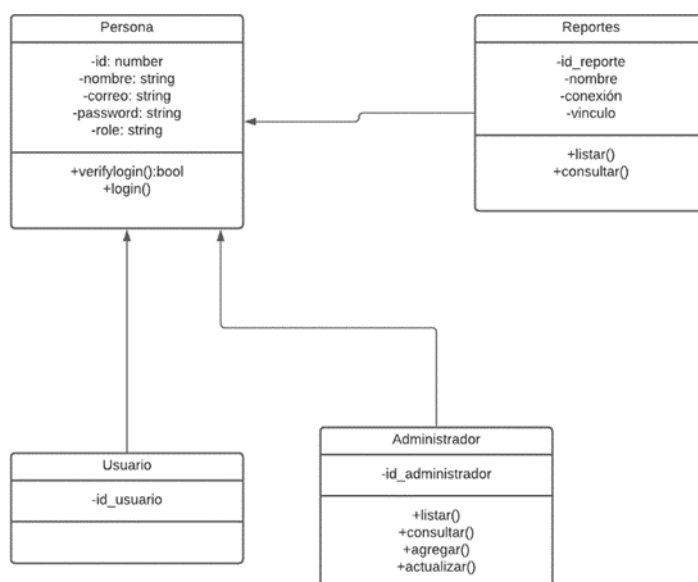
*Tabla 2 Requerimientos no funcionales (elaboración propia)*

ID	Descripción
RNF01	El sistema deberá estar disponible los 365 días del año, las 24 horas del día.
RNF02	Los datos quedaran almacenados en la base de datos, tal cual ingresa en la interfaz, deben coincidir con los datos ingresados para un buen funcionamiento de la aplicación.
RNF03	La aplicación debe soportar actualizaciones tanto en el back-end, front-end y base de datos al ser la primera versión, la aplicación aceptara actualizaciones.
RNF04	La aplicación deberá ser compatible con los principales navegadores para su correcto funcionamiento.
RNF05	La aplicación debe ajustarse a distintas resoluciones para una correcta visualización del sistema.

## 14. DIAGRAMAS UML

### 14.1. Diagrama de clases

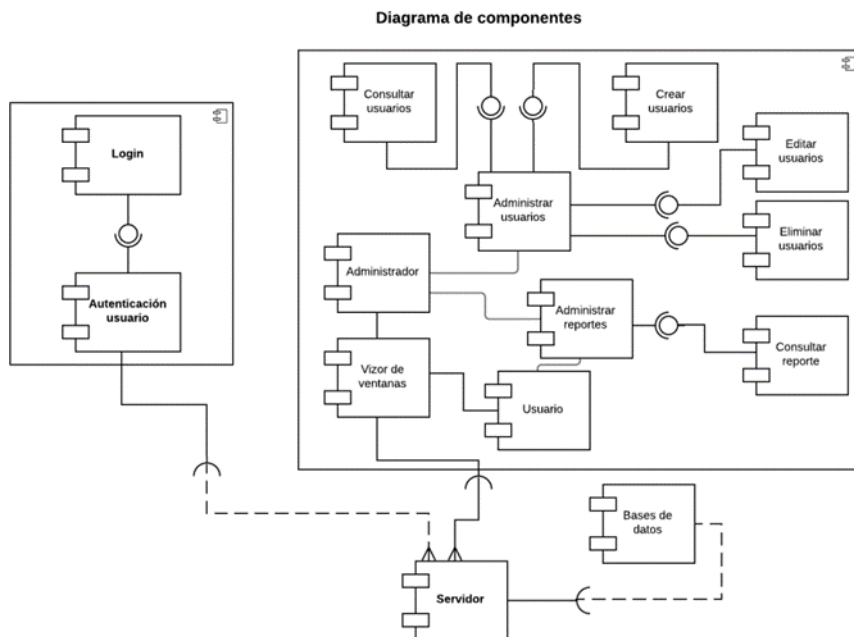
El primer diagrama de clases funciona para definir que clases se usaron, capturando ideas y representando los objetos que se utilizaron, en este se definió la persona con sus características donde hace referencia si está en la tabla de usuarios o administrador, también se ve el objeto reportes con sus características.



*Ilustración 13 Diagrama de clases (elaboración propia)*

## 14.2. Diagrama de componentes

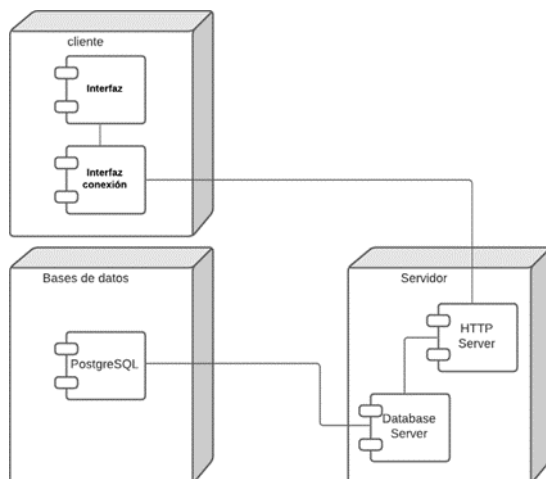
Este diagrama se usa para modelar los componentes que ayudan a definir las funcionalidades, representa la organización y sus dependencias, por ejemplo, en la ilustración 13 este diagrama se usa para definir los componentes y poder visualizar esta organización, tanto como la relación entre cada uno de los elementos del sistema. En este caso se representa los siguientes componentes: El login donde se ingresa a la aplicación, los elementos donde está la interfaz principal de la aplicación, tanto como para el usuario y el administrador, por último, el servidor con la conexión de base de datos.



*Ilustración 14 Diagrama de componentes (elaboración propia)*

### 14.3. Diagrama de despliegue

Este diagrama funciona para visualizar dispositivos de hardware del sistema. En la ilustración 14 se puede definir la arquitectura de la ejecución del sistema. Acá se representa el cliente que contiene los elementos de la interfaz, el servidor donde se puede observar cómo obtiene los servicios HTTP y la configuración para conectarse a la base de datos y por último la base de datos en este caso postgresSQL.



*Ilustración 15 Diagrama de despliegue (elaboración propia)*

#### 14.4. Diagrama de secuencias

Este diagrama funciona para modelar iteraciones entre objetos.

En la ilustración 15, en este diagrama de secuencia se logra ver la iteración a la hora de consultar un usuario por parte del administrador, esto ayuda a ver la forma en que se debe pensar a la hora de programar esta funcionalidad.

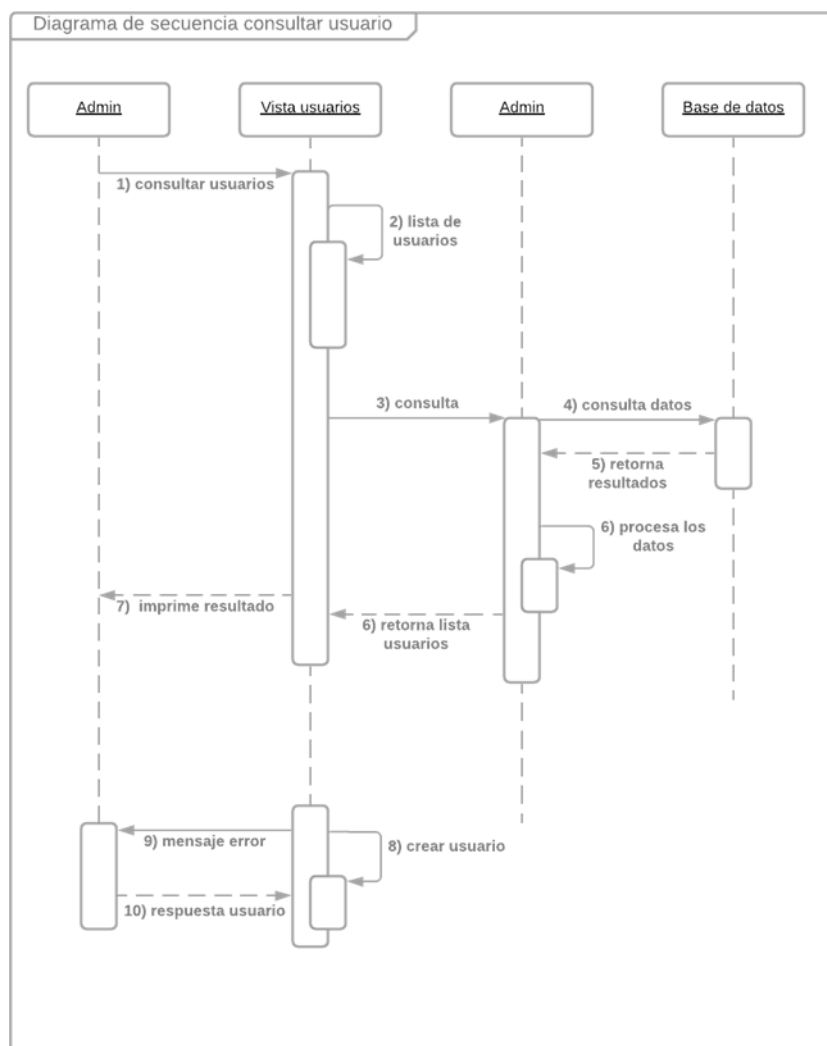


Ilustración 16 Diagrama de secuencia consultar usuario (elaboración propia)

En este diagrama de se puede ver las iteraciones que debe hacer un administrador a la hora de editar usuario. Es bastante funcional para pensar en cómo programar esta funcionalidad para poder evidenciar esto en la ilustración 16 se encuentra el diagrama.

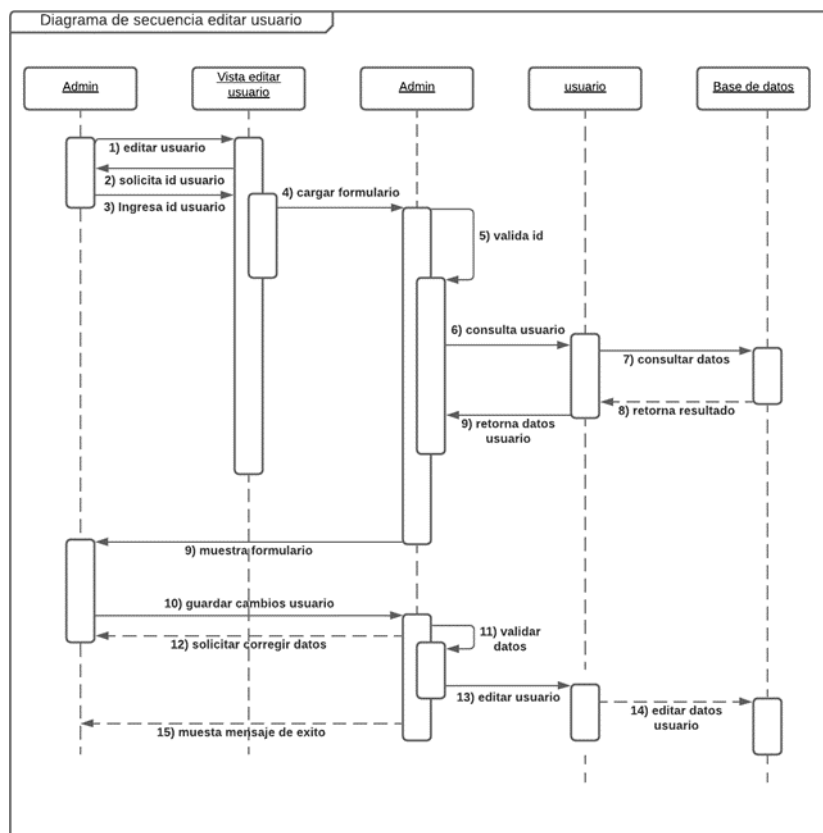


Ilustración 17 Diagrama de secuencia editar usuario (elaboración propia)

En este diagrama se puede analizar las iteraciones que se hace a la hora de generar reportes. Es bastante funcional para pensar en cómo programar esta funcionalidad. En la ilustración 17 se logra ver la secuencia de la consulta de reportes.

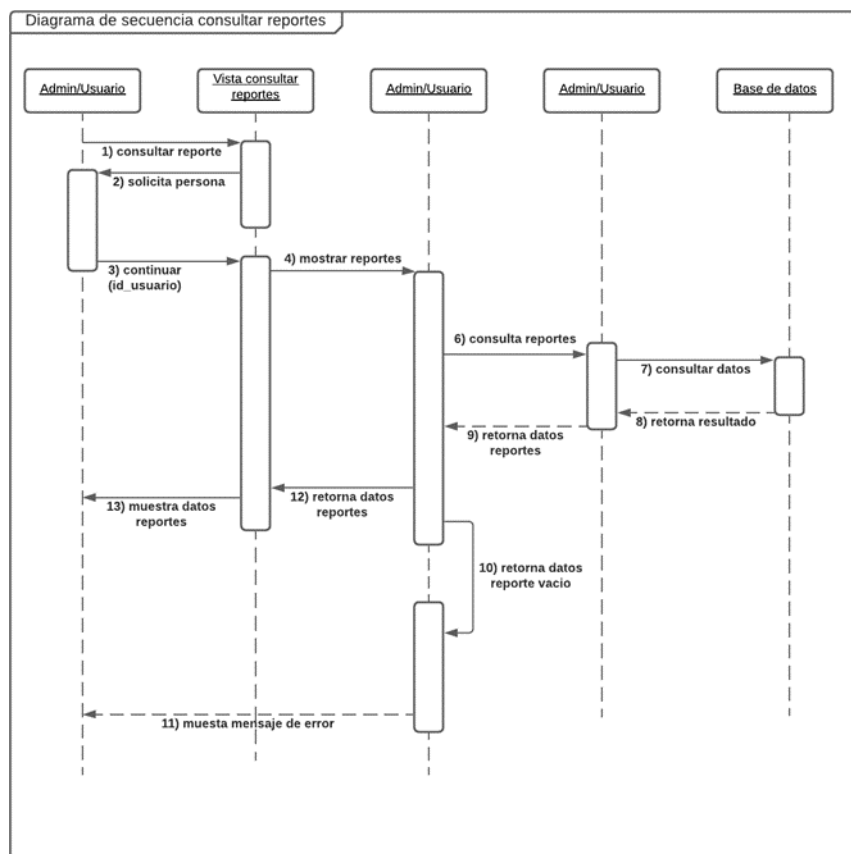


Ilustración 18 Diagrama de secuencia consultar reportes (elaboración propia)



En este diagrama de secuencia se logra contemplar la iteración a la hora de realizar un login, en la ilustración 18 se evidencia este diagrama, esto permite ver la forma en que se pensó para programar el login.

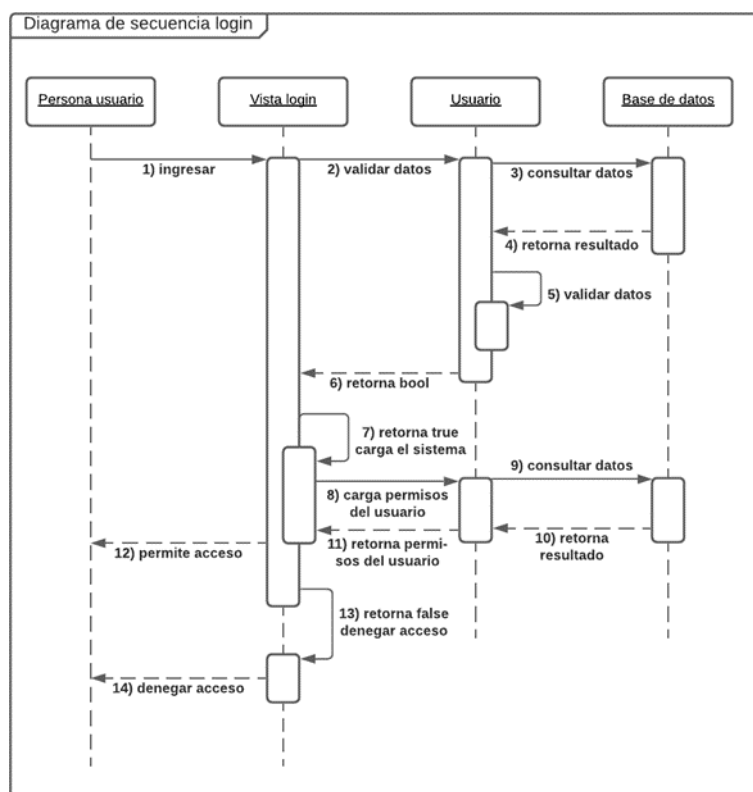


Ilustración 19 Diagrama de secuencia login (elaboración propia)

En este diagrama mostrado en la ilustración 19 se puede relacionar las iteraciones que debe hacer un administrador a la hora de eliminar un usuario. Es bastante funcional para pensar en cómo programar esta funcionalidad.

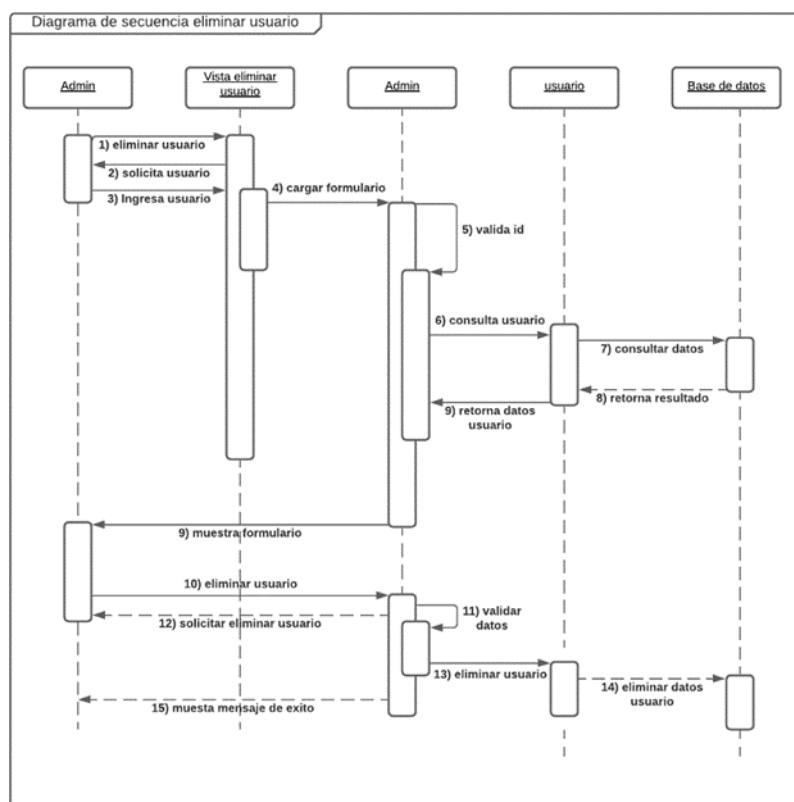


Ilustración 20 Diagrama de secuencia eliminar usuario (elaboración propia)

En este diagrama de secuencia se puede observar las iteraciones que debe hacer un administrador a la hora de registrar un usuario. Se puede ver en la ilustración 20 que es bastante funcional para pensar en cómo programar esta funcionalidad.

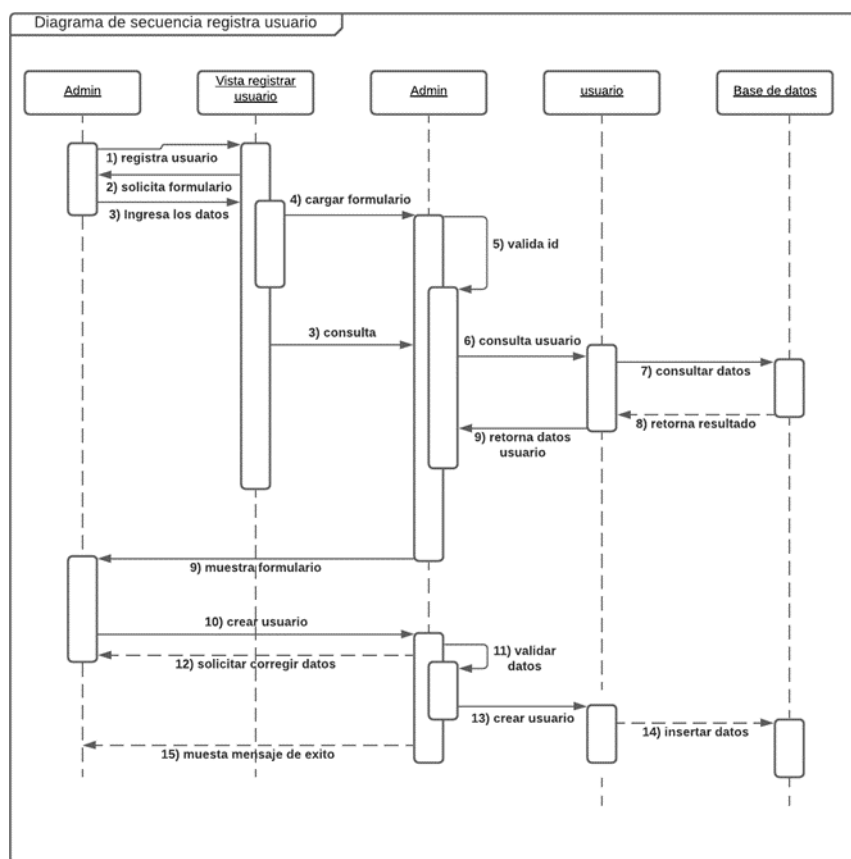
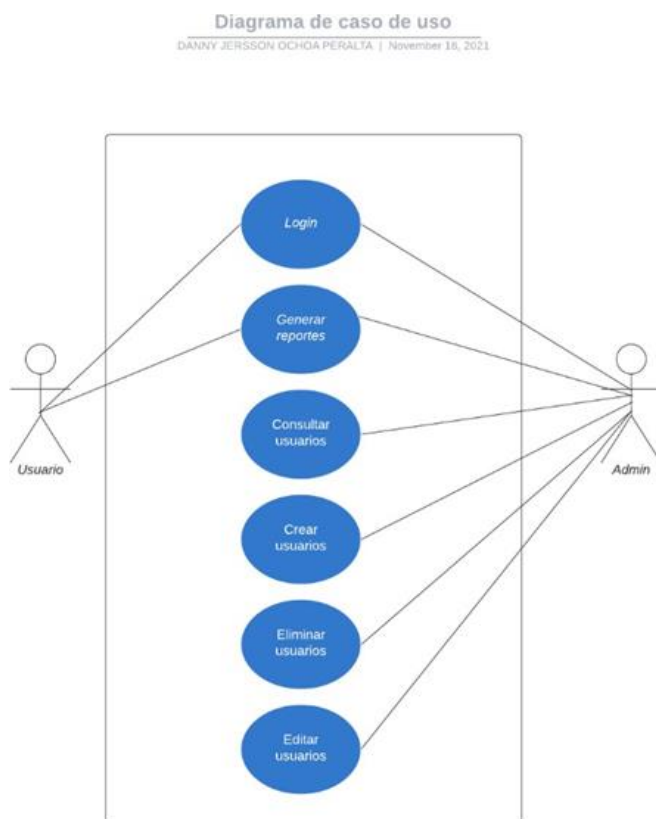


Ilustración 21 Diagrama de secuencia registro usuario (elaboración propia)

### 14.5. Diagrama casos de uso

El diagrama funciona para dar una breve descripción de cada una de las actividades que podrá realizar cada una de las entidades. Con esto se define que actor tiene acceso a la aplicación dependiendo del rol y cada una de las acciones que tiene permitida. En la ilustración 21 se ven los dos actores que se usaron y cada una de las actividades a las que pueden acceder.



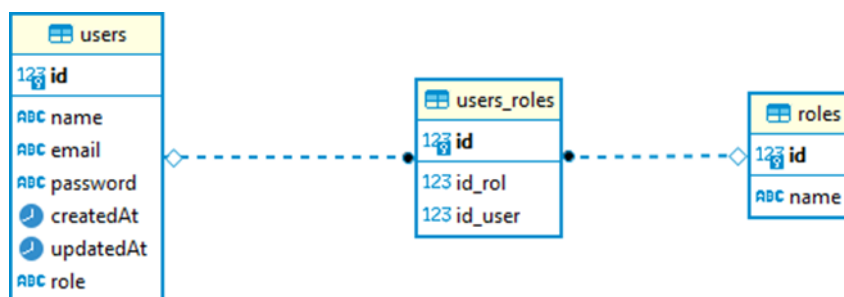
*Ilustración 22 Diagrama de casos de uso (elaboración propia)*

## 15. MODELO BASES DE DATOS

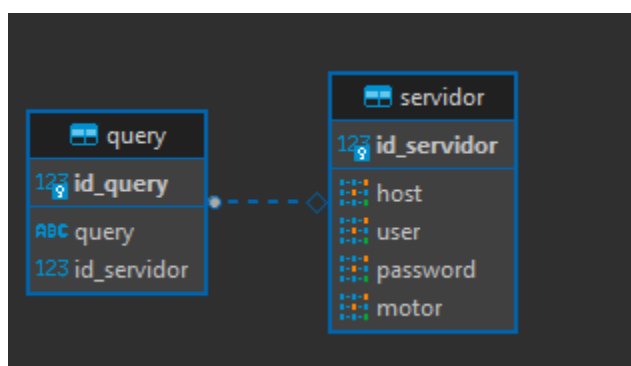
### 15.1. Diagrama entidad-relación

Este diagrama funciona para representar la arquitectura de la base de datos y sus relaciones en la ilustración 22 se observa la entidad usuarios a la cual se le relaciona un rol en una tabla intermedia que hace la conexión entre los usuarios y los roles.

Y en la ilustración 23 se ve la entidad query que es la conexión a un servidor para poder traer los reportes de las bases de datos.



*Ilustración 23 Diagrama entidad-relación (elaboración propia)*



*Ilustración 24 Diagrama generación de reportes (elaboración propia)*

## 16. RESULTADOS

### 16.1. Back-end

Obtuvimos los siguientes resultados al desarrollar el backend que es el API funcional.

#### **Dependencias**

Esta se genera para el entorno de trabajo, estas permiten usar dependencias facilitan el desarrollo adecuado del back-end de la aplicación.

```
"dependencies": {  
  "@hapi/joi": "^17.1.1",  
  "bcrypt": "^5.0.1",  
  "body-parser": "^1.19.0",  
  "cors": "^2.8.5",  
  "dotenv": "^10.0.0",  
  "express": "^4.17.1",  
  "jsonwebtoken": "^8.5.1",  
  "md5": "^2.3.0",  
  "mongoose": "^6.0.9",  
  "nodemon": "^2.0.13",  
  "pg": "^8.7.1"  
}
```

*Ilustración 25 Dependencias backend (elaboración propia)*



## Petición para obtener todos los datos de los usuarios:

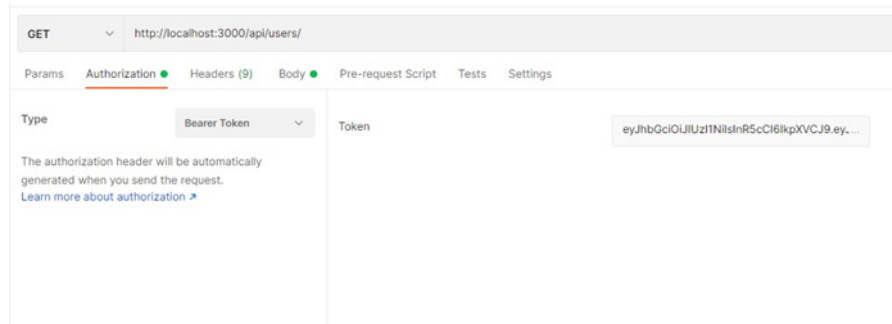


Ilustración 28 GET usuarios (elaboración propia)

Después de ejecutar se obtienen los usuarios de la siguiente manera:

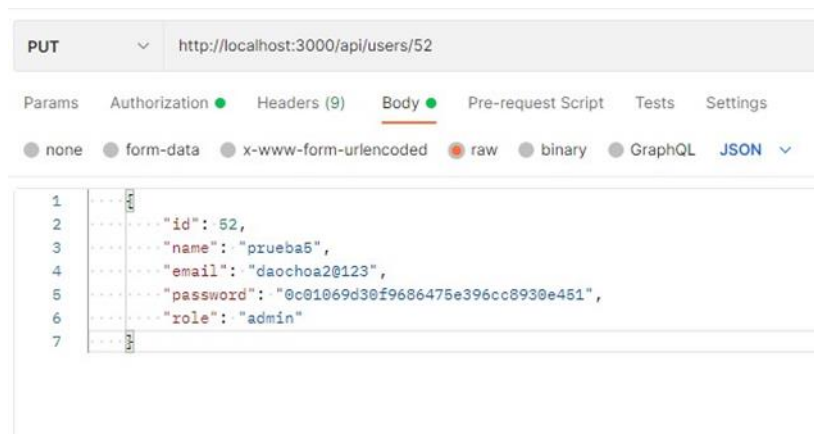
```
{
  "id": 51,
  "name": "prueba5",
  "email": "daochoa2",
  "password": "0c01069d30f9686475e396cc8930e451",
  "role": "admin"
},
{
  "id": 52,
  "name": "prueba5",
  "email": "daochoa2@123",
  "password": "0c01069d30f9686475e396cc8930e451",
  "role": "admin"
},
}
```

Ilustración 29 Respuesta GET usuarios (elaboración propia)



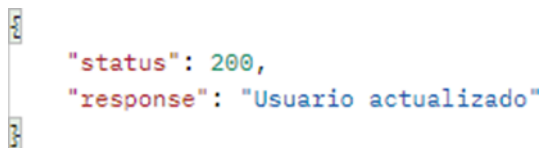
### Petición para editar los datos de un usuario:

Acá se edita el usuario después de validar el token, en el JSON se ingresa los datos que se van a editar.



*Ilustración 30 PUT usuario (elaboración propia)*

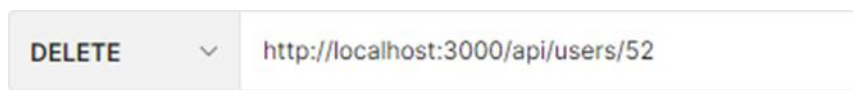
Después de editar si lo hizo de manera correcta obtenemos lo siguiente.



*Ilustración 31 Respuesta PUT usuario elaboración propia (elaboración propia)*

### Petición para borrar un usuario:

Para poder borrarlo se coloca el id del usuario que se desea borrar.



*Ilustración 32 DELETE usuario (elaboración propia)*

Si el usuario existe lo eliminara y arroja el siguiente mensaje mediante un JSON.

```
"status": 200,  
"response": "Usuario eliminado"
```

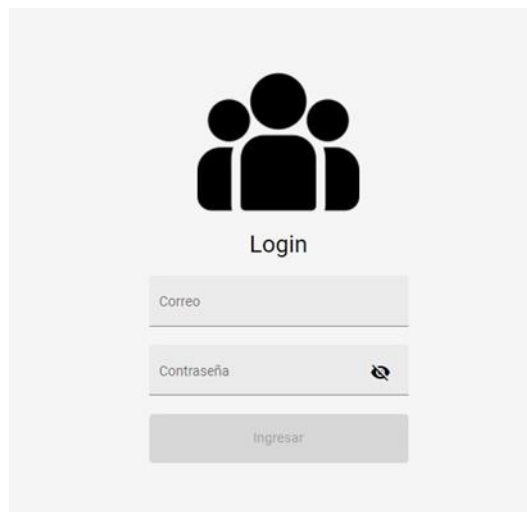
*Ilustración 33 Respuesta DELETE usuario (elaboración propia)*

## 16.2. Front-end

Los resultados obtenidos en el front-end son las vistas generadas en el framework de angular y el cual son funcionales a la hora de la ejecución del sistema en un navegador web.

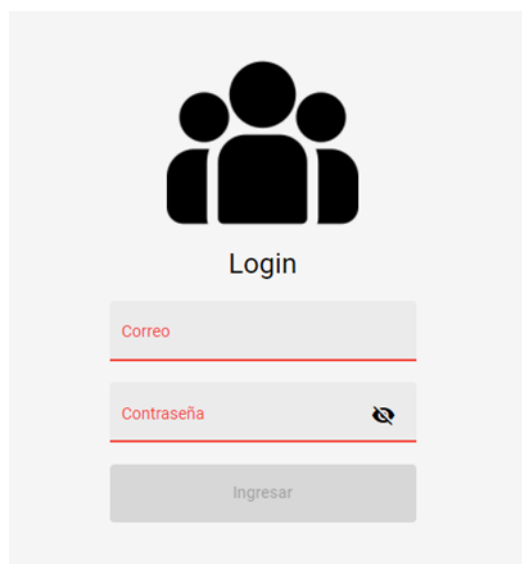
### **Vista login**

La primera vista que tendrán los usuarios es el Login donde el usuario podrá ingresar al sistema esto se puede evidenciar en la ilustración 34:



*Ilustración 34 Vista login de la aplicación (elaboración propia)*

La vista del Login si las credenciales no son ingresadas se puede ver en la figura 35:

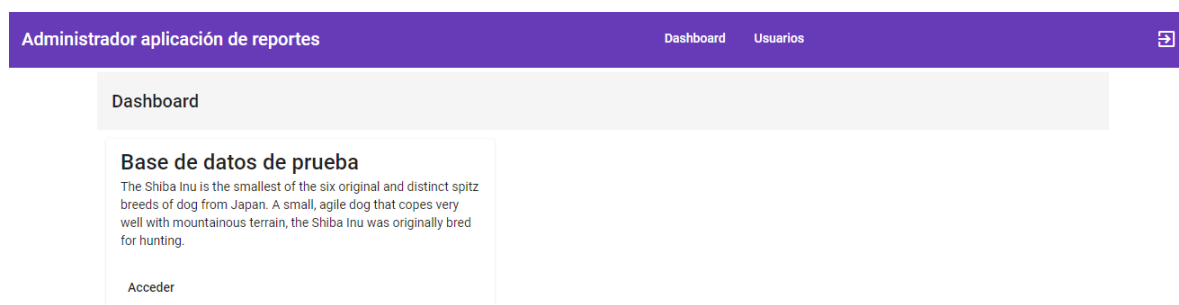


The image shows a login form with a light gray background. At the top center is a black icon of three stylized human figures. Below the icon is the word "Login" in a black sans-serif font. There are two input fields: the first is labeled "Correo" in red text and has a red horizontal line below it; the second is labeled "Contraseña" in red text and has a red horizontal line below it with a small black eye icon to its right. At the bottom is a gray button with the text "Ingresar" in a lighter gray font.

*Ilustración 35 Vista login fallido de la aplicación (elaboración propia)*

## Vista administrador

Si el usuario es un administrador tendrá la siguiente vista mostrada en la ilustración 36 donde estará habilitado un botón de usuarios.













*Ilustración 36 Vista admin de la aplicación (elaboración propia)*

Por ser un usuario administrador el cual tendrá una opción en el menú de navegación que se llama “usuarios”.

## Vista principal usuarios

En la ilustración 37 se puede ver los usuarios, cada registro se puede editar, eliminar y agregar, además de un filtro que permitirá buscar cada usuario.

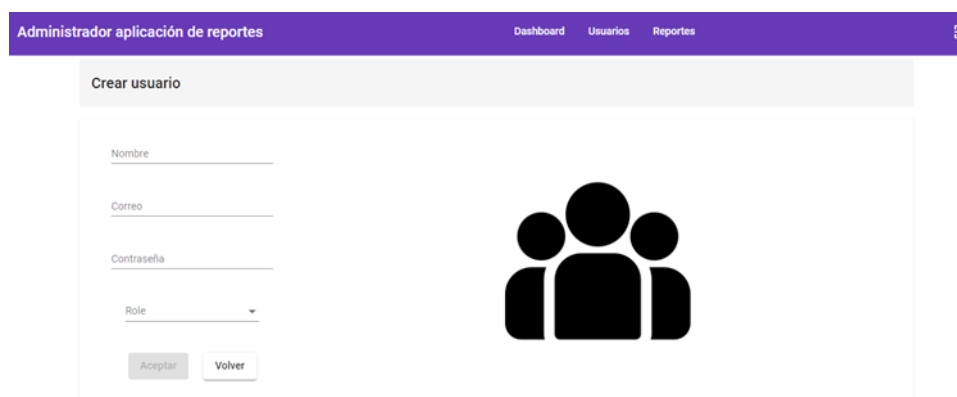
Id.	Nombre	Correo	Rol	
51	prueba5	daochoa2	admin	 
53	Jhon	jhon@	user	 
49	prueba5	daochoa	admin	 
54	Nico	nico	user	 
55	danny	danny	user	 

Items per page: 5 1 - 5 of 6 |< > >|

*Ilustración 37 Vista ver usuarios de la aplicación (elaboración propia)*

## Vista agregar usuarios

En esta vista observada en la ilustración 38 se tiene un formulario donde se deberá ingresar los datos pedidos del usuario nuevo, como el nombre, el correo, la contraseña y por último el rol.

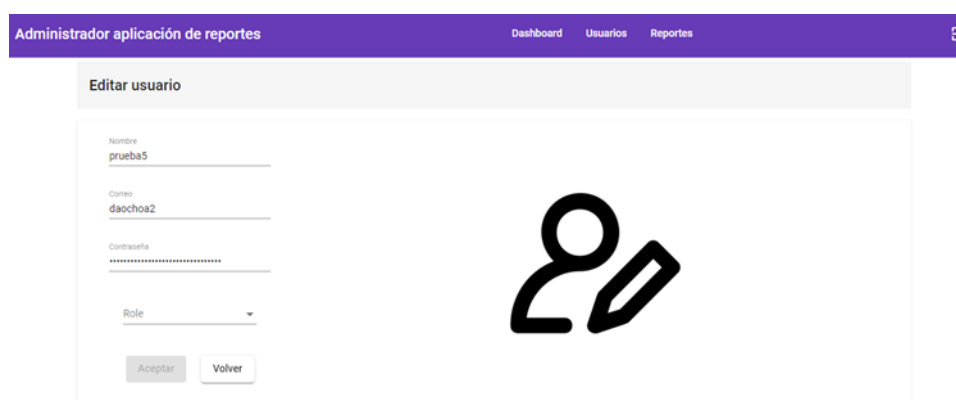


The screenshot shows a web application interface for user management. At the top, there is a purple navigation bar with the text 'Administrador aplicación de reportes' on the left and 'Dashboard', 'Usuarios', and 'Reportes' on the right. Below the navigation bar, the main content area is titled 'Crear usuario'. On the left side of this area, there is a form with four input fields: 'Nombre', 'Correo', 'Contraseña', and 'Role'. The 'Role' field is a dropdown menu. At the bottom of the form are two buttons: 'Aceptar' and 'Volver'. On the right side of the form, there is a large black icon representing three stylized human figures.

*Ilustración 38 Vista agregar usuarios de la aplicación (elaboración propia)*

## Vista editar usuarios

Cuando se selecciona para editar un usuario se cargan los datos de un usuario en un formulario y estos se pueden editar como cada administrador quiera como se puede observar en la ilustración 39:



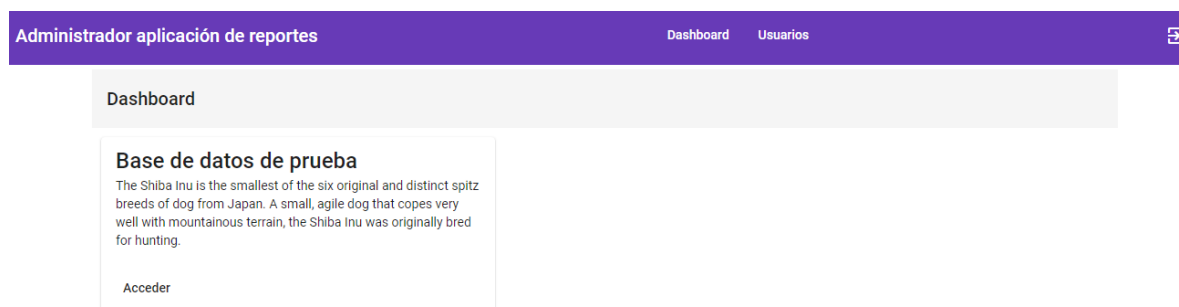
The screenshot shows a web application interface for editing a user. At the top, there is a purple navigation bar with the text 'Administrador aplicación de reportes' on the left and 'Dashboard', 'Usuarios', and 'Reportes' on the right. Below the navigation bar, the main content area is titled 'Editar usuario'. The form contains the following fields: 'Nombre' with the value 'prueba5', 'Correo' with the value 'dsachos2', 'Contraseña' (masked with dots), and a 'Role' dropdown menu. At the bottom of the form, there are two buttons: 'Aceptar' and 'Volver'. To the right of the form, there is a large black icon of a person with a pencil, symbolizing user editing.

*Ilustración 39 Editar usuario de la aplicación (elaboración propia)*



## Vista dashboard

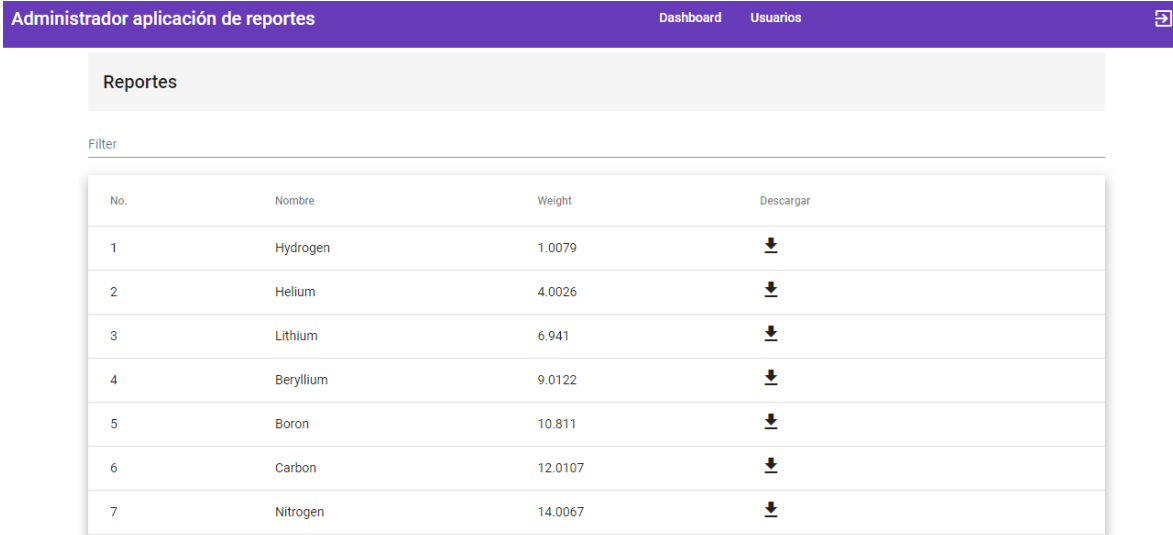
Esta vista la tendrán tanto los administradores como los usuarios. En esta parte se podrán acceder a los sistemas de información desde tarjetas, como la que se puede ver en la ilustración 40, si se ingresa a una tarjeta se ingresará a los reportes de cada base de datos:



*Ilustración 40 Vista dashboard de la aplicación (elaboración propia)*

## Vista reportes

Acá se podrán acceder a los reportes en una lista y se podrán descargar en formato Excel. En esta vista se podrá buscar cada uno de los reportes que se deseen obtener y de esta misma forma descargarlo con un botón, en la ilustración 41 se puede ver un ejemplo de cómo se verían los reportes de la aplicación.



Administrador aplicación de reportes Dashboard Usuarios

Reportes

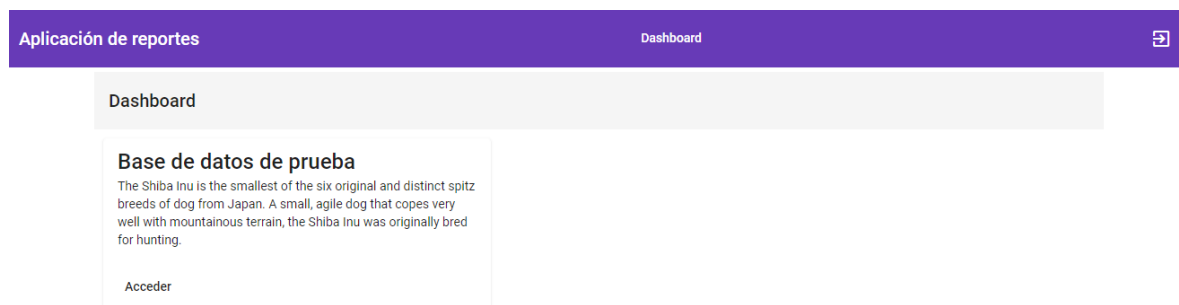
Filter

No.	Nombre	Weight	Descargar
1	Hydrogen	1.0079	↓
2	Helium	4.0026	↓
3	Lithium	6.941	↓
4	Beryllium	9.0122	↓
5	Boron	10.811	↓
6	Carbon	12.0107	↓
7	Nitrogen	14.0067	↓

*Ilustración 41 Vista reportes de la aplicación (elaboración propia)*

## Vista usuarios

En esta vista el usuario solo tendrá desbloqueado el dashboard donde estarán las tarjetas con cada una de las bases de datos evidenciado en la ilustración 42:



*Ilustración 42 Vista usuario de la aplicación (elaboración propia)*

## 17. CRONOGRAMA

El cronograma se divide en diferentes actividades con intervalos de tiempo, a continuación, en una lista se ven las actividades y en la ilustración 43 un diagrama donde se representa con cada uno de sus tiempos de ejecución.

- Recolección de datos de la empresa.
- Reunión para requerimientos.
- Crear los mockups.
- Análisis y Desarrollo de la base de datos.
- Desarrollo del software.
- Pruebas.
- Despliegue de la aplicación.
- Documentación.

Actividad/Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Recolección de datos	✓	✓	✓											
Reunión de requerimientos				✓	✓									
Creación de los mockups					✓	✓	✓							
Análisis y desarrollo de la base de datos							✓	✓						
Desarrollo del software								✓	✓	✓	✓	✓	✓	
Pruebas													✓	✓
Documentación	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

*Ilustración 43 Diagrama de Gantt de la aplicación (elaboración propia)*

## 18. CONCLUSIONES

La primera conclusión de este trabajo fue el progreso y terminación del sistema de información que se llevó a cabo para mejorar el uso de la información dentro de la empresa; con esto resaltamos los requerimientos de las empresas actualmente de generar reportes de información, por eso este tipo de sistemas son de gran utilidad, cumpliendo con los requerimientos necesarios.

Teniendo en cuenta las necesidades encontradas en la empresa por el personal a la hora de generar reportes, se logra crear un sistema tecnológico que automatiza la generación de reportes, aumentando la productividad de la empresa.

La aplicación diseñada y deja en marcha hace que el proceso para lo que fue desarrollada sea totalmente funcional a un 100% esto hace que sea utilizable para toda el área de tecnología de MTI.

Los reportes arrojados por el sistema son fáciles de entender y de analizar correspondientemente a las necesidades de cada usuarios, con un buen flujo de información. El sistema cuenta con una autenticación sencilla, que le permite estar dentro de la aplicación por un tiempo determinado, logrando una mayor seguridad, logrando entrar al sistema las veces que sean necesarias.

Con las preguntas surgidas a lo largo de las practicas, con este trabajo se logró suplir las necesidades manifestadas desde un inicio, dando respuesta a estas preguntas, como la funcionalidad de la aplicación y si era suficiente para lo que necesitaban con los resultados estas preguntas fueron respondidas de manera positiva.

Lo que le queda a la empresa es empezar a acoplar el proyecto a las tareas que necesiten cumplir a la hora de generar los reportes, además de agregarle las actualizaciones que ellos crean que sean necesarias para que esta se siga adaptando a las necesidades.

## 19. BIBLIOGRAFÍA

Archivo General de la Nación / Tablas de Retención Documental - TRD. (2021). Archivo General. <https://www.archivogeneral.gov.co/transparencia/gestion-informacion-publica/Tablas-de-Retencion-Documental-TRD>

A. (2021a, septiembre 8). *Metodología Scrum: Qué es y cómo funciona scrum*. Adri Melús | Blog de informática. <https://adrimelus.com/blog/metodologia-scrum-que-es-y-como-funciona-scrum/>

Base de datos definida. (2021). Oracle.  
<https://www.oracle.com/co/database/what-is-database/>

BBVA. (2021, 6 octubre). *Metodología «scrum»: ¿Qué es un «sprint»?* BBVA NOTICIAS. <https://www.bbva.com/es/metodologia-scrum-que-es-un-sprint/>

Bierman, G. (2016, 28 julio). *Understanding TypeScript*. SpringerLink. [https://link.springer.com/chapter/10.1007/978-3-662-44202-9\\_11?error=cookies\\_not\\_supported&code=49e855ee-2753-48b6-aeafd1d0ea4e18f8](https://link.springer.com/chapter/10.1007/978-3-662-44202-9_11?error=cookies_not_supported&code=49e855ee-2753-48b6-aeafd1d0ea4e18f8)

Cañete, I. (2019, 11 marzo). *Qué es un 'product backlog' y cuál es su función*. BBVA NOTICIAS. <https://www.bbva.com/es/que-es-un-product-backlog-y-cual-es-su-funcion/>

*Check out this amazing template to make your presentations look awesome at.* (s. f.). Slideteam.

<https://www.slideteam.net/process-of-document-management-lifecycle.html>

Correa, J. A., Vásquez, L., Soto, Y. M., & Hernández, M. C. (2016). *Reportes integrados y generación de valor en empresas colombianas incluidas en el Índice de Sostenibilidad Dow Jones*.

COLOMBIA. Código de Procedimiento Civil. Artículo 252. – Distintas clases de documentos.

Control de acceso HTTP (CORS) - HTTP | MDN. (2021, 14 noviembre). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTTP/CORS>

Correa, J. A., Vásquez, L., Soto, Y. M., & Hernández, M. C. (2016). *Reportes integrados y generación de valor en empresas colombianas incluidas en el Índice de Sostenibilidad Dow Jones*.

Dante, G. P. (2003, 1 diciembre). *gestión documental, de información y del conocimiento . . . puntos de contacto y diferencias - Document - Gale OneFile: Informe Académico*.

Gale OneFile.  
<https://go.gale.com/ps/i.do?id=GALE%7CA140489355&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=08644659&p=IFME&sw=w&userGroupName=anon%7Ef7a61022>

Delgado, S. (2017, 29 septiembre). *Cómo funciona la normativa ISO en la Gestión Documental. STDD Gestión Documental*. <https://sisdoc.es/como-funciona-la-normativa-iso-en-la-gestion-documental/>

Elsden, C. (2017, junio). *Designing Documentary Informatics*. <https://dl-acm-org.loginbiblio.poligran.edu.co/doi/10.1145/3064663.3064714>

Enrique, E., Condor Tinoco, C., & Soria, I. (2015). *Programación Web con CSS, JavaScript, PHP y AJAX (1.a ed., Vol. 1)*. Universidad Nacional José María Arguedas.

Goins, A. (2021, 9 noviembre). *Front End vs. Back End: ¿What's the Difference?* Kenzie Academy. <https://www.kenzie.academy/blog/front-end-vs-back-end-whats-the-difference/>



Guerrero, D. (2016). *ALCANCE DEL PROYECTO*. Universidad de Piura.  
<https://pirhua.udep.edu.pe/bitstream/handle/11042/3092/6. Alcance. pdf?sequen>

*Introduction to Agile Software Development Methods*. (2020, 1 diciembre). Inflectra.  
<https://www.inflectra.com/ideas/whitepaper/introduction%20to%20agile%20development%20methods.aspx>

K. (2021). *An Introduciton to CLIENT SERVER COMPUTING [Paperback]*.  
 New AgeInternational Publisher.

*Manejo Técnico de Información*. (s. f.). *Manejo Técnico de Información*.  
<https://thomasmti.com/nosotros/>. Recuperado 15de septiembre de 2021, de  
<https://thomasmti.com/nosotros/>

Marketing. (2021, 13 enero). *La importancia de los reportes gerenciales para la toma de decisiones – INFOSIS – Blog*. Infosis.  
<https://www.infosis.com.ar/blog/202101/la-importancia-de-los-reportes-gerenciales-para-la-toma-de-decisiones/>

(2020b, diciembre 3). *Meet the Rest APIs Tutorial JSON Family And How to make HTTP POST, GET, PUT and DELETE requests with URLSession using iOS Swift 5.2*.  
 appcodezip. <https://www.appcodezip.com/2020/10/rest-apis-json-family.html>

Mexicanos, A. (2021). *Gestión Documental En Las Organizaciones*.  
 Uoc(Universitat Oberta DeCatalunya).

N. (2018). *Acerca. Node.js*. <https://nodejs.org/es/about/>

Navarro, M. E. (2017, 4 septiembre). *Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles*. SEDICI.  
<http://sedici.unlp.edu.ar/handle/10915/62077>

- Otoom, A. H. (2019, agosto). *Automated Classification of Software Bug Reports*. <https://dl-acm-org.loginbiblio.poligran.edu.co/>. <https://doi-org.loginbiblio.poligran.edu.co/10.1145/3357419.3357424>
- Pantoja, B. E. (s. f.). *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. Scielo. [http://www.scielo.org.bo/scielo.php?pid=S1683-07892004000100005&script=sci\\_arttext](http://www.scielo.org.bo/scielo.php?pid=S1683-07892004000100005&script=sci_arttext)
- Pérez, J. E. (2019). *Introducción a JavaScript*.
- Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., & Vrgoč, D. (2016, April). Foundations of JSON schema. In *Proceedings of the 25th International Conference on World Wide Web* (pp. 263-273).
- Potions, C. (2019, 25 noviembre). *Tutorial Angular Material - Aprende a usarlo rápido*. Coding Potions. <https://codingpotions.com/angular-material>
- Prescott, P. (2015). *HTML 5: Discover How To Create HTML 5 Web Pages With Ease: Discover How To Create HTML 5 Web Pages With Ease: 1. Createspace Independent Publishing Platform*.
- S., R. K. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)) (English Edition)* (1.<sup>a</sup> ed.). Addison-Wesley Professional.
- Song, I. (2009). *Entity Relationship Model*. SpringerLink. [https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9\\_148?error=cookies\\_not\\_supported&code=31659176-b315-4954-be53-c5421aa9d136](https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_148?error=cookies_not_supported&code=31659176-b315-4954-be53-c5421aa9d136)
- W. (2019, 14 enero). *¿Qué se conoce como Archivo, en la Gestión Documental? Ats Gestion Documental*. <https://atsgestion.net/archivo-gestion-documental/>

*What is a report? - OWLL - Massey University. (2019). Massey University.*  
<https://owll.massey.ac.nz/assignment-types/what-is-a-report.php#:~:text=A%20report%20is%20a%20specific,findings%20from%20a%20research%20investigation>.

*Zapata, B. D. (2021, 8 julio). Repositorio Institucional Universidad de Antioquia: Características del reporte corporativo en Latinoamérica: un análisis sectorial. Repositorio Institucional Universidad de Antioquia.*  
<http://bibliotecadigital.udea.edu.co/handle/10495/20706>