[INDICACIONES GENERALES 4](#_Toc203571618)

[Formatos y plantilla 4](#_Toc203571619)

[Estética y estilo de redacción 5](#_Toc203571620)

[Normativas y citas 6](#_Toc203571621)

[ESTRUCTURA DEL DOCUMENTO PLANTILLA PROYECTO SENA 6](#_Toc203571622)

[Resumen/abstract 7](#_Toc203571623)

[Introducción 7](#_Toc203571624)

[**1.** Planteamiento y justificación del problema 8](#_Toc203571625)

[**2.** Objetivos del Trabajo final de proyecto productivo TFPF 9](#_Toc203571626)

[2.1. Objetivo general del TFPF 10](#_Toc203571627)

[2.2. Objetivos específicos del TFPF 10](#_Toc203571628)

[**2.3.** Alcance del proyecto 10](#_Toc203571629)

[**3.** Matriz de Riesgo 11](#_Toc203571630)

[**4.** ELICITACIÓN DE REQUISITOS 11](#_Toc203571631)

[4.1. Identificación de procesos 11](#_Toc203571632)

[4.2. Recolección de información del software a construir de acuerdo con las necesidades del cliente 12](#_Toc203571633)

[4.3. Elección de la técnica de recolección de información: 12](#_Toc203571634)

[4.4. Diseño de los formatos, según la técnica o técnicas de recolección de la información seleccionada 12](#_Toc203571635)

[4.5. Aplicación de la técnica de recolección de la información 12](#_Toc203571636)

[4.6. Análisis de la información recolectada 13](#_Toc203571637)

[**5.** ESPECIFICACIÓN DE REQUERIMIENTOS, DE ACUERDO CON LA INFORMACIÓN RECOLECTADA, APLICANDO ESTANDARES DE REQUERIMIENTOS 13](#_Toc203571638)

[5.1. Requerimiento funcionales 13](#_Toc203571639)

[**5.3.** REQUERIMIENTOS NORMATIVOS 16](#_Toc203571640)

[5.4. REGLAS DEL NEGOCIO 16](#_Toc203571641)

[**5.5.** PROPUESTA TECNICA 17](#_Toc203571642)

[**5.5.1.** Cronograma 17](#_Toc203571643)

[**5.5.2.** Costos Estimados 18](#_Toc203571644)

[**5.5.3.** Requisitos de Hardware 18](#_Toc203571645)

[**5.5.4.** Requisitos de Software 18](#_Toc203571646)

[6. ANÁLISIS DE LA ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE. 19](#_Toc203571647)

[6.1. ALTERNATIVAS DE SOLUCIÓN (PROTOTIPO O PROTOTIPOS DEL SISTEMA, MOCKUPS) 19](#_Toc203571648)

[6.2. Historias de Usuario 19](#_Toc203571649)

[6.3. DIAGRAMA DE CASOS DE USO Y EXTENSIBILIDAD DE ESTOS 20](#_Toc203571650)

[6.4. DIAGRAMAS DE ACTIVIDADES 22](#_Toc203571651)

[6.5. DIAGRAMA DE SECUENCIAS 23](#_Toc203571652)

[6.6. CONSTRUIR EL MODELO DE DOMINIO DEL SISTEMA (DIAGRAMA DE CLASES). 24](#_Toc203571653)

[7. DISEÑO DE LA SOLUCIÓN DEL SOFTWARE DE ACUERDO CON LOS PROCEDIMIENTOS Y REQUISITOS TÉCNICOS 26](#_Toc203571654)

[**7.1.1.** Metodología 26](#_Toc203571655)

[7.2. ARQUITECTURA DEL SOFTWARE (DIAGRAMA DE COMPONENTES), Y PATRONES DE DISEÑO DE SOFTWARE. 26](#_Toc203571656)

[7.3. DIAGRAMA DE DESPLIEGUE. 26](#_Toc203571657)

[7.4. DISEÑO FRONT-END - INTERFACES GRÁFICAS DE USUARIO (HTML.CSS Y JS) 26](#_Toc203571658)

[7.5. INTERFACES GRÁFICAS DE USUARIO MÓVILES. 27](#_Toc203571659)

[7.6. MAPA DE NAVEGACIÓN DE LA APLICACIÓN. 27](#_Toc203571660)

[7.7. DETERMINAR TIPOS DE BASES DE DATOS. 27](#_Toc203571661)

[7.8. POLÍTICAS DE SEGURIDAD DE LOS DATOS. 27](#_Toc203571662)

[8. CONSTRUCCIÓN DEL SOFTWARE 27](#_Toc203571663)

[8.1. BASE DE DATOS PARA EL SOFTWARE A PARTIR DEL MODELO DE DATOS. 27](#_Toc203571664)

[8.1.1. Modelo Entidad Relacion MER 27](#_Toc203571665)

[8.1.2. Modelo de Datos Diagrama ER 27](#_Toc203571666)

[8.1.3. Modelo Relacional MR 27](#_Toc203571667)

[8.1.4. OBJETOS DE LA BASE DE DATOS (PROCEDIMIENTOS ALMACENADOS, VISTAS, DISPARADORES) 27](#_Toc203571668)

[PROCEDIMIENTOS ALMACENADOS 27](#_Toc203571669)

[VISTAS 28](#_Toc203571670)

[DISPARADORES 28](#_Toc203571671)

[8.1.5. DICCIONARIO DE DATOS. 28](#_Toc203571672)

[8.1.6. ESQUEMAS DE SEGURIDAD DE LOS DATOS. 28](#_Toc203571673)

[9. CODIFICACIÓN DEL SOFTWARE DE ACUERDO CON EL DISEÑO ESTABLECIDO. 29](#_Toc203571674)

[9.1. FRONT END (HTML, CSS, JAVASCRIPT, RESPONSIVE) 29](#_Toc203571675)

[9.2. ESTÁNDAR DE CODIFICACIÓN (DOCUMENTACIÓN DEL CÓDIGO, APLICABILIDAD DEL ESTANDAR DE CODIFICACIÓN). 30](#_Toc203571676)

[9.3. CÓDIGO FUENTE DE LOS MÓDULOS DEL SOFTWARE WEB Y MÓVIL. 31](#_Toc203571677)

[9.4. SERVICIOS WEB. 31](#_Toc203571678)

[9.5. CONTROL DE VERSIONES 31](#_Toc203571679)

[10. PRUEBAS DEL SOFTWARE 32](#_Toc203571680)

[10.1. PLANEACIÓN Y DISEÑO DE LA PRUEBAS A NIVEL UNITARIO (MÓDULOS), CON BASE EN LO REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES. 32](#_Toc203571681)

[10.2. EJECUCIÓN DE LAS PRUEBAS, INFORME DE HALLAZGOS 33](#_Toc203571682)

[10.3. CORRECCIÓN DEL ERRORES, Y DOCUMENTACIÓN DE LOS MISMOS. 34](#_Toc203571683)

[10.4. MANEJO DE ALERTAS / EXCEPCIONES 34](#_Toc203571684)

[10.5. PLANEACIÓN Y DISEÑO DE LA PRUEBAS A NIVEL SISTEMA, CON BASE EN LO REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES. 34](#_Toc203571685)

[10.6. EJECUCIÓN DE LAS PRUEBAS, INFORME DE HALLAZGOS 34](#_Toc203571686)

[10.7. CORRECCIÓN DE ERRORES, Y DOCUMENTACIÓN DE ESTOS. 34](#_Toc203571687)

[11. PLAN DE DESPLIEGUE (HOSTING, DOMINIO, SUBDOMINIO, COSTOS, HERRAMIENTAS, PLATAFORMA, SEGURIDAD) 35](#_Toc203571688)

[**12.** IMPLANTACIÓN DEL SOFTWARE**.** 35](#_Toc203571689)

[**12.1.** PLAN DE IMPLANTACIÓN (PLATAFORMAS TECNOLOGICAS) 35](#_Toc203571690)

[12.2. PLAN DE CAPACITACIÓN DE USARIOS DEL SISTEMA. 35](#_Toc203571691)

[12.3. MANUAL DEL USUARIO (PDF, MÓDULO AYUDA) 36](#_Toc203571692)

[12.4. COPIAS DE SEGURIDAD DE DATOS Y RESPALDOS 36](#_Toc203571693)

[12.5. GARANTÍA (CONTRATOS, ACUERDOS DE NIVEL DE USUARIO) 36](#_Toc203571694)

[13. ADOPCIÓN DE BUENAS PRÁCTICAS EN EL PROCESO DE DESARROLLO DE SOFTWARE. 36](#_Toc203571695)

[CONCLUSION 36](#_Toc203571696)

[Índice de diagramas 38](#_Toc203571697)

[Índice de tablas 38](#_Toc203571698)

[BIBLIOGRAFIA & CIBERGRAFIA 38](#_Toc203571699)

[GLOSARIO 38](#_Toc203571700)

[Anexos 38](#_Toc203571701)

[Actividades para el cronograma 39](#_Toc203571702)

[Entregables por fase del proyecto productivo 39](#_Toc203571703)

# INDICACIONES GENERALES

Este documento ofrece indicaciones para la elaboración de su trabajo final de proyecto formativo o sus siglas “TFPF”. Como cualquier documento académico sigue una estructura formal y una línea de discurso que debe facilitar su lectura y reflejar el trabajo que ha venido realizando durante todos los trimestres.

Tenga presente que en esta parte se debe tener la solicitud y aprobación de proyecto en las líneas determinadas.

# Formatos y plantilla

**Normas Apa Sena**

[**https://biblioteca.sena.edu.co/images/PDF/InstructivoAPA.pdf**](https://biblioteca.sena.edu.co/images/PDF/InstructivoAPA.pdf)

#### Estética y estilo de redacción

Es fundamental que el TFPF presente un aspecto elegante y correcto. Se trata de un documento académico y debe reflejar la madurez y el nivel formativo de una persona que ha finalizado una carrera o curso de formación técnica o tecnológica. Tener cuenta las siguientes recomendaciones en todas y cada una de las entregas que se realicen en cada trimestre y en especial, en el documento final:

* Verificar la originalidad del documento, asegurándose de que se citaron todas las fuentes consultadas y no existen textos de autoría ajena sin referenciar correctamente.
* Cuidar la presentación del documento. Comprobar que formatos como tipo y tamaño de letra, número de páginas, encabezados, justificación de párrafos, interlineado, etc., son correctos.
* Revisar la ortografía y la redacción. Se sugiere utilizar el corrector de Word para asegurar de que no se ha dejado ninguna errata. Una lectura detenida del documento también ayudará a detectar erratas, omisiones o redundancias. Si es posible, pida a alguien cercano que lo lea y dé su opinión sobre la redacción. Preste especial atención a los siguientes aspectos:
* Revise que los párrafos sigan un orden o hilo argumental lógico.
* Verifique que la información se presente de una manera que facilite su comprensión, definiendo los conceptos necesarios e incluyendo las citas bibliográficas pertinentes.
* Elimine párrafos demasiado cortos. Cada párrafo debería tener al menos tres oraciones.
* Elimine frases superfluas y repeticiones de ideas.
* Escriba siempre al menos un párrafo de introducción en cada apartado, explicando qué va a tratar en esa sección. Evite que aparezcan dos encabezados de nivel consecutivos sin ningún texto entre medias.
* Repase las citas bibliográficas. Compruebe que todas ellas son correctas y siguen la normativa APA.
* Asegúrese de que las figuras y las tablas se ven clara y correctamente, que haces alusión a ellas en el texto (“como se puede observar en la figura 3…”) e incluyen número y título, así como su procedencia o fuente.
* Compruebe que los índices se generan correctamente.

#### Normativas y citas

En todas las publicaciones del SENA se utiliza la Normativa APA para citar y referenciar las fuentes bibliográficas, Ejemplo: (Parker, 2020, p. 67) o (Johnson, 2017, pp. 39–41)

# ESTRUCTURA DEL DOCUMENTO PLANTILLA PROYECTO SENA

En esta sección se describe con mayor profundidad la estructura y los contenidos esperados en cada apartado de su TFPF, no necesariamente se debe elaborar de manera lineal.

# Resumen/abstract

El resumen se redacta en último lugar ya que recoge las contribuciones más importantes del documento. Es necesario tener muy clara y completa cada parte del documento para poder resumirlo correctamente, debe incluir cada una de las fases, Análisis, Planeación, Ejecución y Evaluación.

Tendrá una extensión de 150 a 300 palabras y deberá ofrecer una visión global de lo que el lector encontrará en el documento, destacando sus aspectos fundamentales. Deberá indicar claramente cuál es el objetivo principal del proyecto, la metodología seguida para alcanzarlo, los resultados obtenidos y la principal conclusión extraída.

A continuación, se indicarán de 3 a 5 palabras clave o keywords como descriptores del proyecto que lo enmarcan en unas temáticas determinadas y seguido elaborar el abstract, pero este ha de redactarse, no es aconsejable tomar la traducción literal de un traductor informático.

# Introducción

La introducción presenta el documento al lector: por qué se ha elegido el tema, qué interés personal se tiene en este proyecto (utilizando una fórmula académica e impersonal), cuáles son los problemas que se han encontrado y se plantean, qué resultados se quieren conseguir con el proyecto y cómo se va a enfocar. (Párrafo Mínimo 10 líneas).

En esta introducción se englobarán, también, los siguientes subepígrafes: justificación, planteamiento del problema y objetivos del proyecto.

# Planteamiento y justificación del problema

En este apartado se deberá presentar el objeto del proyecto y justificar su importancia.

La lectura de este apartado debe dar una idea clara de las razones, motivos e intereses que han llevado a la elección de este proyecto (explicar las razones). Es importante que refleje las motivaciones, necesidades o problemas que ha detectado (por qué se desarrollará) y que dan sentido al proyecto planteado (como se realizará). (Mínimo 10 renglones máximo 1 hoja) .

Por ejemplo: “En vista de las de las graves problemáticas de seguridad y pérdida de información que se han presentado en el proceso de compras a cargo del subgerente de la comercializadora, se hace necesario garantizar que la información de las importaciones se maneje en un sistema de información con una base de datos relacional en único servidor accesible vía intranet, sobre la cual se generarían copias de seguridad semanal y mensualmente.

Por su parte en el proceso de ventas, se necesitaría implementar un módulo de facturación donde los precios de los productos se consulten directamente desde la base de datos, evitando de esta forma problemas de digitación y facilitando a su vez la auditoria por parte del gerente de la comercializadora, quien podría conocer las ventas realizadas por los asesores comerciales en tiempo real y generar informes en el momento que lo desee. Adicionalmente la gestión de los clientes en una base de datos abre la posibilidad a facilitar el proceso de publicidad automática para todos los clientes vía correo electrónico.

Para la implementación del presente sistema de información se requeriría la adquisición de lectores de código de barras para el punto de venta y una impresora térmica para la facturación, agilizando de esta forma el proceso de venta al por mayor.”

# Objetivos del Trabajo final de proyecto productivo TFPF

Los objetivos, tanto el general como los específicos, orientan el proyecto, influyen en el planteamiento y en la metodología, y deben ser retomados en las conclusiones para examinar su cumplimiento. Deben reflejar exactamente qué se quiere lograr con el proyecto, así como los pasos que se van a seguir. Por eso, es importante:

* Que estén bien enunciados. Deben ser claros y concisos. Se redactan en infinitivo utilizando verbos como: **Definir, exponer, explicar, identificar, mostrar, argumentar, comprobar, ejemplificar, ordenar, relacionar, aplicar, demostrar, calcular, clasificar, construir, diseñar, medir, utilizar, analizar, comparar, determinar, justificar, seleccionar, evaluar, valorar, verificar.**
* Que sean abarcables. Al terminar el documento se tiene que comprobar si se han cumplido los objetivos; no deben ser excesivamente ambiciosos ni numerosos. Por tanto, a la hora de redactarlos se debe tener en cuenta que su consecución pueda ser evaluada, medida o verificada. Deben ser susceptibles de poder ser alcanzados con la realización del proyecto. Por ejemplo, “Desarrollar el Sistema de Información para la Gestión de Facturación e Inventario en la Comercializadora la Rebaja.” se trata de un objetivo realista y evaluable, frente a, por ejemplo, “Desarrollar el Sistema de Información en la Comercializadora la Rebaja”, objetivo demasiado genérico e impreciso para poder ser alcanzado y/o evaluado de forma adecuada en un TFPF.
* Que se diferencie entre el objetivo general y los objetivos específicos: El objetivo general indica la finalidad del proyecto: “diseñar un sistema de información para mejorar…”.
  1. Objetivo general del TFPF: Qué pretendemos conseguir con el proyecto y para qué lo hacemos.
  2. Objetivos específicos del TFPF**:** Los objetivos específicos son aquellos que concretan las acciones a realizar para el logro del objetivo general, por ejemplo: “diseñar los modelos entidad relación y modelo relacional según los procesos de la empresa…”.
* Lo más apropiado es que un proyecto tenga un objetivo general y tres o cuatro objetivos específicos, aunque estos números son orientativos.

NOTA: si los objetivos iniciales cambian a medida que avanzas en el proyecto, se pueden modificar para mantener la coherencia con el contenido del TFPF.

#### Alcance del proyecto

Define los límites del proyecto, hasta donde se va a realizar, que restricciones hay, concretamente cuales son los productos o resultados del proyecto. (Mínimo 10 renglones máximo 1 hoja)

Por ejemplo: El siguiente proyecto tiene como finalidad desarrollar un sistema de información para la gestión de facturación e inventario de la comercializadora “La Rebaja” tendrán presentes las fases de desarrollo, distribuidas en:

Fase de Análisis de la información, el cual se emplean métodos como entrevista, encuestas, método de observación, técnicas audiovisuales, sesión en grupo, con el fin de concluir aquellas falencias encontradas para posterior organización en el desarrollo.

En la fase de planeación, se estimará tiempos de trabajo, el cual permitirá al programador y diseñador concertar con el cliente fechas y actividades, dispuestas para el desarrollo de este.

Durante la fase de ejecución, se emplea como herramientas de programación fundamentales para la creación del código, visual y para la base de datos MYSQL. Se realizará pruebas, que garantizará que el proyecto cumpla con los requisitos solicitados por el cliente, de forma local, teniendo como objetivo que la información, se encuentre actualizada y lo más importante centralizada.

Por último, se le entregará una copia del Sistema de información al cliente, con los módulos de facturación e inventario totalmente funcionales.

# Matriz de Riesgo

La matriz de riesgos es una herramienta fundamental que permite identificar, analizar y priorizar los riesgos que pueden afectar el desarrollo de un proyecto de software. Su objetivo es anticipar posibles eventos que podrían tener un impacto negativo, de manera que se puedan tomar decisiones informadas y planificar acciones preventivas o correctivas.

# ELICITACIÓN DE REQUISITOS

La elicitación de requisitos es el primer paso en el proceso de ingeniería de requisitos. En esta fase se recopila información clave sobre las necesidades, expectativas y condiciones del sistema que se va a desarrollar. Para ello, se aplican diversas técnicas que permiten obtener una visión clara del problema y de las posibles soluciones.

* 1. Identificación de procesos: Identifica y describe los procesos actuales relacionados con el sistema que se va a desarrollar. Puedes apoyarte en entrevistas, observaciones o información proporcionada por el cliente.  
      Redacta los nombres de los procesos, su propósito y cómo se desarrollan actualmente, indicando fortalezas y debilidades que observes.
  2. Recolección de información del software a construir de acuerdo con las necesidades del cliente: Explicar cómo se va a recolectar la información necesaria para entender las necesidades del cliente.  
     Describa qué técnicas aplicarán (como entrevistas, encuestas, observación, etc.) y cómo estas te permitirán conocer a fondo el entorno del sistema actual y los requerimientos del nuevo software.
  3. Elección de la técnica de recolección de información: Seleccione mínimo dos técnicas de recolección de información. Por cada técnica elegida, redacte:
  + Stakeholder (Definición matriz).
  + Descripción cómo se aplicará de acuerdo con la técnica elegida.
  1. Diseño de los formatos, según la técnica o técnicas de recolección de la información seleccionada**:** Diseñe y anexe los formatos o instrumentos que utilizaron para aplicar las técnicas de recolección seleccionadas:

Formularios de entrevista.

Encuestas.

Listas de chequeo para observación.

Plantillas de recolección documental (bases de datos productos, clientes, entre otros)

* 1. Aplicación de la técnica de recolección de la información**:** Adicione los formatos creados en el entorno real del proyecto. Describir brevemente cómo fue el proceso de aplicación: a quién se entrevistó, qué se observó, cuántas personas participaron, qué dificultades se encontraron, etc.
  2. Análisis de la información recolectada**:** Realizar un análisis por cada técnica utilizada. Sistematizar los datos recolectados, incluir representaciones gráficas como tablas, diagramas o esquemas para mostrar mejor los resultados obtenidos.

#### ESPECIFICACIÓN DE REQUERIMIENTOS, DE ACUERDO CON LA INFORMACIÓN RECOLECTADA, APLICANDO ESTANDARES DE REQUERIMIENTOS

El análisis de requisitos es fundamental para el ciclo de vida del desarrollo de software, ya que establece los lineamientos para la nueva aplicación. Este proceso incluye identificar y documentar las necesidades del producto que se desarrollará, resultando en un documento de especificación que detalla las funcionalidades que debe cumplir el sistema. Los requisitos se definen como condiciones o capacidades necesarias para que los usuarios resuelvan problemas o logren objetivos específicos, abarcando también las normativas que el sistema debe cumplir. La participación de los usuarios en la definición de estos requisitos es crucial, ellos poseen el conocimiento práctico del sistema a automatizar. Es esencial que tanto analistas como clientes colaboren en este proceso, ya que el cliente a menudo no entiende completamente los aspectos técnicos del desarrollo, mientras que el analista puede no conocer a fondo el contexto del cliente. El documento de requisitos debe ser accesible y socializado con el cliente, minimizando malentendidos.

#### Requerimiento funcionales

Debe definirse uno general o agrupado, preferiblemente el primero debe ser Inicio de Sesión. Se define iniciando con RQF y un código de 3 dígitos iniciando en 001

Se sugiere utilizar el siguiente formato:

**Tabla No.1.** *Requerimiento funcional 001*

|  |  |
| --- | --- |
| **Código Requisito** | RQF001 |
| **Nombre** | Inicio de Sesión |
| **Tipo de Usuario** | ------- |
| **Prioridad** | Alta/Esencial |
| **Descripción:** | El sistema de información permitirá el acceso de los usuarios  al sistema previa validación de un usuario y contraseña y le mostrará las acciones a efectuar según su rol. |

**Fuente:** Elaboración propia

**Nota.** En la tabla se visualiza el requerimiento funcional 001 del módulo de usuario

**Tabla No.2.** *Requerimiento funcional 002*

|  |  |
| --- | --- |
| **Código Requisito** | RQF002 |
| **Nombre** | Gestionar Información Empleado |
| **Tipo de Usuario** | Gerente y SubGerente |
| **Prioridad** | Alta/Esencial |
| **Descripción:** | El sistema deberá permitir al Gerente y SubGerente registrar, consultar, modificar o inactivar la información de los empleados de la empresa tales como Número de  identificación, apellidos, nombres, dirección, teléfono, cargo. |

**Fuente:** Elaboración propia

**Nota.** En la tabla se visualiza el requerimiento funcional 002 del módulo de usuario

**Prioridad:** Relevancia de la funcionalidad en el sistema. Puede ser:

* Alta/Esencial o Media/Deseado
* Baja/Opcional
  1. REQUERIMIENTOS NO FUNCIONALES

Solicitar solo Requerimientos a nivel de Hardware y Diseño. Se define iniciando con RQNF y un código de 3 dígitos iniciando en 001. Se sugiere utilizar el siguiente formato:

**Tabla No.3** *Requerimiento No funcional 001*

|  |  |
| --- | --- |
| **Código Requisito** | RQNF001 |
| **Nombre** | Especificaciones Mínimas de Hardware y Software |
| **Descripción:** | Tener acceso a uno o más computadores: Mínimo 2 GB RAM  Sistema Operativo Windows 7 o superior Disco duro de ??? GB  Que tenga las herramientas y programas necesarios para llevar  a cabo el aplicativo, base de datos (SQL SERVER). |

**Fuente:** Elaboración propia

**Nota.**

**Tabla No.4** *Requerimiento No funcional 002*

|  |  |
| --- | --- |
| **Código Requisito** | RQNF002 |
| **Nombre** | Conectividad |
| **Descripción:** | La empresa debe contar con una red LAN que le permitirá conectar sus terminales de trabajo al servidor de una forma más adecuada para hacer uso del aplicativo |

**Fuente:** Elaboración propia

**Nota.**

**Tabla No.5** *Requerimiento No funcional 003*

|  |  |
| --- | --- |
| **Código Requisito** | RQNF003 |
| **Nombre** | Diseño e Imagen |
| **Descripción:** | El aplicativo debe hacer uso de colores y fuentes que faciliten  la adecuada visualización por parte del usuario y sean acordes con la imagen corporativa de la empresa. |

**Fuente:** Elaboración propia

**Nota.**

Generalidades sobre los requerimientos funcionales

1. Las interfaces principales deben ser responsive
2. La seguridad de la plataforma debe garantizar la integridad, disponibilidad y la privacidad.
3. Para los contenidos de procesos almacenados se realizaron pruebas de inyección de código SQL.

#### REQUERIMIENTOS NORMATIVOS

Cumplen con el conjunto de normas, estándares legales, políticas y lineamientos técnicos que deben cumplirse durante todo el ciclo de vida del proyecto. Están orientados a garantizar que el software producido sea seguro, confiable, cumpla obligaciones legales y se adhiera a las mejores prácticas aceptadas por la industria. Ejemplo:

El sistema debe controlar el acceso y permitir el ingreso únicamente a usuarios autorizados mediante nombre de usuario y contraseña, cumpliendo con las políticas internas de seguridad y normas legales de protección de datos.

La aplicación debe generar y almacenar trazas de auditoría para todas las operaciones críticas, conforme a la normatividad vigente que exige la capacidad de rastrear acciones para fines de control y cumplimiento.

#### REGLAS DEL NEGOCIO

Las reglas de negocio son directrices, condiciones o restricciones internas que determinan cómo debe operar una organización o un sistema dentro de un entorno específico. Estas reglas reflejan políticas, procesos, límites legales y prácticas del negocio y orientan tanto el desarrollo de software como la toma de decisiones operativas. Ejemplos de reglas de negocio:

Gestión de inventario:

"No se permite realizar una venta si la cantidad solicitada por el cliente supera el stock disponible en bodega."

Control de acceso de usuarios:

"Un empleado únicamente puede acceder al sistema fuera del horario laboral si cuenta con autorización especial del área de seguridad."

#### PROPUESTA TECNICA

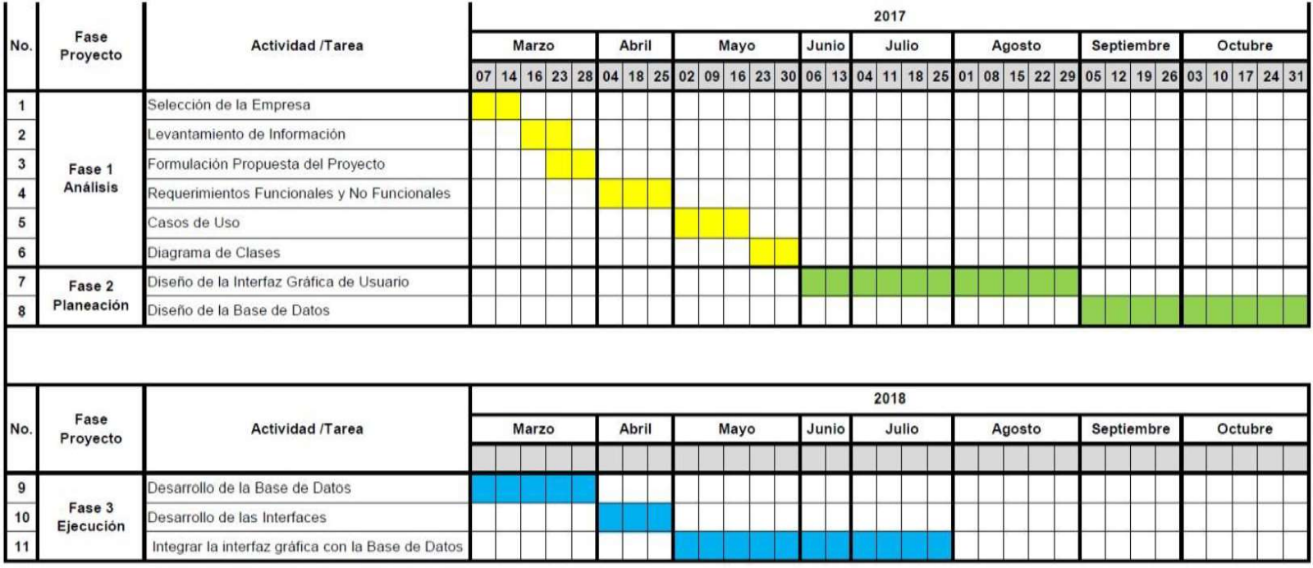
Se agregan los detalles técnicos que apoyaran el desarrollo del software como el tiempo de desarrollo, costos estimados, requisitos de hardware y software necesarios para el desarrollo del proyecto

#### Cronograma

Se debe desarrollar la temporalización general del proyecto (por fases, meses, semanas, días). La innovación es un proceso permanente de mejora, sin embargo, este tipo de proyectos responden a un proceso puntual por lo que se especificará en este apartado las fases, tiempos, así como la fecha de inicio y de fin del proyecto.

* Es importante realizar un cronograma con dicha temporalización para visualizarlo de forma más clara y directa. En el cronograma se puede incluir la secuenciación de objetivos, fases y actividades. Lo indicado sería aportar un cuadro o tabla. No olvide indicar frente a cada actividad, el nombre del responsable. Ejemplo:

**Tabla No.6** *Cronograma Proyecto*



**Fuente:** Elaboración propia

#### Costos Estimados

Tener cuenta necesidades del cliente, contratación, costos y requerimientos de hardware y software. Ejemplo:

**Tabla No.7** *Costos estimados*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Item | Unidad de Medida | Costo Unitario | Cantidad | Costo Total |
| 1 | Ingeniero de requerimientos | Mensual | 3.500.000 | 12 | 42.000.000 |
| 2 | Servidor en la nube | Anual | 5.000.000 | 1 | 5.000.000 |
| 3 | Dominio | Anual | 150.000 | 3 | 450.000 |

**Fuente:** Elaboración propia

#### Requisitos de Hardware

Especifican las características físicas y técnicas mínimas o recomendadas que debe tener el equipo donde se instalará, ejecutará o se probará el software. Estos requisitos aseguran que el sistema funcione de forma eficiente y estable, minimizando problemas de desempeño o incompatibilidad. Ejemplos:

1. El sistema debe ejecutarse en un equipo con al menos 8 GB de memoria RAM y un procesador Intel i5 o superior.
2. Es necesario contar con 50 GB de espacio libre en disco duro y una tarjeta gráfica compatible con DirectX 12.

#### Requisitos de Software

Detallan las aplicaciones, sistemas operativos y componentes adicionales que deben estar instalados para que el software funcione correctamente. Incluyen versiones específicas de sistemas operativos, frameworks, librerías y otros programas necesarios para la operación y desarrollo. Ejemplos:

1. El software debe instalarse en equipos con Windows 10 o superior y .NET Framework 4.8.
2. Requiere la presencia de Java 17 y base de datos PostgreSQL 13 para su ejecución y almacenamiento de datos.

# ANÁLISIS DE LA ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE.

#### ALTERNATIVAS DE SOLUCIÓN (PROTOTIPO O PROTOTIPOS DEL SISTEMA, MOCKUPS)

Los mockups son representaciones visuales detalladas de la interfaz de usuario de un sistema antes de su desarrollo, permiten visualizar y validar cómo lucirá y funcionará el producto final, facilitando la identificación temprana de problemas de diseño y usabilidad.

Programas sugeridos para desarrollar los mockups:

<https://mockflow.com/>

<https://wireframe.cc/>

<https://moqups.com/es/>

#### Historias de Usuario

permiten expresar necesidades o expectativas del usuario de forma clara, concisa y orientada a valor.

Componentes clave:

1. Rol: Quién se beneficia o realiza la acción (por ejemplo, “administrador”, “cliente”).
2. Necesidad: Qué se desea lograr o realizar (la funcionalidad específica).
3. Motivación/valor: Por qué se necesita, cuál es el beneficio esperado.

Ejemplo de historia de usuario

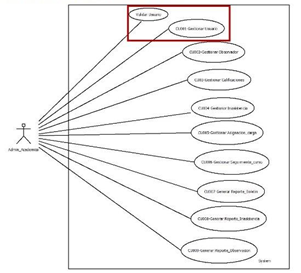
Como cliente registrado, quiero poder recuperar mi contraseña, para que pueda acceder al sistema en caso de olvido.

Como administrador, quiero poder ver un informe detallado de las ventas para tomar decisiones informadas.

#### DIAGRAMA DE CASOS DE USO Y EXTENSIBILIDAD DE ESTOS

Se debe hacer uso de una herramienta digital (por ejemplo DIA) Identificar cada caso de uso con **CU001**

**Figura No.1.** *Diagrama de caso de uso*



**Fuente:** Elaboración propia

**Nota.**

**Tabla No.8** *Documentación caso de uso*

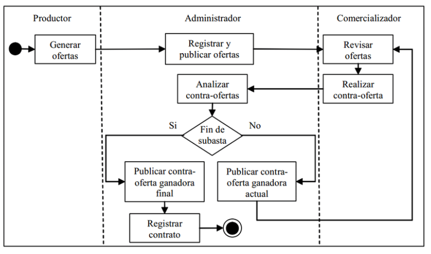
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. IDENTIFICACIÓN DE CASO DE USO | | | | | | | | | | | |
| 1.1 Id Caso | | | | CUD 002 | | 1.2 Nombre | Registrar Empleado | | | | |
| 2. HISTORICO DE CASO DE USO | | | | | | | | | | | |
| 2.1 Autor | | | | | Juan Pérez (aprendiz) | | | | | | |
| 2.2 Fecha Creación | | | | | 30/03/2017 | | | 2.3 Última Actualización | | | 30/03/2017 |
| 2.4 Actualizado por | | | | | Juan Pérez | | | | | 2.5 Versión | 1.0 |
| 3. DEFINICION DE UN CASO DE USO | | | | | | | | | | | |
| **3.1 DESCRIPCIÓN** | | | | | | | | | | | |
| Permite el registro de información de un nuevo empleado | | | | | | | | | | | |
| **3.2 ACTORES** | | | | | | | | | | | |
| Gerente, Subgerente, Vendedor | | | | | | | | | | | |
| **3.3 PRECONDICIONES** | | | | | | | | | | | |
| 1.3.1 El usuario debe estar autenticado en el sistema CU 001 | | | | | | | | | | | |
| **3.4 FLUJO NORMAL** | | | | | | | | | | | |
| Contando con las precondiciones el flujo normal será el siguiente: | | | | | | | | | | | |
|  | **Paso** | | **Actor** | | | | | | **Sistema** | | |  |
|  | 1 | | Usuario da clic en el menú principal en Gestión de Empleados | | | | | | Despliega una ventana con los servicios (Registrar Empleado, Consultar empleado, Actualizar  Empleado, Inactivar Empleado) | | |  |
|  | 2 | | Selecciona mediante un clic el ítem Registrar Empleados | | | | | | Muestra un formulario con la casilla activa para búsqueda de empleado por nombre y  documento. | | |  |
|  | 3 | | Usuario diligencia datos y da clic en buscar. | | | | | | Muestra formulario listo de coincidencias, si no hay se puede proceder con la creación del  empleado. | | |  |
|  | 4 | | Selecciona Registrar | | | | | | Muestra formulario de empleado, con los campos en blanco para registro de información | | |  |
|  | 5 | | Diligencia información solicitada del formulario y da clic en registrar | | | | | | Genera mensaje donde informa la creación exitosa del empleado. | | |  |
| **3.5 FLUJO ALTERNATIVO** | | | | | | | | | | | |
| Si existe otra forma de acceder al caso de uso, describir los pasos. | | | | | | | | | | | |
|  | **Paso** | **Actor** | | | | | | | **Sistema** | | |  |
|  |  |  | | | | | | |  | | |  |
|  |  |  | | | | | | |  | | |  |
| **3.5 FLUJO XCEPCIONAL** | | | | | | | | | | | |
|  | | | | | | | | | | | |
|  | **Paso** | **Actor** | | | | | | | **Sistema** | | |  |
|  | 4.a |  | | | | | | |  | | |  |
|  | 5.a | Usuario no llena completos los datos del formulario | | | | | | | El sistema muestra mensaje “ falta n campos por diligenciar” por favor completar | | |  |
| **3.7 POS CONDICIONES** | | | | | | | | | | | |
| Sistema genera confirmación de empleado registrado con éxito. | | | | | | | | | | | |
| **3.8 FRECUENCIA** | | | | | | | | | | | |
| Que frecuencia tiene el CU Alta Media Baja | | | | | | | | | | | |

**Fuente:** Elaboración propia

#### DIAGRAMAS DE ACTIVIDADES

Representan de manera sencilla y clara, cómo se desarrollan los procesos o flujos de trabajo dentro de un sistema de software, muestra de principio a fin, los pasos que siguen las personas o el sistema para completar una tarea específica. Ejemplo:

**Figura No.2** *Diagrama de actividad*



**Fuente:** <https://www.researchgate.net/figure/Figura-6-Ejemplo-de-Diagrama-de-Actividad_fig4_326207598>

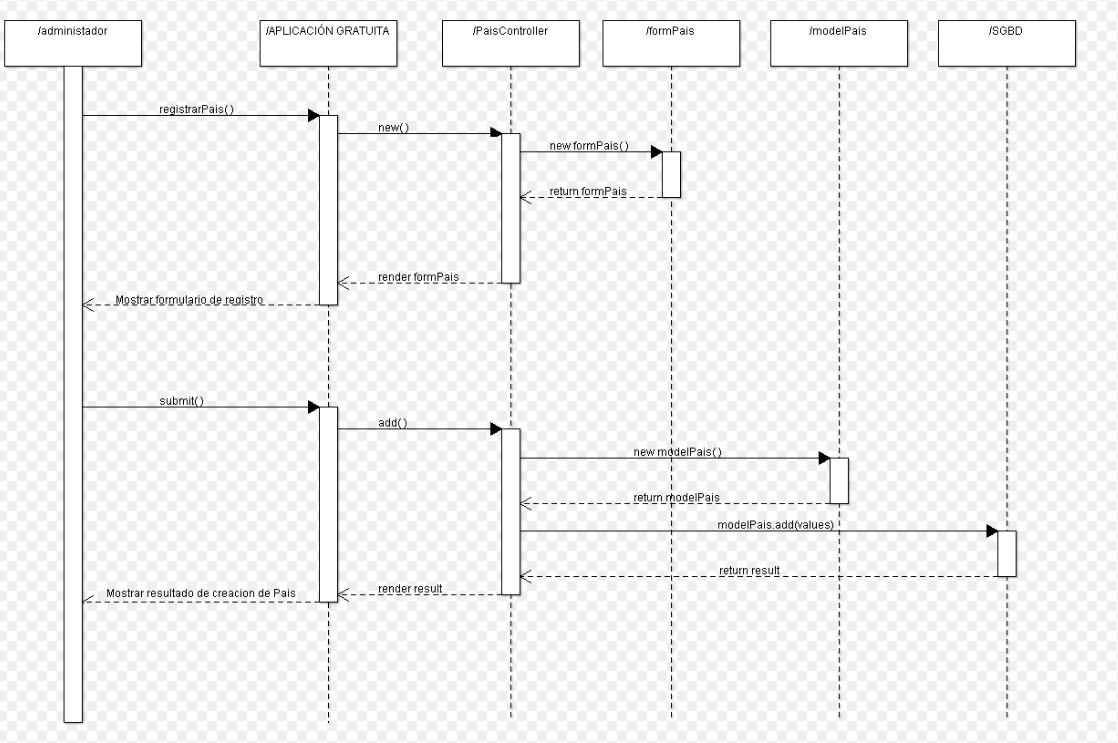
#### DIAGRAMA DE SECUENCIAS

Muestra de manera visual y ordenada cómo interactúan los diferentes objetos o componentes de un sistema en un escenario específico y en qué orden ocurren estas interacciones.

Los componentes a tener en cuenta son:

* Actores u objetos: Representan las entidades (usuarios, sistemas, módulos) que participan en la interacción. Se dibujan como rectángulos en la parte superior del diagrama.
* Líneas de vida: Líneas verticales discontinuas que descienden desde cada objeto o actor. Indican la existencia de ese objeto a lo largo del tiempo en la secuencia.
* Mensajes: Flechas horizontales que muestran la comunicación entre los objetos/actores. Hay varios tipos:
  1. Mensajes síncronos: El emisor espera respuesta antes de continuar.
  2. Mensajes asíncronos: El emisor no espera respuesta.
  3. Mensajes de respuesta: Indican la devolución de información.
  4. Mensajes de creación y destrucción: Indican la creación o eliminación de objetos.
* Cuadros de activación (barras de activación): Barras rectangulares sobre la línea de vida que indican el periodo durante el cual un objeto está realizando una acción o proceso.
* Fragmentos combinados o marcos: Rectángulos que agrupan partes del diagrama y representan alternativas, bucles, condiciones, etc. Por ejemplo: “opción”, “alternativa”, “bucle”. Ejemplo:

**Figura No.3** *Diagrama de secuencias*

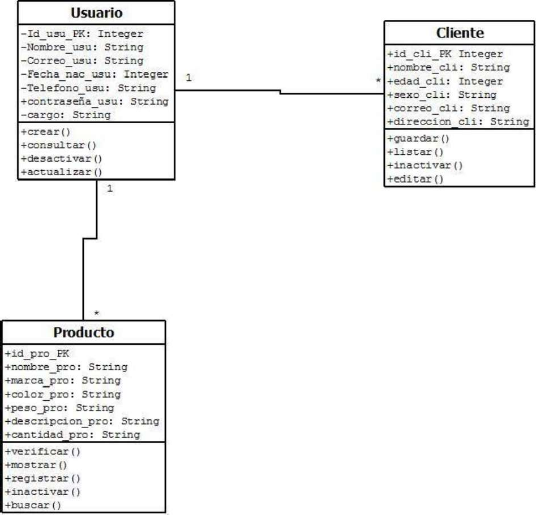


**Fuente:** wikipedia.org

#### CONSTRUIR EL MODELO DE DOMINIO DEL SISTEMA (DIAGRAMA DE CLASES).

* Se sugiere el uso de la herramienta digital .

**Figura No.4** *Diagrama de clases*



**Fuente:** Elaboración propia

**Nota.**

# DISEÑO DE LA SOLUCIÓN DEL SOFTWARE DE ACUERDO CON LOS PROCEDIMIENTOS Y REQUISITOS TÉCNICOS

#### Metodología

La metodología para utilizar, por ejemplo: principios metodológicos que seguiremos, formas de agrupamiento, recursos personales, roles, recursos, actividades, áreas implicadas, entre otros.

* Informe sobre Metodologías de desarrollo Ágil y descripción del por qué fue elegida.
* Aplicación de Metodología. Si fue SCRUM (Definición de Roles - Sprints - Tablero de Tareas)

Anexar la herramienta de gestión utilizada para realizar el seguimiento de las tareas (Trello, Jira, entre otras)

#### ARQUITECTURA DEL SOFTWARE (DIAGRAMA DE COMPONENTES), Y PATRONES DE DISEÑO DE SOFTWARE.

Evidencia de buenas prácticas, patrones o lineamientos empleados.

Por ejemplo:

* ¿El software desarrollado cumple con el patrón MVC o estructurado?
* Framework empleado.

#### DIAGRAMA DE DESPLIEGUE.

crea la vista de despliegue del software para determinar condiciones de la implantación de la solución informática.

#### DISEÑO FRONT-END - INTERFACES GRÁFICAS DE USUARIO (HTML.CSS Y JS)

Genera Plantillas Y Estilos De Acuerdo Con El Diseño Establecido.

Crea Interfaces De Usuario Aplicando Buenas Prácticas De Usabilidad Y Accesibilidad.

#### INTERFACES GRÁFICAS DE USUARIO MÓVILES.

Diseño de interfaz para dispositivos móviles: conceptos, dispositivos, características, reglas de diseño.

#### MAPA DE NAVEGACIÓN DE LA APLICACIÓN.

Construye el mapa de navegación cumpliendo reglas de usabilidad y accesibilidad, este facilita la comprensión de como los usuarios interactúan con el sistema

#### DETERMINAR TIPOS DE BASES DE DATOS.

#### POLÍTICAS DE SEGURIDAD DE LOS DATOS.

# CONSTRUCCIÓN DEL SOFTWARE

#### BASE DE DATOS PARA EL SOFTWARE A PARTIR DEL MODELO DE DATOS.

#### Modelo Entidad Relacion MER

Es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener.

**Figura No.5** *Modelo Entidad Relación MER*



Fuente: Elaboración propia

#### Modelo de Datos Diagrama ER

#### Modelo Relacional MR

#### OBJETOS DE LA BASE DE DATOS (PROCEDIMIENTOS ALMACENADOS, VISTAS, DISPARADORES)

Scrit BD..

Scrit Tablas

Select

PROCEDIMIENTOS ALMACENADOS

Los procedimientos almacenados son conjuntos de instrucciones SQL pre-compiladas y almacenadas que se ejecutan como una unidad.

#### VISTAS

Son consultas guardadas que se comportan como tablas virtuales. Ocultan la complejidad de las consultas subyacentes y simplifican el acceso a los datos.

#### DISPARADORES

Son procedimientos almacenados que se ejecutan automáticamente en respuesta a eventos en la base de datos, como inserciones, actualizaciones o eliminaciones de datos en una tabla.

#### DICCIONARIO DE DATOS.

**Tabla No.9** *Diccionario de datos*

Tabla

Descripción generada automáticamente

**Fuente:** Elaboración propia

**Nota.**

#### ESQUEMAS DE SEGURIDAD DE LOS DATOS.

Concepto, políticas, encriptación de datos, tratamiento de datos personales, niveles de accesos.

Son planes o modelos que establecen cómo se protegerá la información confidencial dentro de una organización, definiendo controles de acceso, permisos y otras medidas para prevenir el acceso no autorizado y proteger la integridad de los datos.

Pantalla de video juego

El contenido generado por IA puede ser incorrecto.**Figura No.6** Representación gráfica *Esquema de seguridad*

Fuente:

# CODIFICACIÓN DEL SOFTWARE DE ACUERDO CON EL DISEÑO ESTABLECIDO.

En esta sección se establecen y enmarcan la codificación a utilizar en el desarrollo del proyecto de acuerdo con las directrices y las reglas encontradas en el ámbito del desarrollo del software. A continuación, las mencionaremos de acuerdo con las necesidades establecidas.

FRONT: hace referencia explícitamente a la codificación aplicada en la parte de las vistas del sistema de información (Lo que el usuario va a observar en el momento de ejecución del sistema).

Backend: hace referencia a la codificación explicita desarrollada en la parte del código interno que manejara el sistema.

#### FRONT END (HTML, CSS, JAVASCRIPT, RESPONSIVE)

El frontend de una aplicación se refiere a la parte de la aplicación con la que el usuario interactúa directamente, es decir, la interfaz visual que ve y con la que puede interactuar. Engloba elementos como botones, menús, imágenes y textos que permiten al usuario usar la aplicación.

**Figura No.7** Representación gráfica *Dashboard*



**Fuente:**

#### ESTÁNDAR DE CODIFICACIÓN (DOCUMENTACIÓN DEL CÓDIGO, APLICABILIDAD DEL ESTANDAR DE CODIFICACIÓN).

Hace referencia al modelo que se utilizara en el desarrollo del código que aplicará el sistema, sea este en la parte del Front o Backend, en cualquiera de las dos formas mencionadas anteriormente, deberá utilizar cualquiera de los estilos conocidos en el ámbito de desarrollo, entre otros encontramos:

Método Húngaro:

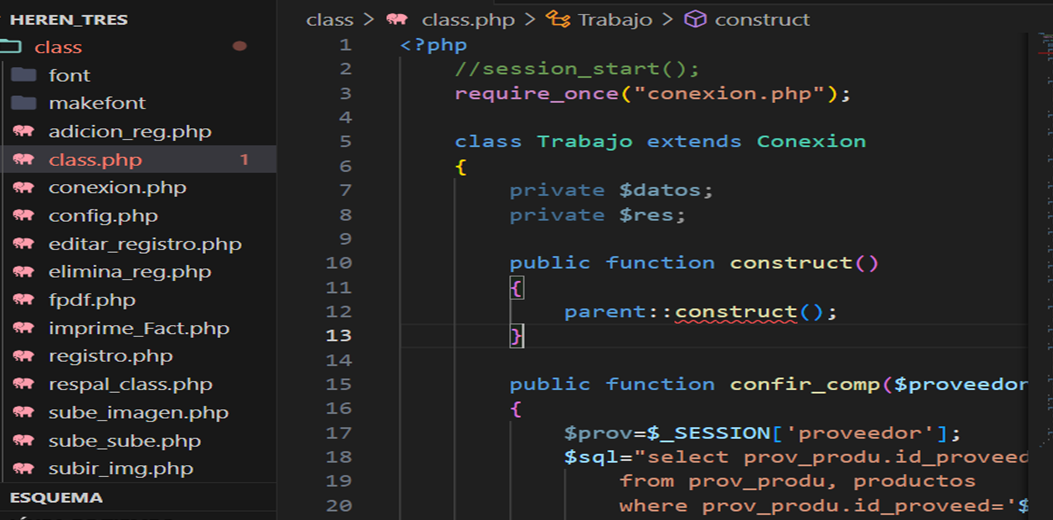
Método Camel Case

Definiciones variables: $datos, $datos\_Empleado, etc.

Codificación: mantener la indentacion del código a escribir

Ejemplo:

**Figura No.8** Estándar de codificación



**Fuente:**

#### CÓDIGO FUENTE DE LOS MÓDULOS DEL SOFTWARE WEB Y MÓVIL.

Hace referencia a las instrucciones escritas en el lenguaje de programación escogido por el Analista y/o Ingeniero y el cual debe cumplir a cabalidad con las directrices establecidas en los puntos anteriores.

#### SERVICIOS WEB.

Este concepto hace referencia principalmente a los diferentes servicios que utilizará o podrá utilizar nuestro sistema de información con otros servidores o aplicaciones, encontrados en la red:

Ejemplos

HTTP: Hipertext Transfer Protocol.

SOAP: Simple Object Access Protocol

#### CONTROL DE VERSIONES

Control de versiones: O Control de Código Fuente: este punto hace referencia al número de cambios realizados en el código fuente, desde su inicio a su finalización. por lo general se utiliza para este punto el versionamiento de software a través de herramientas como GIT y GITHUB. O en caso contrario el versionamiento manual que puede desarrollar el analista ante el sistema.

El control de versiones rastrea los cambios realizados en cada versión de un documento. Es una herramienta que organiza, etiqueta y conserva cada borrador hasta que se completa una versión final.

Herramientas para control de versiones:

Git

Subversion (SVN)

Mercurial

GIt – GitHub

Comandos básicos

Repositorio local

Repositorio remoto

#### PRUEBAS DEL SOFTWARE

En este punto se busca verificar que el código desarrollado cumpla con los requerimientos funcionales y no funcionales establecidos al inicio del levantamiento de requerimientos del software, la unicidad en el desarrollo del software y el cumplimento de los estándares de codificación encontrados y establecidos en el proyecto.

#### PLANEACIÓN Y DISEÑO DE LA PRUEBAS A NIVEL UNITARIO (MÓDULOS), CON BASE EN LO REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.

Hace referencia a los instrumentos, documentos y valores a utilizar en las pruebas a desarrollar en el sistema, sea esta; cualquiera de las tres siguientes opciones:

Unitarias: hacen referencia al análisis de resultados obtenidos de una unidad mínima del software, frente a los resultados esperados o suministrados en la operación realizada. En este punto es necesario aclarar, que en algunas ocasiones se desarrollan pruebas denominadas:

Pruebas de Caja Blanca (en la cual un sujeto obteniendo acceso al código fuente y la estructura interna del software, realiza las pruebas necesarias para verificar los resultados)

Pruebas de Caja Negra (este método desarrolla las pruebas necesarias del sistema, sin tener acceso al código fuente, para verificar los resultados finales)

Modulares: es un tipo de prueba, que busca hallar en forma explícita, los posibles errores en los módulos desarrollados en un sistema de información.

De Sistema: este tipo de prueba busca establecer y evaluar el comportamiento y rendimiento del sistema, frente a sus objetivos y alcances establecidos en la fase de levantamiento y diseño del sistema de información. De igual forma determina el cumplimiento del sistema frente a los requisitos funcionales y no funcionales.

#### EJECUCIÓN DE LAS PRUEBAS, INFORME DE HALLAZGOS

Este punto hace referencia a la forma en la cual se desarrollarán los diferentes tipos de pruebas del software a verificar, se tiene que tener encienta en este punto; que deberán desarrollarse los instrumentos físicos y/o magnéticos establecidos en la planeación y diseño de los tipos de pruebas a desarrollar ante el sistema, mencionados en los puntos anteriores.

#### CORRECCIÓN DEL ERRORES, Y DOCUMENTACIÓN DE LOS MISMOS.

Hace referencia a la verificación y respaldo de las modificaciones desarrolladas en los posibles errores encontrados en el proceso de ejecución de pruebas. En este caso se deberá generar un documento que respalde la verificación y aceptación de correcciones a los errores encontrados.

#### MANEJO DE ALERTAS / EXCEPCIONES

Hacen referencia a la forma de responder el sistema ante situaciones inesperadas y la posible forma controlada de terminar el proceso.

#### PLANEACIÓN Y DISEÑO DE LA PRUEBAS A NIVEL SISTEMA, CON BASE EN LO REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.

Este proceso establece las herramientas, métodos, soportes y documentación que se desarrollaran para dar soporte a las pruebas a realizar con el objeto de verificar el funcionamiento del sistema

#### EJECUCIÓN DE LAS PRUEBAS, INFORME DE HALLAZGOS

Este proceso se deberá desarrollar de acuerdo con los objetivos trazados en el punto anterior y cumplir los alcances establecidos para la verificación de las pruebas; deberá generarse un documento soporte que garantice el desarrollo, verificación de las pruebas realizadas

#### CORRECCIÓN DE ERRORES, Y DOCUMENTACIÓN DE ESTOS.

En base a los resultados obtenidos en el punto anterior, se deberá entregar el informe desarrollado, para que se inicien los cambios necesarios en los programas desarrollados; esperando la entrega de resultados y su posterior verificación.

#### PLAN DE DESPLIEGUE (HOSTING, DOMINIO, SUBDOMINIO, COSTOS, HERRAMIENTAS, PLATAFORMA, SEGURIDAD)

Identificación de los requerimientos, tanto lógicos como físicos, para la puesta en marcha del software. Se debe redactar el objetivo del despliegue, componentes involucrados (backend, frontend, bases de datos, API’s, etc), entornos involucrados (Desarrollo, Pruebas / QA, Staging / Pre-producción, Producción), equipo responsable, y cronograma para el despliegue.

Se deben explorar las diferentes opciones de hosting, en cuanto a los servicios ofrecidos, que garantices el correcto funcionamiento de la herramienta, alojamiento (dominio), almacenamiento (aplicación y bases de datos), correo electrónico, entre otros; teniendo en cuenta los costos, herramientas de software, infraestructura tecnológica que nos garanticen la funcionalidad, confiabilidad y seguridad de la información.

#### IMPLANTACIÓN DEL SOFTWARE**.**

#### PLAN DE IMPLANTACIÓN (PLATAFORMAS TECNOLOGICAS)

* 1. Preparar la plataforma tecnológica.
  2. Verificar el cumplimiento de las características mínimas de hardware requeridas para el software desarrollado.
  3. Diseñar el plan de migración de datos.
  4. Diseñar el plan de respaldo de los datos.
  5. Elaborar el plan de instalación.

#### PLAN DE CAPACITACIÓN DE USARIOS DEL SISTEMA.

Un plan de capacitación de usuarios para un sistema implica la planificación sistemática de actividades para asegurar que los usuarios comprendan y utilicen eficientemente el sistema. Este plan debe considerar las necesidades de capacitación, los objetivos, el contenido, la metodología, el cronograma y los recursos necesarios.

#### MANUAL DEL USUARIO (PDF, MÓDULO AYUDA)

Elaborar planes de mantenimiento y soporte del software

Documentar el proceso de migración y respaldo de los datos.

Elaborar manuales de instalación, técnico y de usuario.

#### COPIAS DE SEGURIDAD DE DATOS Y RESPALDOS

El software de copias de seguridad y respaldos, también conocido como software de respaldo, permite crear copias de datos importantes para protegerlos contra pérdidas o daños, ya sean causados por errores humanos, fallos del sistema, ataques cibernéticos o desastres naturales

#### GARANTÍA (CONTRATOS, ACUERDOS DE NIVEL DE USUARIO)

Una garantía es el compromiso del proveedor de que el software funcionará como se describe durante un período determinado. Si hay errores o fallos, se deben corregir, actualizar el software o incluso reemplazarlo.

#### ADOPCIÓN DE BUENAS PRÁCTICAS EN EL PROCESO DE DESARROLLO DE SOFTWARE.

#### CONCLUSIONES

La adopción de buenas prácticas en el desarrollo de software es crucial para crear sistemas robustos, seguros y eficientes. Estas prácticas abarcan desde la planificación y el análisis de requisitos hasta el mantenimiento y las actualizaciones, impactando positivamente en la calidad, la mantenibilidad y el éxito a largo plazo del producto. La implementación de buenas prácticas, como la automatización de procesos, pruebas rigurosas, y la colaboración efectiva, no solo mejora la calidad del software, sino que también reduce costos y tiempos de entrega.

Como cierre del documento, se debe recoger las conclusiones a las que se llegó consecuencia del TFPF realizado y, por lo tanto, deberá marcar el grado de consecución de los objetivos propuestos.

Debe tener en cuenta que las conclusiones son una parte importantísima del proyecto. Son el fruto del trabajo: los resultados que se han obtenido después de realizar el proyecto productivo, las reflexiones que han surgido tras tanto tiempo trabajando el proyecto. Por esta razón, en las conclusiones no se presentan nuevas ideas, pues son la evaluación personal, nuestras ideas finales sobre el proyecto, de ahí que no presenten citas de otros autores.

Las conclusiones ofrecidas deberán ser consecuencia del trabajo realizado y, por lo tanto, deberán marcar el grado de consecución de los objetivos propuestos (cada objetivo del documento se enlazará con una conclusión).

Un posible esquema de las conclusiones sería el siguiente:

* Empiece con un párrafo que retome el objetivo general del documento y qué problema o aspecto buscaba subsanar.
* Retome cada uno de los objetivos específicos y analice si los has cumplido y cómo.
* Argumente las principales aportaciones que ofrece su proyecto. Resalte lo relevante.
* Cierre este apartado con una frase especialmente cuidada que muestre la aportación a su proyecto.

BIBLIOGRAFIA & CIBERGRAFIA

Una vez que el documento está terminado se debe revisar que el apartado Referencias bibliográficas contiene todas las citas incluidas en el texto. Si se ha usado un sistema automático (un gestor bibliográfico tipo Endnote, Refworks o Mendeley), inserta la bibliografía en la opción adecuada (APA).

Si se ha realizado manualmente, repasa que todo es correcto: aparecen todas las referencias citadas en el texto, las referencias están ordenadas alfabéticamente por apellidos, las cursivas son correctas, los artículos tienen números de páginas, no faltan años ni ciudades de edición, se cumple en todas las referencias la normativa APA, etc.

# GLOSARIO

# Anexos

En los anexos se recoge todo aquello que puede ser interesante para el documento pero que no es estrictamente esencial, y que distraería la lectura si se colocara en el cuerpo: el cuestionario utilizado, el desarrollo de las actividades, listas de chequeo, encuestas, entrevistas aplicadas (diligenciadas por la empresa avaladas con firma y/o sello), fotos (proceso de observación), formatos (facturas, nóminas, plantillas, entre otros), tabulación en caso de aplicar encuestas o cuestionarios.

Se debe incluir todos los anexos que se consideren necesarios y los importantes que soporten o hagan parte funcional del proyecto (Manuales, informes, plan y aplicación de pruebas, propuestas, entre otros. Cada parte adicional se numera como un anexo y se ofrece en una página diferente.

# Actividades para el cronograma

Es importante describir en el cronograma las actividades que se proponen para trabajar en cada una de las fases, a continuación, se realiza una descripción:

#### Entregables por fase del proyecto productivo

En la tabla 10 se relaciona cada uno de los entregables por fases. Es imprescindible, que durante el proceso de revisión de proyectos tenga en cuenta no solo los entregables solicitados en la lista de chequeo, sino a qué fase de proyecto pertenece cada uno.

**Tabla No.10.** *Fases del proyecto*

|  |  |
| --- | --- |
| **FORMULACIÓN**  **DEL PROYECTO** | **PROPUESTA DEL PROYECTO**   * NOMBRE DEL PROYECTO * INTRODUCCION * JUSTIFICACIÓN DEL PROYECTO * DESCRIPCIÓN DEL PROBLEMA * OBJETIVOS * ALCANCE * IMPACTOS * CRONOGRAMA |
| **ANALISIS** | **ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE**   * IDENTIFICACIÓN DE PROCESOS? * RECOLECCIÓN DE LA INFORMACIÓN DEL SOFTWARE A CONSTRUIR DE ACUERDO CON LAS NECESIDADES DEL CLIENTE. * ELECCIÓN DE LA TECNICA DE RECOLECCION DE LA INFORMACION. * DISEÑO DE LOS FORMATOS, SEGÚN LA TÉCNICA O TÉCNICAS DE RECOLECCIÓN DE LA INFORMACIÓN SELECCIONADA(S). * APLICACIÓN DE LA TÉCNICA DE RECOLECCIÓN DE LA INFORMACIÓN. * ORGANIZACIÓN DE LA INFORMACIÓN RECOLECTADA. * ESPECIFICACIÓN DE REQUERIMIENTOS, DE ACUERDO A LA INFORMACIÓN RECOLECTADA, APLICANDO ESTANDARES DE REQUERIMIENTOS   **ANÁLISIS DE LA ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE.**   * ALTERNATIVAS DE SOLUCIÓN (PROTOTIPO O PROTOTIPOS DEL SISTEMA, MOCKUPS) * DIAGRAMA DE CASOS DE USO Y EXTENSIBILIDAD DE LOS MISMOS * DIAGRAMAS DE ACTIVIDADES, Y SECUENCIAS * CONSTRUIR EL MODELO DE DOMINIO DEL SISTEMA (DIAGRAMA DE CLASES). * ELABORAR EL MODELO ENTIDAD RELACIÓN.   **DISEÑO DE LA SOLUCIÓN DEL SOFTWARE DE ACUERDO CON LOS PROCEDIMIENTOS Y REQUISITOS TÉCNICOS**   * ARQUITECTURA DEL SOFTWARE (DIAGRAMA DE COMPONENTES), Y PATRONES DE DISEÑO DE SOFTWARE. * DIAGRAMA DE DESPLIEGUE. * INTERFACES GRÁFICAS DE USUARIO (HTML.CSS Y JS) * INTERFACES GRÁFICAS DE USUARIO MÓVILES. * MAPA DE NAVEGACIÓN DE LA APLICACIÓN. * DETERMINAR TIPOS DE BASES DE DATOS. * MODELO DE DATOS DIAGRAMA ER (NORMALIZACIÓN BASE DE LA DATOS). * DICCIONARIO DE DATOS. * POLÍTICAS DE SEGURIDAD DE LOS DATOS. |
| **DESARROLLO** | **CONSTRUCCIÓN DEL SOFTWARE.**   * BASE DE DATOS PARA EL SOFTWARE A PARTIR DEL MODELO DE DATOS. * OBJETOS DE LA BASE DE DATOS (PROCEDIMIENTOS ALMACENADOS, VISTAS, DISPARADORES). * ESQUEMAS DE SEGURIDAD DE LOS DATOS. * CODIFICACIÓN DEL SOFTWARE DE ACUERDO CON EL DISEÑO ESTABLECIDO. * ESTÁNDAR DE CODIFICACIÓN (DOCUMENTACIÓN DEL CÓDIGO, APLICABILIDAD DEL ESTANDAR DE CODIFICACIÓN). * CÓDIGO FUENTE DE LOS MÓDULOS DE SOFTWARE WEB Y MÓVIL. * SERVICIOS WEB. * PRUEBAS DEL SOFTWARE * PLANEACIÓN Y DISEÑO DE LA PRUEBAS A NIVEL UNITARIO(MÓDULOS), CON BASE EN LO REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES. * EJECUCIÓN DE LAS PRUEBAS, INFORME DE HALLAZGOS * CORRECCIÓN DEL ERRORES, Y DOCUMENTACIÓN DE LOS MISMOS. * PLANEACIÓN Y DISEÑO DE LA PRUEBAS A NIVEL SISTEMA, CON BASE EN LO REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES. * EJECUCIÓN DE LAS PRUEBAS, INFORME DE HALLAZGOS * CORRECCIÓN DEL ERRORES, Y DOCUMENTACIÓN DE LOS MISMOS. |
| **IMPLEMENTACIÓN** | **IMPLANTACIÓN DEL SOFTWARE.**   * PLAN DE IMPLANTACIÓN (PLATAFORMAS TECNOLOGICAS) * PLAN DE CAPACITACIÓN DE USARIOS DEL SISTEMA. * MANUAL DEL USUARIO (PDF, MÓDULO AYUDA) * COPIAS DE SEGURIDAD DE DATOS Y RESPALDOS * GRANTÍA.   ADOPCIÓN DE BUENAS PRÁCTICAS EN EL PROCESO DE DESARROLLO DE SOFTWARE. |
| **VERIFICACIÓN** | 1. Documento de proyecto en norma APA y en digital que incluya la fase de análisis,   desarrrollo y implementación completa y con las correcciones aplicadas.   1. Proyecto de aplicación desarrollado en el lenguaje Html, Css, JavaScript, Php y Mysql por proyectos):   Verificación de implementación de CRUD con procedimientos almacenados para el(los) módulos 100% funcional.  Validación campos, campos calculados, control de errores y excepciones Login de usuario y privilegios de usuario mínimo para 2 Actores (Roles) Manual de usuario  Manual de usuario  4.2 Planificar y desplegar el software Construido  4.3 Testing del software |

**Fuente:** Nestor montaño

**2.11** Control del Documento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | Lyda Patricia Caicedo Monroy | Instructora | Regional Distrito capital. CDM | Septiembre 2024 |
|  | Washington Nieto | Instructor | Regional Distrito capital. CDM | Septiembre 2024 |
|  | Luis Enrique Arias Chavarro | Instructor | Regional Distrito capital. CDM | Septiembre 2024 |
|  | Nestor Montaño | Instructor | Regional Distrito capital. CDM | Septiembre 2024 |
|  | Andres Zambrano | Instructor | Regional Distrito capital. CDM | Septiembre 2024 |
|  | Norelys Buelvas | Instructora | Regional Distrito capital. CDM | Septiembre 2024 |

Control de cambios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | Lyda Patricia Caicedo Monroy | Instructora | Regional Distrito capital. CDM | Julio 15 2025 |
|  | Washington Nieto | Instructor | Regional Distrito capital. CDM | Julio 15 2025 |
|  | Andres Zambrano | Instructor | Regional Distrito capital. CDM | Julio 15 2025 |
|  | Jefry | Instructor | Regional Distrito capital. CDM | Julio 15 2025 |
|  | William Herreño Meneses | Instructor | Regional Distrito capital. CDM | Julio 15 2025 |
|  |  | Instructor | Regional Distrito capital. CDM | Julio 15 2025 |