**UNIVERSIDAD TECNOLÓGICA**

**EMILIANO ZAPATA DEL ESTADO DE MORELOS**

**DIVISIÓN ACADÉMICA DE TECNOLOGÍAS DE LA INFORMACIÓN Y DISEÑO**

**MODULO EVALUACIÓN DOCENTE PARA EL SISTEMA INTEGRAL DE SERVICIOS ACADÉMICOS DE LA UNIVERSIDAD DE CIENCIAS JURÍDICAS DE MORELOS S.C**

**REPORTE DE ESTADÍA**

**QUE PARA OBTENER EL GRADO DE:**

**NOMBRE DEL PROGRAMA EDUCATIVO**

**ÁREA DEL PROGRAMA (SI APLICA)**

|  |  |  |
| --- | --- | --- |
| **PRESENTA:**  **NOE MARTINEZ FLORES** | | |
| ASESORA EMPRESARIAL |  | ASESOR UNIVERSITARIO |
| **DRA. MARTHA FABIOLA WENCES DÍAZ** |  | **I.T.I ERICK MIRELES MERCHANT** |

**EMILIANO ZAPATA, MOR., ABRIL DE 2023**

|  |
| --- |
| **CONTENIDO** |

Índice de figuras

Índice de tablas

Agradecimientos

Resumen

Summary

[CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO 8](#_Toc103865162)

[1.1 Datos generales de la empresa 8](#_Toc103865163)

[1.2 Antecedentes del proyecto 9](#_Toc103865164)

[1.3 Objetivo general 9](#_Toc103865165)

[1.4 Objetivos específicos 9](#_Toc103865166)

[1.5 Justificación 10](#_Toc103865167)

[1.6 Alcances 10](#_Toc103865168)

[1.7 Restricciones 11](#_Toc103865169)

[CAPÍTULO 2. MARCO DE REFERENCIA 12](#_Toc103865170)

[2.1 Conceptos, metodologías y herramientas 12](#_Toc103865171)

[2.2 Propuesta de solución 19](#_Toc103865172)

[CAPÍTULO 3. DESARROLLO 21](#_Toc103865173)

[3.1 Inicio 21](#_Toc103865174)

[3.2 Planeación 22](#_Toc103865175)

[3.3 Ejecución 23](#_Toc103865176)

[3.4 Control 26](#_Toc103865177)

[3.5 Cierre 26](#_Toc103865178)

[CAPÍTULO 4. CONCLUSIONES 28](#_Toc103865179)

[4.1 Cumplimiento de objetivos 28](#_Toc103865180)

[4.2 Resultados 28](#_Toc103865181)

[4.3 Contribuciones 28](#_Toc103865182)

**REFERENCIAS**

**ANEXOS**

|  |
| --- |
| **ÍNDICE DE FIGURAS** |

[*Figura 1.1 Logotipo de la empresa* 8](#_Toc161666791)

[*Figura 3.1 Logotipo de la empresa* 24](#_Toc161666792)

[*Figura 3.2 Logotipo de la empresa* 26](#_Toc161666793)

|  |
| --- |
| **ÍNDICE DE TABLAS** |

[Tabla 3.1 Involucrados y sus funciones 21](#_Toc103865184)

|  |
| --- |
| **AGRADECIMIENTOS** |

Texto libre escrito por el estudiante.

|  |
| --- |
| **RESUMEN** |

El texto del resumen debe redactarse para captar la atención del lector y describir brevemente lo que encontrará dentro del documento. Debe responder a las preguntas:

¿qué se hizo?, ¿cómo se hizo? y ¿qué impacto tuvo en la empresa?

De igual forma, debe incluir la estructura del documento descrita por capítulos.

|  |
| --- |
| **SUMMARY** |

Debe evitarse la traducción directa realizada por una herramienta tecnológica. Por ejemplo: Google Translator. Solicitar al alumno que sea revisado por un profesor de inglés en los horarios de asesoría.

|  |
| --- |
| DESCRIPCIÓN DEL PROYECTO |

## Datos generales de la empresa

El centro de desarrollo de software es un centro de entrenamiento para personas y/o estudiantes de nivel superior enfocados en la carrera de Tecnologías de la Información y Comunicación, donde los participantes no solo adquieran conocimientos teóricos, sino que también obtengan una valiosa experiencia práctica en proyectos reales centrados en el desarrollo de software.



*Figura 1.1 Logotipo de la empresa*

CDS se encuentra dentro de las instalaciones de la UTEZ ubicada en avenida Universidad Tecnológica 1,Palo Escrito, Emiliano Zapata Morelos, C.P 62765 frente a la preparatoria CECyTE, cuenta aproximadamente con 500 empleados los cuales son distribuidos en ciertas áreas o carreras que son distribuidas a lo largo de toda la universidad.  
   
Este proyecto se realizará bajo la tutela de Erick Mireles Merchant, docente de   
la Universidad Tecnológica de Emiliano Zapata (UTEZ) que actualmente desempeña   
el cargo de docente y del proceso de Gestión de Proyectos dentro del CDS-UTEZ.

## Antecedentes del proyecto

Con el constante crecimiento de la Universidad de Ciencias Jurídicas, la gestión de estudiantes, docentes y procesos administrativos ha generado una demanda cada vez más compleja y robusta. El sistema existente, aunque funcional en ciertos aspectos, ha evidenciado limitaciones que no cumplen adecuadamente con las necesidades actuales de la institución.

Se ha decidido buscar el apoyo del Centro de Desarrollo de Software (CDS) para llevar a cabo la creación de una aplicación híbrida que aborde de manera específica y efectiva los desafíos presentes en la gestión académica y administrativa. Aunque actualmente existe un sistema en funcionamiento, su capacidad se ha quedado corta y no logra satisfacer completamente las demandas y expectativas de la universidad.

## Objetivo general

Administrar el proceso de evaluación docente en la Universidad de Ciencias Jurídicas a través de una aplicación híbrida, con el propósito de proporcionar a todos los docentes retroalimentación directa de los alumnos, permitiéndoles identificar y mejorar sus puntos de mejora o fortalezas en el ámbito educativo.

## Objetivos específicos

1. Diseñar los modelos necesarios dentro de la base de datos para almacenar la información de la evaluación.
2. Diseñar e implementar los servicios API para consulta, registro y cambio de estado de aquellos datos relacionados a la evaluación docente.
3. Implementar los privilegios que tendrán los usuarios para poder realizar los procesos que conllevan a la creación y modificación de la evaluación docente.
4. Diseñar una interfaz intuitiva y amigable para el usuario final, garantizando la accesibilidad y facilidad de uso para los estudiantes y los encargados de hacer la evaluación docente.
5. Implementar las funcionalidades para la gestión de secciones, preguntas y consulta de las respuestas de la evaluación docente.
6. Implementar pruebas funcionales y probar la aplicación en un servidor.

## Justificación

El desarrollo e implementación de un Módulo de Evaluación Docente en la Universidad de Ciencias Jurídicas es esencial para optimizar el proceso de evaluación del desempeño docente. Este módulo permitirá a los estudiantes expresar sus opiniones y evaluaciones de manera estructurada, recopilando datos significativos y proporcionando retroalimentación detallada.

La introducción de este sistema contribuirá a la transparencia y rendición de cuentas, promoviendo una cultura de mejora continua en la universidad. Además, alineado con las mejores prácticas educativas, el Módulo de Evaluación Docente no solo cumplirá con estándares de calidad, sino que también fortalecerá la reputación de la institución al demostrar su compromiso con la excelencia académica y la satisfacción estudiantil.

## Alcances

* Generación del manual de usuario para las respectivas funcionalidades de: secciones, preguntas, detalles de la evaluación docente, evaluación docente.
* Pruebas al sistema.
* Redacción del DFR.
* Generación de un plan de pruebas.
* Generación de análisis y diseño del módulo.
* El desarrollo del proyecto no incluirá el desarrollo de una aplicación móvil.

## Restricciones

* El proyecto debe cumplir con reglas de usabilidad para facilitar la experiencia del usuario final.
* Las reuniones son echas de manera remota, por video llamada por medio de plataformas como Google Meet, esto conlleva a ideas no completas o casos donde la información de un receptor a otro no sea lo suficientemente clara para el avance del proyecto.

|  |
| --- |
| MARCO DE REFERENCIA |

## Conceptos, metodologías y herramientas

Para comprender el contexto dentro del proyecto, es necesario conocer los conceptos acerca de las tecnologías, lenguajes, frameworks, entornos de trabajo, entre otros.

### Lenguajes de Programación

**Java**

El lenguaje de programación de Java fue seleccionado gracias a su alta madurez en el desarrollo de software, cuenta con una gran escalabilidad para proyectos medianos y grandes, cuenta con variables con un alto tipado estricto, lo cual lo hace ideal para el manejo de datos por parte del backend en el desarrollo de este proyecto. Además, fomenta el uso de buenas prácticas, así como de la arquitectura del proyecto.

**JavaScript**

Es un lenguaje altamente compatible para utilizar dentro de los navegadores web de hoy en día, además de ello que cuenta con una extensa cantidad de framewroks, librerías creadas por la comunidad de desarrolladores, soporte anual gracias a los módulos del ECMAScript. Junto a que, gracias a la modificación y completo control del DOM (Docuemnt Object Model) se puede crear experiencias interactivas entre el usuario dinámicas y completamente personalizables.

**TypeScript**

TypeScript, como un superconjunto de JavaScript, aporta una valiosa capa de tipado estático y funciones avanzadas al lenguaje. La introducción del tipado estático proporciona claridad y confianza al código, ya que los tipos de datos pueden ser definidos y verificados durante la fase de compilación, eliminando potenciales errores de tipo en tiempo de ejecución. Esta característica no solo facilita la comprensión del código, sino que también mejora significativamente la seguridad y robustez del desarrollo. El tipado estático de TypeScript no solo contribuye a la claridad, sino que también reduce la mutación de variables, fomentando buenas prácticas de programación y facilitando el mantenimiento del código a lo largo del tiempo.

**SQL**

El lenguaje SQL, Structured Query Language, es una elección sólida para proyectos de cualquier escala debido a su capacidad integral para gestionar datos en bases de datos. Este lenguaje permite realizar operaciones fundamentales como el almacenamiento, consulta, modificación y eliminación de datos. Ademas de ello, el uso de tablas garantiza una estructura organizada permitiendo presentar la información de manera clara y lógica. La capacidad de establecer relaciones entre tablas gracias a las llaves foráneas facilita el uso de consultas o instrucciones complejas.

### Herramientas de prototipado

Parte de los prototipos son una pieza fundamental para llevar a cabo dicho proceso,

ya que con esto se tiene una mejor visión de los conceptos principales y básicos del

proyecto, con esto se puede compartir las opiniones con el equipo y ver cuál opción

es la más idónea.

Según (Galina, 2021) aparte fabricar un prototipo puede aportar varios beneficios

como:

* Evitar hacer una gran inversión al principio en un producto que no termina de gustar a los usuarios.
* Reduce el riesgo, ya que, si alguna función del producto no guste, se deja de implementarla antes de la fabricación.
* Se puede mejorar el producto según la aceptación que ha tenido entre los clientes.

**Figma**

Según (Blandino, 2023) Figma bien permite diseñar prototipos web o móvil, presentaciones, es muy útil porque facilita el trabajo, ya que es basada en navegador 14 y puedes interactuar con las personas de tu equipo ya que es una herramienta colaborativa.

### Frameworks utilizados para frontend

**Vue JS**

Según (Jesus, 2022) permite una interfaz de usuario más reactiva y receptiva. Además de ello, El objetivo principal de Vue JS es simplificar el desarrollo web y los sitios web. Gracias a su pequeño tamaño de compilación, su facilidad de uso, aprendizaje, y rendimiento.

**Vue Boostrap**

Vue Bootstrap combina la potencia del framework de desarrollo frontend Vue.js con la versatilidad y estilos predefinidos de Bootstrap. Vue JS proporciona un enfoque reactivo para las interfaces de usuario, mientras que Bootstrap ofrece un conjunto de estilos y componentes que agilizan el diseño y la apariencia de las aplicaciones.

### Framework utilizados para backend

**Spring Boot**

Según (Fernández, 2022) Estas son las características más destacadas de este framework:

* Permite crear todo tipo de aplicaciones en el lado del back-end de forma independiente.
* Facilita el trabajo con otras herramientas como Tomcat, Jetty o Undertow. Lo hace directamente, sin necesidad de implementar archivos específicos para ello.
* Simplifica las dependencias para mejorar la configuración final del proyecto que se desarrolla con Spring Boot.
* Se trata de un framework que se configura de manera simple y es compatible con bibliotecas de terceros.
* Facilita la creación de listas, controles de estado y mejora la configuración externa para el desarrollo de aplicaciones.
* No es necesario generar código para los aspectos que controla Spring Boot y no hay requisitos para la configuración XML.

Gracias a estas características, Spring Boot facilita la creación de todo tipo de aplicaciones basadas en él de manera independiente con el mínimo esfuerzo por parte de los desarrolladores.

### Herramientas para la gestion de base de datos

**PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) que destaca por su robustez, flexibilidad y capacidad para manejar cargas de trabajo variadas.

Segun (Borges, 2015) estas son algunas de sus mas grandes caracteristicas:

* Multiplataforma: La versatilidad de PostgreSQL es evidente en su capacidad para funcionar en diversas plataformas, incluyendo sistemas operativos Unix y Windows. Esta característica elimina las preocupaciones relacionadas con la compatibilidad y permite su uso tanto localmente como a través de conexiones remotas, ofreciendo flexibilidad para adaptarse a diferentes entornos.
* Alto Volumen y Rendimiento: PostgreSQL destaca en el manejo de bases de datos con alto volumen de información gracias a su eficiente método de Control de Concurrencias Multiversión (MVCC). Este enfoque permite lecturas y escrituras simultáneas sin bloquear operaciones, mejorando significativamente el rendimiento en entornos con movimientos intensivos en la base de datos.
* Facilidad de Manejo: El principal administrador de bases de datos de PostgreSQL, PgAdmin, ofrece una interfaz sencilla y fácil de manejar. Ideal para aquellos que están comenzando en el mundo de las bases de datos o necesitan una solución para sus proyectos, PgAdmin simplifica tareas complejas mediante simples clics. Además, proporciona opciones avanzadas para usuarios experimentados que prefieren gestionar códigos específicos de manera directa.

**DataGrid**

### Herramientas de desarrollo

**VSCode**

VSCode cuenta con una interfaz intuitiva y con una capacidad ampliamente compatible para aceptar y ejcutar varios lenguajes de programación, lo que lo convierte en una herramienta valiosa para desarrolladores de diferentes tecnologías. Además, su amplio conjunto de extensiones permite personalizar el entorno de desarrollo según las necesidades específicas del proyecto y del desarrollador.

La integración de Git directamente en el IDE facilita el control de versiones y la colaboración en equipo. También ofrece características como IntelliSense, depuración integrada, resaltado de sintaxis y una poderosa capacidad de búsqueda, lo que mejora la productividad y la eficiencia durante el desarrollo.

**Intellij**

Segun (Trucoteam, 2024) Es un IDE que ofrece una amplia variedad de ventajas, entre ellas se pueden econtrar:

Mejora de la productividad: Ofrece un conjunto de herramientas que te ayudarán a optimizar tu tiempo al escribir código y revisar errores, reduciendo las horas de trabajo necesarias para programar una aplicación.

* Entorno de trabajo flexible: IntelliJ IDEA se puede personalizar fácilmente para adaptarse a tus necesidades y a la arquitectura de tu aplicación. Además, puedes cambiar entre una amplia variedad de lenguajes de manera rápida y sencilla, permitiéndote aprovechar las características de diferentes lenguajes sin problemas.
* Depuración intuitiva: El IDE viene con una amplia variedad de herramientas para ayudarte a encontrar y corregir errores rápidamente. Es fácil realizar pruebas de rendimiento profundo con IntelliJ IDEA, y puedes hacer uso del potente analizador de líneas de código para rastrear problemas.
* Soporte para múltiples plataformas: IntelliJ IDEA es compatible con Java, Kotlin, Groovy y otros lenguajes de programación más recientes, y también ofrece una solución óptima para plataformas como Android, Spring, Maven y Gradle.
* Más de 100 integraciones disponibles: El IDE te proporciona una amplia gama de integraciones y contenido adicional para ayudarte a realizar tus tareas de manera más rápida. Estas herramientas incluyen una construcción de prueba JUnit, actualización del depurador, herramienta analizadora de reglas de codificación, control de versiones y mucho más.

**GIT**

Git es un sistema de control de versiones distribuido, forma parte de los equipos de desarrollo donde gestionan y colaboran en el código fuente. Git no solo es usado únicamente por desarrolladores o creadores de código, también es utilizado por profesionales de distintos enfoques como Diseñadores digitales ya que su uso no es únicamente para archivos de programación, si no que involucra cualquier tipo de archivo electrónico.

Permite un seguimiento preciso, facilita la creación y fusión de ramas, optimiza la colaboración entre los integrantes de un equipo o de manera individual, y garantiza un historial detallado de los cambios realizados a los archivos afectados o modificados.

### Arquitectura

**Clean Arquitecture**

### Metodologías empleadas

**SCRUM**

Según (Schwaber & Sutherland, 2020) Scrum involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir tales habilidades según sea necesario.

El equipo de Scrum se compromete a lograr sus objetivos y apoyarse mutuamente. Su enfoque principal es el trabajo del Sprint para hacer el mejor progreso posible hacia estos objetivos. El equipo de Scrum y sus partes interesadas están abiertos sobre el trabajo y los desafíos. Los miembros del equipo de Scrum se respetan mutuamente para ser personas capaces e independientes, y son respetados como tales por las personas con las que trabajan. Los miembros del equipo de Scrum tienen el valor de hacer lo correcto y de trabajar en problemas complejos.

### Aplicación de colaboración empresarial

Al tener una mejor comunicación con los miembros del equipo mejora los procesos creativos dentro de la empresa en la que se elabora, por eso es eficaz tener un canal con el cuál compartir información interna con el equipo de trabajo.

Una de las aplicaciones para llevar una mejor comunicación es la siguiente:

**Slack**

Esta es una plataforma que permite interactuar con los miembros del equipo, así como puedes conectar las herramientas que se llega a ocupar dentro del desarrollo, así como sus servicios.

Según (Martín, 2023) Slack cuenta con las siguientes características.

* Permite tener conversaciones organizadas (enviar mensajes y compartir archivos, crear canales para equipos, oficinas, proyectos, etcétera).
* Puedes buscar mensajes en el historial de conversaciones del equipo de Slack.
* Añadir aplicaciones al espacio de trabajo para conectarlas. De esta forma, te mantendrás organizado y productivo.
* Realizar llamadas y vídeo llamadas con cualquier otro miembro del espacio de trabajo, así como compartir la pantalla si necesitas mostrarle algo.

## Propuesta de solución

Con base en la investigación anterior para la realización de la aplicación web, se requirió de la tecnología de Vue para el desarrollo del frontend, con ayuda del framework de Vue Bootstrap, esto con el fin de tener una mejor interacción con el usuario, por medio de sus interfaces gráficas que ofrecen estos frameworks. Así como se optó por PostgreSQL y así poder acceder a la información almacenada, además de utilizar Spring Boot como apoyo para el backend con ayuda de IntelliJ IDEA para su desarrollo.

Se determinó usar estas herramientas por lo fáciles que son, como lo es con spring para separar los microservicios y que sea más fácil reutilizarlos. Así mismo de lo rápido que es en ejecutar y el poder conectarse a una base de datos. De igual forma con vue, ya que se ahorra un poco más de tiempo y es mucho más fácil de usar con las directivas que tiene, esto lo que hace es extender el HTML, creando componentes, clases y atributos con funcionalidades ya dadas.

Los servicios que ofrecen las distintas tecnologías con tal de generar un mejor producto, con un mejor rendimiento y más ligero a fin de prevenir conflictos y 26 aprovechar las distintas funciones que ofrece cada una de las mismas, las cuales fueron cruciales para elegir el conjunto de programas mencionados.

Al ser un proyecto nuevo, se obtendrá una aplicación eficiente y fácil de utilizar para el usuario, este proyecto contará con nuevas tecnologías, unas conocidas, otras serán necesario la búsqueda de información para implementarlo correctamente, misma que será por los medios de la documentación oficial que ofrecen estas tecnologías.

|  |
| --- |
| DESARROLLO |

## Inicio

Con el fin de obtener un poco más del conocimiento acerca del funcionamiento y problemática acerca de este módulo, se requirió llevar a cabo una reunión con el responsable de proyecto, con el fin de identificar satisfactoriamente las funciones y el objetivo exacto que se necesita para poder implementar adecuadamente el desarrollo de evaluación docente dentro del sistema ya existente.

En dicha reunión, donde estuvieron presentes el cliente, responsable de desarrollo, administrador de proyecto y el equipo de desarrollo, se llegó al objetivo de que la universidad requiere la funcionalidad de evaluación para que todos los estudiantados puedan ofrecer una completa retroalimentación de sus docentes y opiniones acerca de ellos, junto a que la universidad tendrá el poder consultar aquellas respuestas en cualquier momento. Los diferentes involucrados del proyecto se encuentran definidos como se observa en la Tabla 3.1

Tabla 3.1 Involucrados y sus funciones

|  |  |  |
| --- | --- | --- |
| **INVOLUCRADOS EN EL PROYECTO** | | |
| **Nombre de la persona** | **Función** | |
| José Christian Narváez Figueroa | | Sponsor  *Interno* |
| Mónica Sotelo Rosales | | Cliente  *Externo* |
| Erick Mireles Merchant | | Asesor  *Interno* |
| Erick Mireles Merchant | | Administrador del proyecto  *Interno* |

## Planeación

Para realizar la planeación correspondiente al proyecto y que este tuviera un control de las actividades a realizar en el desarrollo, se requirió elaborar la documentación necesaria donde se tomaron en cuenta las fechas para la entrega y el módulo a cumplir.

### Diagrama de Gantt

Uno de los documentos para planificar el desarrollo es el diagrama de Gantt como se muestra completo en ANEXO A. DIAGRAMA DE GANTT, el cual relata las actividades que se deben hacer en cada una de las semanas para llevar un buen control en el desarrollo y no presentar retrasos en ello. Este abarca las 15 semanas que dura el cuatrimestre en el cual se tuvo el tiempo para completar el proyecto.

### Documento Formal de Requerimientos

Para lograr comprender e identificar los requerimientos funcionales y no funcionales del módulo de evaluación docente, se llevó a cabo la realización de un análisis de la información obtenida en la reunión con el cliente, dónde se pudo identificar los puntos que permitieran la elaboración del respectivo documento formal de requerimientos como se muestra en el ANEXO B. DOCUMENTO FORMAL DE REQUISITOS.

El documento Formal de Requerimientos define todas las necesidades del cliente en manera de funcionalidades, entre ellos se encuentra. Secciones, Preguntas, Consulta de Respuestas, Evaluar docente por Alumno. En cada una de las funcionalidades se define los datos de entrada, datos de salida (si este aplica), criterios de aceptación, reglas de negocio, requerimientos funcionales y Notas para aclarar posibles dudas o inquietudes por parte de los involucrados.

Esto permite que tanto el desarrollador como el cliente tengan los detalles claros con respecto a lo que podrá hacer la aplicación una vez que se llegue a su etapa de entrega.

## Ejecución

Una vez finalizado el Documento Formal de Requerimientos, se llevaron a cabo varios procesos conforme al marco DMS que regula las operaciones del Centro de Desarrollo de Software. Estos procesos incluyeron la revisión y validación de los requisitos por parte de los interesados, así como la planificación detallada de las etapas de desarrollo y las actividades de prueba. Además, se establecieron los roles y responsabilidades de los miembros del equipo de desarrollo, asegurando una clara comprensión de las tareas asignadas y los objetivos del proyecto.

### Diagramas de Casos de Uso

El diagrama fue elaborado con el propósito de establecer un modelo de representación que abarque los distintos requerimientos del módulo en cuestión. Este modelo consta de la identificación de los diversos roles que pueden interactuar con la aplicación, así como de las funciones asociadas a cada uno de estos roles, incluyendo las que se desglosan de estas funciones principales.

Inicialmente, se creó un diagrama general que ilustra estos casos de uso, tal como se muestra en la Figura 3.1 Diagrama de casos de uso. Posteriormente, se detallaron estos casos de uso en el Anexo C, donde se presentan los Diagramas de Caso de Uso de manera más específica.

El propósito fundamental de este enfoque es proporcionar una comprensión clara y completa de las interacciones que pueden ocurrir dentro del sistema, tanto desde la perspectiva de los usuarios como de las funcionalidades que estos pueden llevar a cabo. Esto facilita el diseño y desarrollo del módulo, al establecer una base sólida para la implementación de las diversas características y su integración en la aplicación final.

A diagram of a process

Description automatically generated

*Figura 3.1 Diagrama de casos de uso*

### Plan de Reporte de Pruebas

En este documento se comienza a estructurar el correcto funcionamiento deseado para la aplicación. Se abordan aspectos tanto del diseño estructurado como del funcional, incluyendo la realización de pruebas pequeñas destinadas a validar el comportamiento y la lógica del sistema. Este proceso implica la identificación y descripción detallada de los diferentes comportamientos esperados de la aplicación. Estas descripciones proporcionan una visión clara de las funcionalidades y escenarios de uso previstos, lo que constituye una base sólida para el diseño y desarrollo subsiguientes. La realización de estas pruebas es esencial para asegurar que la aplicación cumpla con los requerimientos establecidos y funcione de manera coherente y efectiva una vez implementada.

**Pruebas de Sistema**

En pruebas de sistema se describe todo lo relacionado en comprobar el buen funcionamiento de la aplicación por requisito. Se trata de relatar el número del requisito antes puesto en el documento formal de requisitos, la sesión, que quiere decir, qué roles lo pueden ejecutar, el nombre del componente, una pequeña descripción de lo que trata, los prerrequisitos si llegara a aplicar, que estos son una especie de aprobación, por su contenido o por especiales razones administrativas, los datos de entrada que necesita para esta prueba, el procedimiento que se debe hacer para llegar a la respuesta ideal, la acción que hace esta prueba y el resultado esperado, que en este caso pueden ser varios resultados esperados, contando los correctos e incorrectos.

**Pruebas No Funcionales**

Dentro de estas pruebas se describen la verificación de los requisitos no funcionales basados en la operación de un software, es decir, no se basa en la funcionalidad, puede ayudar a determinar la carga que soporta el producto, o para ver si su rendimiento es el correcto con el servidor.

En esta también se describen los requerimientos de interfaz de usuario que son las expectativas que esta tiene por cumplir con las necesidades del sistema. También se describen los requerimientos de confiabilidad, los de eficiencia que esto es el acto y consecuencia de pedir o necesitar algo, los de mantenimiento, los de portabilidad, los de interoperatividad, los de reusabilidad entre otros requerimientos no funcionales.

**Pruebas de Integración**

Estas pruebas se hacen una vez que ya se aprobaron las pruebas unitarias y lo que esto hace es probar todos los elementos unitarios que componen el software, que funcionan juntos correctamente probándolos en grupo, así mismo para verificar el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen. Esto con el fin de verificar que interactúan correctamente a través de sus interfaces, tanto como las internas como externas y que cubran su funcionalidad establecida.

### Análisis y Diseño

**Diagrama de despliegue**

En este diagrama se muestra la arquitectura de cómo está el sistema, del despliegue de los artefactos, para este diagrama se separó el BackEnd y el FontEnd para tener una mejor visualización de las tecnologías que le pertenecen a cada uno, quedando así de esta manera como se muestra en la siguiente Figura 3.2 DIAGRAMA DE DESPLIEGUE

A screenshot of a computer

Description automatically generated

*Figura 3.2 Diagrama de Despliegue*

**Diagrama de componentes**

Este diagrama consiste en separar como su nombre todos los componentes que hay y adecuarlos en su módulo correspondiente, describiendo dentro de los componentes todos los requerimientos que este puede hacer, basándose siempre del documento formal de requisitos. Este mismo muestras las relaciones que tienen entre ellos, así como sus dependencias entre estos componentes, como se muestra en la siguiente Figura 3.3 Diagrama de componentes.

A screenshot of a computer screen

Description automatically generated

*Figura 3.3 Diagrama de Componentes*

**Diagrama de Clases**

Este diagrama fue creado de acuerdo al diagrama de componentes, donde, del sisitema existente se han agregado nuevos elementos para llevar acabo la evaluación docente, anexando en ellos todos los atributos que se describieron en el documento formal de requisitos para llevar a cabo su registro, de igual forma llevando una relación entre ellos para tener mejor visualización entre las clases al momento de crearse. A continuación, se muestra la ANEXO D. DIAGRAMA DE CLASES.

**Diagrama de Actividades**

Este es un diagrama de flujo que ayuda a representar de manera gráfica el algoritmo o proceso de cada uno de los requerimientos con el fin de cuando translade a códifo se tenga la idea de lo que hará si eligen las distintas opciones que se presentan.

Básicamente se modela el comportamiento que tiene el sistema, mostrando así una secuencia de acciones, como se muestra en el ANEXO E. DIAGRAMA DE ACTIVIDADES.

**Diagrama de Secuencia**

Este diagrama se desarrolló con el fin de tener un acercamiento a como se iban a llamar las clases y los métodos para las funcionalidades, lo que se pedirá de valor inicial en cada uno de ellos, lo que va a devolver cada una de las peticiones solicitadas, todo esto para tener claro la interacción entre los objetos de un sistema, esto se hace por cada caso de uso, para determinar qué objetos son necesarios para la implementación del escenario, como se muestra en en el ANEXO E. DIAGRAMA DE SECUENCIA.

**Arquitectura del Producto**

Este se hace con el fin de saber cómo es que van a interactuar los distintos componentes una vez estructurados en microservicios, se llegó a hacer algo parecido al diagrama de despliegue, pero en este estructurando ya los servicios, controladores y los repositorios, así como el JPA que es la API de persistencia desarrollada y el JWT que es un JSON Web Token que es para la creación de tokens. La arquitectura del producto quedó de la siguiente forma como se muestra en la Figura 3.4 Arquitectura del Proyecto

A diagram of a computer network

Description automatically generated with medium confidence

*Figura 3.4 Arquitectura del Proyecto*

**Árbol de Navegabilidad**

Este de desarrolló para tener un poco más claro la navegabilidad que la aplicación que se tendría durante su desarrollo y tener en cuenta con cuales iban a estar interactuando cada una de las vistas, como se muestra en la siguiente Anexo F. Árbol de Navegabilidad

**Mockups**

Dada a las necesidades escritas dentro del Documento Formal de Requerimientos (DFR) se hicieron varias pantallas denominadas Mockups, los cuales son un bosquejo de la idea por cada una de las funcionalidades implícitamente agregadas en el DFR. La concepción de los mockups sirvió para ayudar al equipo de Frontend y al cliente en conocer mejor el flujo de los datos y el flujo que un usuario tendría que seguir para llevar a cabo cada una de las funcionalidades que cuenta todo el Módulo de evaluación Docente.

Los mockups fueron enviados y posteriormente revisados por el cliente, el cual presento su respectiva retroalimentación para poder añadir o eliminar elementos que en ese momento consideraba necesarios.

### Construcción y Codificación

Para la inicialización correcta del proyecto se necesitó clonar el repositorio del proyecto a una carpeta, dicho proceso fue realizado con el control de versiones Git. Actualmente, hay diferentes ramas las cuales contienen los cambios del repositorio remoto dentro de nuestro dispositivo de manera local. Para términos de este proyecto se creó una rama local que posteriormente será unida a la rama principal.

**Construcción del Backend**

Esta clase representa la estructura de los datos que deseas almacenar en tu base de datos. En Spring, se anota con @Entity. Contiene atributos que representan las columnas de la tabla en la base de datos.

Se continuó con la programación de los archivos Data Transfer Object (DTO). El DTO es una clase que se utiliza para transferir datos entre el frontend y el backend. Se usa para definir la estructura de datos que se envía o recibe en las solicitudes HTTP. En el DTO, se pueden aplicar validaciones como @NotBlank para asegurar que los datos sean válidos antes de ser procesados.

Una vez hecho eso, se continuó creando un repositorio, el repositorio es una interfaz y esta proporciona métodos para realizar operaciones CRUD en la base de datos, como guardar, eliminar, buscar, etc.

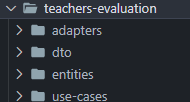
Cuando se concluyó el repositorio se empezó a hacer la clase servicio. El servicio es donde se encuentra la lógica de negocio de la aplicación. Aquí se implementan los métodos para crear, leer, actualizar y eliminar registros en la base de datos. El servicio importa el repositorio y utiliza sus métodos para interactuar con la base de datos.

Ya teniendo terminado el servicio, repositorio, DTO y el modelo, finalmente se desarrolló el controlador. El controlador es responsable de manejar las solicitudes HTTP entrantes y enviar respuestas de vuelta al cliente. Se anota con @RestController y define métodos que corresponden a diferentes endpoints de la API. El controlador utiliza el DTO para recibir datos del cliente y el servicio para procesar y manipular esos datos como se muestra en la Figura 3.

**Construcción del Frontend**

Dentro del proyecto clonado se encuentra la carpeta 094-SISE-UCJ/Front/sise-ucj/src/modules. Dentro de la carpeta de módulos es donde todos los modulos junto a sus respectivas funcionalidades que tiene el proyecto. Para este caso se creó una carpeta con el nombre teachers-evaluation para desarrollar todas las funcionalidades de la evaluación docente.

Como se puede observar en la Figura 3. Encarpetado, se muestra el contenido que la carpeta teachers-evaluation contiene, esto gracias a la metodología arquitectura limpia

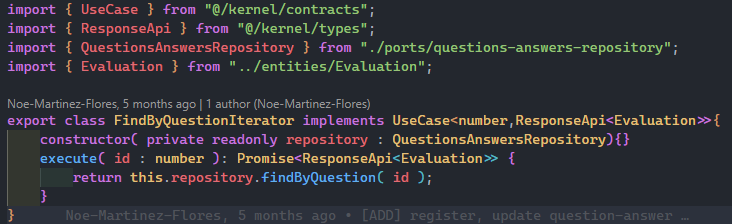


*Figura 3.4 Encarpetado*

Como primer punto a destacar se necesita crear un archivo dentro de use-cases/ports para poder definir todos aquellos métodos que posteriormente serán utilizados con los servicios provistos por él backend. Esto utilizando TypeScript, Se utiliza interfaces ya que proporciona beneficios en términos de abstracción, desacoplamiento, flexibilidad, pruebas unitarias, legibilidad y tipado estático, lo que contribuye a la creación de sistemas más robustos y mantenibles. Esto se puede ver en el Anexo G Repositorio

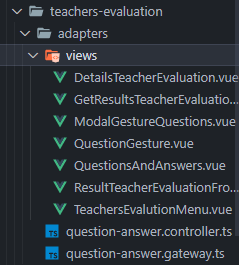
Posteriormente se creó un archivo por cada interfaz definida, dentro del cual se le conocerá como un iterador. Para esto, se necesita declarar una nueva Clase en TypeScript el cual implementará una interfaz previamente definida en el proyecto conocido como UseCase, el cual recibe dos tipos de parámetros, estos parámetros son esenciales para el correcto funcionamiento de esta implementación. El primer parámetro funciona como dato de entrada, y el segundo parámetro recibido es lo que espera ser regresado. Esto es completamente dependiente del tipo de dato se espera tanto por recibir como por devolver.

Posteriormente se define un constructor, los constructores son utilizados para realizar una función en específico cuando se llama una instancia de la misma clase antes creada. El cual se indica que recibirá solamente un atributo el cual solo puede ser solo lectura y usable solo en este punto del código, para ello se utilizaron las palabras reservadas de private y readonly, este atributo se debe especificar el tipo de dato que corresponde para posteriormente definir la ejecución del código, el cual recibe como parámetro un dato con el tipo de dato definido previamente en el useCase, para posteriormente indicar que regresará una promesa. Se define un tipo de promesa el cual regresará uno o varios datos, esto dependiendo del tipo de dato, haciendo referencia la segundo parámetro que recibió el UseCase. Esto se puede mostrar en la Figura 3. Iteratdor.



*Figura 3.4 Iterador*

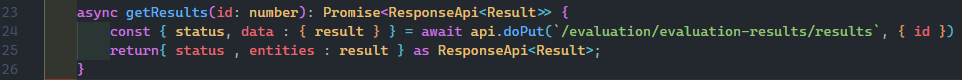
Posteriormente se dentro de la carpeta adapters se crearon dos archivos, el primero de ellos hace referencia a tener la funcionalidad de un Gateway, el segundo de un Controlador. Esto se puede mostrar en la Figura 3. Adaptadores



*Figura 3.4 Iterador*

Dentro del archivo con terminación gateway.ts, fue creada una clase que implementa todos las interfaces definidas anteriormente, estas interfaces pueden ser vistas en el Anexo G Repositorio, la implementación de estas interfaz van acorde a cada uno de los endpoints definidos por parte del backend, su estructura cuenta primeramente la definición de la palabra reservada async, el cual indica que es un evento asíncrono posteriormente, el o los parámetros necesarios para el uso del futuro endpoint, justo después se deja explicito que tipo de dato regresará la función, en todos estos casos son promesas con diferentes tipos de datos. Seguidamente se utiliza la palabra reservada await, que indica que el tiempo de ejecución del código en ese preciso momento del tiempo no será síncrono si no que podrá tomar más tiempo. Utilizando una configuración previa del llamado a los endpoints, se utiliza el método correspondiente para ejecutar la acción, justo después de ello se especifica cada uno de los endpoints junto al tipo de dato que recibe. Para recolectar los datos obtenidos y posibles errores se utiliza la desestructuración propia de JavaScript, el cual retorna un status y una data, dentro de data cuenta con otra propiedad el cual es accedido por el nombre de result.

Posteriormente se indica el retorno, está constituido por el status y el result, si son varios datos o posibles arreglos, retorna entities pero si solo es un elemento retorna entitty, si retorna una respuesta boolena entonces solo retorna result, finalmente se indica el tipo de retorno, que es fácilmente identificable por el tipo de dato que retorna toda la función en general. Todo ello es visible en la Figura 3. Gateway



*Figura 3.4 Gateway*

Una vez realizado los respectivos gateways, se comenzó de inmediato a crear la clase del archivo con terminación controller.ts, dentro de ellos se define cada uno de las interfaces creadas anteriormente, estas ahora servirán como métodos de clases para poder acceder directamente a su configuración y ejecución del código. Cada una de las funciones recibirá un tipo de dato específico, se llamara una nueva instancia de los respectivos iteradores junto a una nueva instancia del archivo con terminación gateway.ts

## Control

Esta sección debe describir las acciones realizadas para llevar el control del proyecto. Los elementos por colocar pueden variar dependiendo del proyecto. Se pueden colocar elementos como: descripción de reuniones periódicas para el seguimiento del proyecto, seguimiento del proyecto considerando la planeación de actividades en comparación con el avance real, bitácora de cambios en el proyecto o seguimiento de riesgos.

## Cierre

Esta sección debe describir las acciones realizadas para concluir el proyecto de manera formal. Debe incluir al menos las acciones realizadas para entregar el proyecto (reunión), la descripción de los entregables del proyecto al cliente y la aceptación del cliente (o carta de liberación).

|  |
| --- |
| CONCLUSIONES |

## Cumplimiento de objetivos

Se debe redactar una descripción que indique cómo se cumplieron los objetivos (generalmente los específicos) y por qué se dice que se cumplieron o por qué no se cumplieron. Para ambos casos, se debe argumentar la razón.

## Resultados

Se deben describir los entregables finales, probados y en operación, así como los beneficios obtenidos por la empresa como resultado del proyecto.

## Contribuciones

Se deben describir las aportaciones a la empresa. Generalmente, no medibles. Por ejemplo: se utilizó una metodología nueva para el desarrollo. ¿Qué se utiliza ahora en la empresa que no se usaba antes?

De igual forma, se debe describir como benefició el proyecto a la formación profesional

del alumno. NO ESCRIBIR EN PRIMERA PERSONA. Al igual que el resto del documento, se debe escribir en tercera persona.

|  |
| --- |
| **REFERENCIAS** |

Galina, P. (21 de 05 de 2021). *iebschool*. Obtenido de iebschool: https://www.iebschool.com/blog/herramientas-prototipado-analitica-usabilidad/

Blandino, G. (1 de 4 de 2023). *Pixarprinting*. Obtenido de Pixarprinting: https://www.pixartprinting.es/blog/figma-que-es/

Jesus. (11 de 10 de 2022). *Dongee*. Obtenido de https://www.dongee.com/tutoriales/que-es-vue-js-para-que-sirve/

Fernández, E. C. (25 de 10 de 2022). *tokioSchool*. Obtenido de tokioSchool: https://www.tokioschool.com/noticias/spring-boot/#:~:text=Spring%20Boot%20cuenta%20con%20una,back%2Dend%20de%20forma%20independiente.

Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum.*

Trucoteam. (2024). *trucoteca*. Obtenido de trucoteca: https://trucoteca.com/en/que-ventajas-ofrece-intellij-idea/

Borges, S. (26 de 09 de 2015). *infranetworking*. Obtenido de infranetworking: https://blog.infranetworking.com/5-razones-por-las-cuales-debes-usar-postgresql/

Martín, E. (2023). *grupocibernos*. Obtenido de grupocibernos: https://www.grupocibernos.com/blog/7-aplicaciones-de-comunicacion-y-trabajo-en-equipo

|  |
| --- |
| **ANEXOS** |

**ANEXO A. NOMBRE DEL ANEXO**

Los anexos, además de tener una letra, deberán indicar un nombre para poder identificarlos. No son títulos, por lo que no aparecerán en el índice.

**ANEXO B. NOMBRE DEL ANEXO**

**Estilos del formato de reporte**

Este archivo incluye 5 estilos definidos para títulos y texto del contenido, que puedes encontrar en la banda **Inicio** de Word, y son:

* **Título 1**: para los títulos de capítulo.
* **Título 2**: para los títulos de segundo nivel. Por ejemplo: 1.2
* **Título 3**: para los títulos de tercer nivel. Por ejemplo: 1.2.1
* **Descripción**: para los títulos de figuras y tablas.
* **Normal**: para el texto del contenido.



Todo el texto del documento debe tener aplicado al menos uno de estos estilos, según corresponda.

Se deben utilizar los estilos para que, al actualizar los índices de contenido, se adopten adecuadamente los textos colocados como títulos y se indiquen las páginas donde se encuentran.

Los títulos del índice de contenido deben ser los únicos elementos numerados.

**Portada**

Toda la portada deberá estar escrita en mayúsculas. Se debe elegir la palabra que defina correctamente a los asesores empresarial y universitario (asesor o asesora, universitario o universitaria); es decir, el texto colocado en rojo deberá sustituirse según corresponda.



Se debe colocar correctamente el nombre del programa educativo, considerando dos escenarios:

* Los nombres de programas que no tienen área (Ingenierías) se colocan en tamaño 16.
* Los nombres de programas que tienen área (TSU) deben seguir la regla anterior y, además, colocar el área en tamaño 14.

Es necesario que la portada tenga centrados tanto el nombre del proyecto como el grado. Esto implica que no solo debe estar centrado horizontalmente, sino también verticalmente en el espacio que corresponde a cada texto (área blanca para el nombre del proyecto y área azul para grado).

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

El lugar y fecha para el reporte se encuentra en la parte inferior de la portada. Se debe verificar que el mes que se indica sea un mes posterior a la finalización de la estadía.



**Formato del documento**

Se deberá respetar el formato del documento actual. Si por algún motivo se llegase a modificar, es necesario revisar los siguiente:

* Los márgenes del documento serán de 3cm. a la izquierda, 2.5 cm. a la derecha, superior e inferior.
* No se deberá incluir ninguna leyenda (ni líneas, ni títulos de trabajos, ni nombre del alumno) de encabezado.
* El pie de página solo deberá contener el número de la página fuente arial 10, alineado a la derecha.

**Cantidad de páginas**

* Las páginas del documento empiezan a contar después de la portada, que se considera la página 0, pero el número de página se hace visible hasta el Capítulo 1.
* La cantidad de páginas que deberá contener el reporte de estadía será de 40 como mínimo, las cuáles **se empiezan a contar a partir de la** **primera página del Capítulo 1** y **hasta la última página del capítulo 4**. No se consideran portada, índices, agradecimientos, resumen, referencias o anexos.
* Por ejemplo: un reporte que **inicia en la página 8** deberá **termina**r, al menos, **en la página 47**.

**Saltos de línea**

Se deberán considerar las siguientes reglas para los saltos de línea o renglones vacíos entre párrafos, títulos, imágenes y tablas:

* Se coloca un salto de línea entre título y párrafo; y entre párrafo y párrafo.
* Las imágenes y las tablas se consideran parte del texto o un párrafo. Si una imagen se encuentra con una tabla, o viceversa, se coloca un salto de línea entre ellas.
* Se colocarán dos saltos de línea entre párrafo y título siguiente.

Cuando un título queda solo al final de una página, existen dos opciones:

* Si el espacio restante es un salto de línea, se puede insertar un salto de página antes del título para moverlo a la siguiente página.
* Si el espacio restante es más de un salto de línea, se debe complementar la redacción del texto previo para evitar que el título quede solo al final de la página.

****

**Redacción y ortografía**

El documento debe estar escrito en tercera persona y evitar el uso del verbo ser seguido de un adjetivo. Por ejemplo:

* **No** **escribir**: Fue necesario, fue requerido, fue desarrollado, etc.
* **Utilizar**: Se necesitó, se requirió, se desarrolló.

Se debe evitar el uso de la palabra nosotros. Por ejemplo:

* **No escribir**: Tuvimos, realizamos, obtendremos, etc.
* **Utilizar**: Se tuvo, se realizó, se obtendrá, etc.

Es necesario revisar acentuación del documento (F7 con Word) y prestar especial atención en verbos escritos en pasado y futuro.

Se debe revisar que el texto este redactado procurando utilizar enunciados cortos. Generalmente se hace un uso excesivo de comas en lugares erróneos. Para verificar la necesidad de un punto, solicitar:

***Lee los párrafos en voz alta. Si durante la lectura te falta el aire para terminar, es posible evidencia de que requieres un punto o punto y coma.***

**Numeración y viñetas**

Se prefieren las viñetas sobre la numeración, salvo cuando sea necesario indicar un orden. **NO** se deben colocar saltos de línea entre numeración o viñetas. **Esto se debe observar a detalle, ya que se abusa de ello** con el fin de obtener hojas adicionales.

Se debe colocar un salto de línea antes de iniciar el listado con las viñetas.

**Referencias**

Las referencias se deben incluir en el texto y, en la medida de lo posible, seguir el formato APA utilizado por Microsoft Word. Para ello, se deben utilizar las herramientas para la gestión de referencias que proporciona el editor.

Se deberá evitar el plagio. Evitar el texto copiado y pegado. Si no hubiese otra opción, referenciarlo correctamente. Es preferible parafrasear o escribir con palabras propias.

**Referencia parafraseada:** comunica el mensaje del autor original con palabras propias y conservando el sentido en cuanto a organización, continuidad y coherencia. Generalmente, se inicia con una frase introductoria (así como, por tanto, de acuerdo con, según, etc.)

*De acuerdo con (Salazar, 2018), un programa computacional aplicado (PCA) se encarga de resolver un problema trivial en el campo...*

**Referencia textual:** es una cita fiel del texto original de un autor, palabra por palabra. El texto debe ponerse entre comillas, acompañados de la referencia.

*Para comprender los ejes corporales del cuerpo humano, "se han concebido planos imaginarios de referencia que atraviesan el cuerpo de forma que unos son perpendiculares a otros" (Palastanga, Field & Soames, 2000).*

**Las referencias deben tomarse de sitios confiables** y debe priorizarse la referencia de elementos obtenidos de libros, enciclopedias, artículos científicos, artículos de revistas académicas, entre otros.

Los documentos o sitios Web deben ser confiables, de entidades reconocidas como expertos o de renombre en el área.

Si se trata de una **entrada de blog**, debe tratarse de una persona reconocida en el campo (experto, no necesariamente popular). Por ejemplo: Richard Stallman, Bruno Munari, entre otros.

**Imágenes y tablas**

Si se añade una imagen es porque:

* Ayuda a la compresión del texto y debe describirse.
* El lector puede distinguir el texto y formas que incluye la imagen.
* Es necesaria y no se repite en algún otro lugar del documento.

El **tamaño de las imágenes** en el reporte quedará **a consideración del asesor universitario**, mismo que validará que sean necesarias y no se abuse de ellas para ocupar espacio.

Es válido colocar dos imágenes seguidas siempre que sea necesario para mantener el correcto formato del documento. **Se debe verificar que no se abuse de esto.**

Las imágenes y tablas NO deben colocarse en cuadros de texto o con diseño flotante (que pueda arrastrarse con el mouse sobre el documento). Las imágenes deben colocarse en línea con el texto.

Las imágenes deben estar referenciadas en el texto. Si se añade una imagen es porque ayuda a la compresión del texto y debe describirse. Por ejemplo, para hacer referencia a la figura con título Figura 2.1 en el texto descriptivo, se deberá hacer como:

“… en la Figura 2.1 se puede observar el funcionamiento paso a paso del algoritmo de búsqueda Bubble Sort para un arreglo de datos numéricos aleatorios.”

El rótulo y número, en el título de las imágenes, debe colocarse en texto normal (sin cursivas o negritas). Por ejemplo: Figura 2.1 Servidor de base de datos. El título de las tablas se coloca en la parte superior del encabezado, también en texto normal (sin cursivas o negritas).

Los títulos de imágenes o tablas se deben colocar sin punto.

Si una imagen no fue de creación propia, es necesario agregar la referencia al título de las imágenes. Por ejemplo:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

No debe existir algún salto de línea entre la imagen y su título de imagen. Deben cortarse los espacios en blanco alrededor de la imagen.



Cuando un título queda solo al final de una página, existen dos opciones:

* Si el espacio restante es un salto de línea, se puede insertar un salto de página antes del título para moverlo a la siguiente página.
* Si el espacio restante es más de un salto de línea, se debe complementar la redacción del texto previo para evitar que el título quede solo al final de la página. Por ejemplo:

