

**INCATEC**

**Título del proyecto**

**MARIA**

Presentado por:

**Maria Antonia Diaz Mendoza**

**Jhon Anthony Uribe Ahumada**

**Miguel Alberto Medrano**

**Juan Manuel Arenas Perdomo**

Grupo:

**EQUIPO DE SISTEMAS INCATEC**

Evidencia de desarrollo de Software.

**Barranquilla – Colombia**

**Abril, 2025**



## Historial de Cambios

Revisión	Fecha	Ítem	Descripción de Cambio	Responsable
1	10/04/20 25	1-6	SE DA INICIO A LA DOCUMENTACION DE MARIA	JUAN MANUEL ARENAS PERDOMO

## Tabla de contenido.

<b>1. Proyecto de Producto de Software. ....</b>	<b>6</b>
1.1. IDENTIFICACIÓN Y CONTEXTO DEL PROBLEMA. ....	6
1.1.1. <i>Contexto del problema.</i> ....	6
1.1.1.1. Análisis de riesgo. ....	6
1.1.1.1.1. Identificación de riesgos. ....	6
1.1.1.1.2. Estimación de riesgos. ....	7
1.1.1.1.3. Resultados de estimación de riesgos. ....	8
1.1.2. <i>Identificación del problema.</i> ....	8
1.2. PROPÓSITO, ALCANCE Y OBJETIVO DEL PROYECTO. ....	9
1.2.1. <i>Propósito.</i> ....	9
1.2.2. <i>Alcance.</i> ....	10
1.2.3. <i>Objetivo del proyecto</i> ....	10
1.2.3.1. Objetivo general. ....	10
1.2.3.2. Objetivos específicos. ....	11
1.3. SUPOSICIONES Y RESTRICCIONES. ....	11
1.3.1. <i>Suposiciones.</i> ....	11
1.3.2. <i>Restricciones.</i> ....	12
1.3.2.1. Restricciones de alcance. ....	12
1.3.2.2. Restricciones de tiempo. ....	12
1.3.2.3. Restricciones de recursos. ....	13
1.4. ENTREGABLES DEL PROYECTO. ....	14
1.4.1. <i>Estructura de desglose de trabajo (EDT/WBS).</i> ....	14
1.5. CRONOGRAMA Y RESUMEN DEL PRESUPUESTO. ....	15
1.5.1. <i>Cronograma.</i> ....	15
1.5.2. <i>Presupuesto.</i> ....	16
<b>2. Documentos de referencia.....</b>	<b>17</b>
2.1. REFERENCIAS. ....	17
2.2. ESTÁNDARES. ....	17
<b>3. Definiciones y acrónimos.....</b>	<b>17</b>
3.1. DEFINICIONES. ....	17
3.2. ACRÓNIMOS. ....	¡ERROR! MARCADOR NO DEFINIDO.
<b>4. Resumen del trabajo requerido. ....</b>	<b>18</b>
<b>5. Organización y control del proyecto.....</b>	<b>21</b>
<b>6. Proceso de desarrollo de software.....</b>	<b>21</b>
6.1. ENTORNO DEL PROCESO DE SOFTWARE. ....	21
6.2. DISEÑO DE SOFTWARE. ....	22
6.2.1. <i>Definición de roles o actores.</i> ....	22
6.2.2. <i>Análisis de requerimientos de software.</i> ....	23
6.2.3. <i>Requerimientos funcionales.</i> ....	24
6.2.4. <i>Requerimientos no funcionales.</i> ....	24
6.3. MODELADO DE SOFTWARE. ....	25

6.3.1.	<i>Diagramas de Casos de uso.</i>	25
6.3.2.	<i>Diagrama de secuencia.</i>	25
6.3.3.	<i>Diagrama de actividades.</i>	25
6.3.4.	<i>Diagramas de clases.</i>	25
6.4.	DISEÑO DE LA INTERFAZ.	25
6.4.1.	<i>Arquitectura de información.</i>	25
6.4.2.	<i>Wireframe.</i>	26
6.5.	DISEÑO DE DATOS.	26
6.5.1.	<i>Método de acceso a la base de datos.</i>	26
6.5.2.	<i>Definición de las entidades de datos.</i>	26
6.5.3.	<i>Diagrama entidad relación.</i>	26
6.5.4.	<i>Diccionario de datos.</i>	26
6.5.5.	<i>Restricciones de seguridad e integridad.</i>	26
6.6.	DISEÑO DE LA ARQUITECTURA DE SOFTWARE.	26
6.6.1.	<i>Patrón de diseño.</i>	26
6.6.2.	<i>Diagrama de componentes.</i>	26
6.6.3.	<i>Diagrama de despliegue.</i>	26
6.7.	CONSTRUCCIÓN E IMPLEMENTACIÓN DE SOFTWARE.	27
6.7.1.	<i>Integración del sistema de software.</i>	27
6.7.2.	<i>Verificación y validación del sistema de software.</i>	27
6.7.3.	<i>Integración de software en hardware.</i>	27
6.7.4.	<i>Transición de software y gestión de lanzamientos.</i>	27
6.7.5.	<i>Mantenimiento de software.</i>	27
6.7.6.	<i>Gestión de defectos y problemas de software.</i>	27
<b>7.</b>	<b>Aspectos éticos del proyecto.</b>	<b>27</b>
7.1.	CONSIDERACIONES ÉTICAS EN LA PLANEACIÓN DEL PROYECTO.	27
7.2.	CONSIDERACIONES ÉTICAS EN EL DESARROLLO DEL PROYECTO.	27
7.3.	CONSIDERACIONES ÉTICAS EN LA ETAPA DE PRODUCCIÓN Y MANTENIMIENTO DE LA SOLUCIÓN APORTADA POR EL PRODUCTO DESARROLLADO POR EL PROYECTO.	27
<b>8.</b>	<b>Conclusiones.</b>	<b>27</b>
8.1.	CONCLUSIONES DEL ESTUDIANTE.	27
8.2.	CONCLUSIÓN DEL TUTOR EN SITIO DEL PROYECTO.	27
8.3.	LECCIONES APRENDIDAS	28
<b>Anexos.</b>		<b>29</b>

## 1. Proyecto de Producto de Software.

### 1.1. Identificación y Contexto del Problema.

#### 1.1.1. Contexto del problema.

Durante varios años, el Instituto Técnico de Administración y Salud INCATEC utilizó un sistema denominado **SIC** (Sistema de Información de Control), desarrollado con tecnologías de generación anterior. Este sistema operaba sobre una base de datos en **MySQL versión 5.0**, sin contar con una estructura relacional sólida ni mecanismos adecuados de control de integridad y seguridad de la información.

Desde el punto de vista **histórico y tecnológico**, el sistema fue creado como una solución interna con limitaciones técnicas importantes. Su diseño no contemplaba la escalabilidad, trazabilidad de acciones, ni la interoperabilidad con nuevas plataformas digitales. Con el crecimiento institucional y el aumento de la demanda en la gestión académica, administrativa y financiera, comenzaron a evidenciarse fallas estructurales que afectaron seriamente la operación del instituto.

En el aspecto **social y operativo**, los usuarios del sistema, tanto administrativos como estudiantes, reportaban frecuentemente dificultades para acceder y comprender la plataforma, debido a una interfaz poco amigable y procesos poco intuitivos. La falta de validaciones y automatizaciones provocaba errores constantes en los registros.

Desde una **perspectiva económica**, las fallas en el sistema derivaron en pérdidas asociadas a recaudos mal gestionados, duplicidad de pagos, y generación de reportes financieros inexactos, afectando los procesos de toma de decisiones y la planeación estratégica institucional.

Una auditoría forense realizada en 2024 evidenció además vulnerabilidades críticas de seguridad, datos huérfanos, duplicados e inexactos, y la existencia de recaudos financieros no asociados correctamente, exponiendo la posibilidad de **fraudes internos**, situación que comprometió la confianza y reputación institucional.

En respuesta a este entorno problemático, surge la necesidad de diseñar un nuevo producto de software que no solo subsane las fallas del sistema anterior, sino que se proyecte como una plataforma moderna, segura, escalable y centrada en el usuario: el sistema **MARIA**.

#### 1.1.1.1. Análisis de riesgo.

##### 1.1.1.1.1. Identificación de riesgos.

La permanencia del sistema SIC como plataforma institucional generó una serie de riesgos latentes y evidentes, los cuales fueron identificados como factores críticos para la operación y la toma de decisiones en **INCATEC**. A continuación, se listan los principales riesgos asociados al problema:

- **Riesgo de pérdida de integridad de datos**

La ausencia de integridad referencial por falta de claves foráneas y restricciones estructurales permitió la proliferación de datos huérfanos, duplicados, incompletos y erróneos, impactando directamente la fiabilidad de la información institucional.

- **Riesgo operativo**  
Las inconsistencias en los datos y la imposibilidad de generar reportes confiables comprometieron la eficiencia de los procesos académicos, administrativos y financieros, generando retrasos, reprocesos y decisiones mal fundamentadas.
- **Riesgo de fraude**  
La auditoría forense evidenció recaudos duplicados y no asociados a estudiantes, así como registros de pagos sin respaldo transaccional claro, lo que facilitó escenarios para el uso indebido de los recursos institucionales.
- **Riesgo reputacional**  
La divulgación de información errónea, errores en facturación o comunicaciones institucionales derivadas de datos inexactos afectaron negativamente la percepción de estudiantes, familias y aliados estratégicos frente a la confiabilidad de la institución.
- **Riesgo de seguridad informática**  
La carencia de controles de acceso robustos, cifrado de datos y monitoreo activo de la base de datos dejó expuestos los sistemas a vulnerabilidades que podrían ser explotadas por actores maliciosos, con consecuencias legales y de protección de datos personales.
- **Riesgo financiero**  
La falta de control y validación en los procesos de recaudo y facturación representó un impacto económico directo, tanto por pérdidas como por la necesidad de realizar procesos correctivos manuales que implicaban gasto de tiempo y recursos humanos.

La identificación de estos riesgos permitió visualizar con claridad la necesidad urgente de reemplazar el sistema SIC por una solución moderna, confiable y segura, lo cual se materializa en la creación del sistema **MARIA**.

#### **1.1.1.1.2. Estimación de riesgos.**

La estimación de los riesgos asociados al sistema anterior (SIC) se realizó tomando en cuenta los hallazgos de auditoría, las afectaciones históricas y las observaciones de los usuarios del sistema. Esta evaluación se basa en dos dimensiones principales:

- **Probabilidad de ocurrencia:** Baja, Media o Alta.
- **Impacto en la operación institucional:** Bajo, Medio o Alto.

#### 1.1.1.1.3. Resultados de estimación de riesgos.

RIESGO IDENTIFICADO	PROBABILIDAD	IMPACTO	NIVEL DE RIESGO
PÉRDIDA DE INTEGRIDAD DE DATOS	Alta	Alto	Crítico
INCONSISTENCIAS OPERATIVAS	Alta	Alto	Crítico
RIESGO DE FRAUDE FINANCIERO	Media	Alto	Alto
RIESGO REPUTACIONAL	Media	Medio	Medio
RIESGO DE SEGURIDAD INFORMÁTICA	Alta	Alto	Crítico
PÉRDIDAS FINANCIERAS	Media	Medio	Crítico

#### Interpretación:

- **Riesgo crítico:** Requiere atención inmediata con acciones correctivas prioritarias.
- **Riesgo alto:** Debe ser gestionado con planes de mitigación urgentes.
- **Riesgo medio:** Requiere monitoreo y estrategias de control preventivo.

Esta estimación validó la urgencia de reemplazar el sistema SIC por una solución más segura y estructurada. Así, **MARIA** se proyecta como un sistema que desde su diseño incorpora mecanismos para **mitigar estos riesgos**, a través de buenas prácticas de desarrollo, seguridad, y administración de datos.

#### 1.1.2. Identificación del problema.

La necesidad de transformación digital en el Instituto Técnico de Administración y Salud **INCATEC** evidenció una problemática estructural en el sistema que se venía utilizando: el **SIC**. Este sistema, implementado en versiones antiguas de MySQL (5.0) y alojado en servidores con configuraciones limitadas, presentaba un conjunto de deficiencias técnicas que afectaban directamente la operación institucional y la experiencia de los usuarios.

La problemática fue identificada a partir de:

- Quejas recurrentes de usuarios administrativos y estudiantes por **fallas de acceso, lentitud** y errores frecuentes en el sistema.
- Dificultades en la generación de reportes financieros, académicos y administrativos por **falta de integridad en los datos**.



- Imposibilidad de escalar el sistema o integrarlo con nuevas plataformas digitales, limitando los procesos de modernización institucional.

#### **Entorno del problema**

Desde una perspectiva institucional, **INCATEC** experimenta un crecimiento en la demanda de programas de formación, lo que exige sistemas de información robustos, escalables y seguros. Sin embargo, el entorno tecnológico del sistema **SIC** no respondía a esas necesidades:

- **Falta de relaciones entre tablas** en la base de datos, lo cual generaba información inconexa y redundante.
- Existencia de **datos huérfanos, duplicados e inconsistentes**, tal como lo evidencia el **Informe de Auditoría Forense (2024/08/05)**.
- **Recaudos financieros sin trazabilidad** y casos de duplicidad de pagos registrados a distintos estudiantes.
- Exposición a **vulnerabilidades críticas de seguridad**, como la ausencia de cifrado de datos y control de accesos.

#### **Impacto en las personas**

La problemática afectaba directamente a las personas involucradas en los procesos institucionales:

- El personal administrativo debía realizar procesos **manuales y repetitivos** para corregir errores del sistema.
- Los estudiantes recibían **información incorrecta** sobre pagos, matrículas y certificados.
- La comunidad educativa comenzó a **desconfiar** de los sistemas institucionales por la falta de precisión y transparencia.

Estas necesidades reales de la comunidad educativa de **INCATEC**, sumadas a los hallazgos técnicos de la auditoría, justifican plenamente la identificación del problema: un sistema obsoleto, inseguro y poco eficiente, que se convirtió en un **obstáculo para la gestión institucional** y que requería una **renovación total mediante un nuevo producto de software: MARIA**.

## **1.2. Propósito, alcance y objetivo del proyecto.**

### **1.2.1. Propósito.**

El propósito del proyecto **MARIA** es diseñar, desarrollar e implementar un sistema de información académico, administrativo y financiero que brinde una **solución integral, segura y escalable** a las necesidades operativas del Instituto Técnico de Administración y Salud INCATEC.

Este nuevo producto de software tiene como finalidad **sustituir el antiguo sistema SIC**, superando sus deficiencias estructurales y ofreciendo una plataforma moderna que permita:

- Gestionar eficientemente los procesos institucionales.
- Garantizar la **integridad, trazabilidad y seguridad** de los datos.
- Mejorar la experiencia del usuario a través de una interfaz intuitiva y funcional.
- Facilitar la **toma de decisiones estratégicas**, gracias a reportes precisos y en tiempo real.

- Adaptarse a futuras necesidades mediante una arquitectura modular y extensible.

En síntesis, cuando **MARIA** se encuentre en producción, se espera contar con una **herramienta tecnológica confiable y centrada en el usuario**, que eleve la calidad de la gestión educativa de **INCATEC**, fortalezca la transparencia institucional y optimice el uso de los recursos.

### 1.2.2. Alcance.

El proyecto **MARIA** contempla el desarrollo e implementación de un sistema de información que abarque integralmente los procesos **académicos, administrativos y financieros** del Instituto Técnico de Administración y Salud **INCATEC**, con un enfoque modular y escalable.

El alcance del proyecto incluye:

- **Diseño y desarrollo** de una base de datos relacional robusta, normalizada y segura, que garantice integridad y consistencia en los datos.
- **Implementación de módulos funcionales como:**
  - Gestión de usuarios y control de acceso por roles.
  - Administración de sedes, periodos académicos y programas de formación.
  - Matrículas, asignación de cursos, registro de calificaciones y seguimiento académico.
  - Gestión financiera: recaudos, facturación, reportes y control de cartera.
  - Reportes administrativos y académicos con trazabilidad total.
- Interfaz de usuario intuitiva, optimizada para dispositivos de escritorio.
- Panel de gestión con opciones diferenciadas según el rol del usuario (colaborador, estudiante, docente, etc.).
- Registro y control de acciones del sistema (logs) para auditoría y seguridad.
- **Migración segura de información histórica desde el sistema anterior (SIC).**
- Capacitación a usuarios clave y documentación técnica del sistema.

Este alcance puede extenderse en futuras fases para incluir funcionalidades como interoperabilidad con otras plataformas (pagos electrónicos, LMS, etc.), **integración móvil**, y analítica avanzada para toma de decisiones.

### 1.2.3. Objetivo del proyecto

#### 1.2.3.1. Objetivo general.

- Desarrollar un sistema de información integral, seguro y escalable denominado **MARIA**, que permita gestionar los procesos académicos, administrativos y financieros del Instituto Técnico de Administración y Salud **INCATEC**, garantizando la integridad, trazabilidad y confiabilidad de los datos, y mejorando la experiencia del usuario institucional.

#### 1.2.3.2. Objetivos específicos.

- Diseñar e implementar una base de datos relacional segura y normalizada que permita mantener la integridad y consistencia de la información institucional.
- Desarrollar módulos funcionales para la gestión académica, administrativa y financiera, alineados con las necesidades operativas de **INCATEC**.
- Implementar un sistema de control de accesos y permisos por roles que garantice la seguridad y trazabilidad de las acciones realizadas por los usuarios.
- Diseñar una interfaz de usuario intuitiva y amigable que facilite la navegación y el uso eficiente del sistema por parte del personal administrativo, docentes y estudiantes.
- Integrar herramientas de monitoreo y generación de reportes que permitan la toma de decisiones basada en datos confiables y en tiempo real.
- **Garantizar la migración segura y controlada de los datos históricos del sistema SIC al nuevo sistema MARIA.**
- Capacitar a los usuarios clave en el uso del sistema y entregar documentación técnica y funcional para su mantenimiento y evolución.

### 1.3. Suposiciones y restricciones.

#### 1.3.1. Suposiciones.

##### 1.3.1. Suposiciones

Durante la planificación y desarrollo del proyecto **MARIA**, se han establecido una serie de suposiciones clave que permiten estructurar las etapas del proyecto bajo ciertos escenarios previstos. Estas suposiciones, aunque no garantizadas, son consideradas válidas para la ejecución efectiva del sistema:

- **El equipo de desarrollo cuenta con las competencias técnicas necesarias** para diseñar, implementar y mantener el sistema utilizando tecnologías como Django, PostgreSQL, HTML, CSS y JavaScript.
- **La alta dirección de INCATEC respalda el proyecto institucionalmente**, proporcionando los recursos humanos, técnicos y logísticos requeridos para su ejecución.
- **Los usuarios finales (administrativos, docentes y estudiantes) estarán disponibles para participar en procesos de prueba, validación y retroalimentación del sistema.**
- **La infraestructura tecnológica existente no es suficiente para alojar y operar el sistema MARIA** en sus primeras fases, o bien se realizarán ajustes progresivos para su adecuada implementación.
- **La información proveniente del sistema SIC será accesible y estará en condiciones de ser migrada**, con los debidos procesos de depuración y transformación de datos. **(alto riesgo de implementación)**
- **Se contará con el compromiso institucional de adoptar el sistema MARIA como plataforma oficial**, una vez superada la etapa de pruebas y puesta en marcha controlada.

- **Se prevé un crecimiento progresivo en el uso del sistema**, lo cual permitirá su mejora continua a través de nuevas funcionalidades, adaptaciones o integraciones futuras.

### **1.3.2. Restricciones.**

#### **1.3.2.1. Restricciones de alcance.**

- Aunque el proyecto **MARIA** está diseñado para ser un sistema integral, durante su planeación se han establecido ciertas restricciones de alcance que determinan los límites de lo que será abordado en esta etapa inicial del desarrollo:
- No se incluye el desarrollo de una aplicación móvil nativa en esta fase. El sistema estará disponible inicialmente para acceso desde dispositivos de escritorio mediante navegador web.
- No se contempla la integración inmediata con sistemas externos, tales como, pasarelas de pago en línea o sistemas de LMS. Estas integraciones se proyectan para fases posteriores del proyecto.
- Los módulos relacionados con gestión egresados o bienestar estudiantil, infraestructura, mantenimiento, nómina no serán abordados en la primera versión del sistema, aunque se prevé su desarrollo en versiones futuras.
- La migración de datos desde el sistema **SIC** se realizará de forma selectiva, priorizando la información académica, administrativa y financiera esencial. No se garantiza la migración completa de todos los registros históricos.
- El sistema no incluirá funciones de analítica avanzada ni inteligencia artificial en esta fase de implementación. Se centrará en reportes operativos y funcionales predefinidos.
- El soporte multilinguaje no será considerado en esta versión inicial, ya que el enfoque institucional está centrado en el idioma español.

Estas restricciones son necesarias para garantizar la viabilidad técnica y administrativa del proyecto, permitiendo enfocar los recursos disponibles en el cumplimiento de los objetivos establecidos para la primera etapa del sistema.

#### **1.3.2.2. Restricciones de tiempo.**

El equipo de desarrollo ha establecido una duración estimada de 9 meses a partir de la fecha de inicio del proyecto (24/09/2024), lo cual representa una restricción temporal crítica para la ejecución de este.

Este límite de tiempo impone las siguientes consideraciones:

Las funcionalidades incluidas en esta primera fase deben ser priorizadas con base en su impacto y necesidad operativa, postergando aquellas que puedan desarrollarse en fases futuras sin afectar la funcionalidad esencial del sistema.

Las etapas de análisis, diseño, desarrollo, pruebas, migración y capacitación deben estar calendarizadas de forma estricta, permitiendo un seguimiento constante del avance frente al cronograma.

La ejecución simultánea de algunos procesos (como desarrollo y pruebas internas) será necesaria para optimizar tiempos sin comprometer la calidad.

Cualquier cambio significativo en el alcance o en las condiciones del entorno deberá ser evaluado cuidadosamente para no comprometer los plazos establecidos.

El cumplimiento de esta restricción temporal es clave para la adopción institucional oportuna del sistema **MARIA**, así como para responder a las necesidades urgentes identificadas en el diagnóstico inicial.

#### **1.3.2.3. Restricciones de recursos.**

El proyecto **MARIA** enfrenta diversas restricciones de recursos que deben ser consideradas durante su planificación y ejecución. Estas limitaciones están relacionadas con aspectos humanos, técnicos y financieros que condicionan el ritmo y alcance del desarrollo:

**Personal técnico limitado:** El equipo de desarrollo está conformado por un grupo reducido de profesionales que asumen roles clave en el desarrollo del sistema **MARIA**:

**Jhon Anthony Uribe Ahumada**, Desarrollador de Software, responsable de la arquitectura del backend y la implementación lógica del sistema.

**Miguel Alberto Medrano Perez**, Desarrollador Full Stack en formación, encargado del diseño y construcción de la interfaz de usuario y el soporte a la integración con el backend.

**Juan Manuel Arenas Perdomo**, Administrador de Base de Datos, encargado del diseño, creación, mantenimiento y aseguramiento de la integridad y seguridad de los datos institucionales.

**Disponibilidad parcial del equipo:** Algunos miembros del equipo comparten responsabilidades con otras funciones institucionales, lo cual puede afectar la dedicación exclusiva al proyecto en ciertos momentos.

La disponibilidad parcial y la distribución funcional del equipo requiere una gestión eficiente del tiempo, asignación estratégica de tareas y coordinación continua para garantizar el cumplimiento de los objetivos en los tiempos establecidos.

**Recursos financieros ajustados:** El financiamiento del proyecto proviene de recursos institucionales propios, lo cual restringe la posibilidad de contratar personal adicional, adquirir servicios externos o ampliar la infraestructura tecnológica más allá de lo estrictamente necesario.

**Infraestructura tecnológica heredada:** El sistema se está desarrollando sobre servidores existentes, con ciertas limitaciones de capacidad y escalabilidad, lo cual puede afectar el despliegue de funcionalidades avanzadas en etapas tempranas.

**Herramientas y licencias:** Se prioriza el uso de tecnologías de código abierto y herramientas gratuitas, lo que limita el acceso a ciertos recursos de desarrollo empresarial o plataformas especializadas.

Estas restricciones obligan al equipo a adoptar una estrategia de desarrollo progresiva y bien focalizada, maximizando el uso eficiente de los recursos disponibles para cumplir con los objetivos del proyecto dentro de los plazos establecidos.

#### **1.4. Entregables del proyecto.**

##### **1.4.1. Estructura de desglose de trabajo (EDT/WBS).**

La Estructura de Desglose del Trabajo (EDT) para el proyecto **MARIA** permite visualizar de forma jerárquica y organizada los componentes y tareas que conforman el proyecto, facilitando su planificación, seguimiento y control.

A continuación, se presenta un resumen estructurado en niveles de trabajo:

#### **1. Planificación del Proyecto**

1.1. Análisis de requerimientos técnicos y funcionales

1.2. Definición del modelo de datos

1.3. Cronograma y asignación de responsabilidades

#### **2. Desarrollo del Sistema MARIA**

2.1. Diseño y configuración de la base de datos

2.2. Desarrollo del backend (lógica y control de acceso)

2.3. Desarrollo del frontend (interfaz de usuario)

2.4. Integración entre módulos

2.5. Pruebas internas (unitarias y de integración)

### **3. Migración de Datos**

3.1. Análisis de datos históricos del sistema SIC

3.2. Limpieza y transformación de datos

3.3. Cargue en el nuevo sistema

### **4. Implementación y Puesta en Marcha**

4.1. Despliegue del sistema en entorno de producción

4.2. Pruebas de aceptación con usuarios finales

4.3. Ajustes finales y corrección de incidencias

### **5. Capacitación y Documentación**

5.1. Manual de usuario y guías técnicas

5.2. Capacitación a usuarios clave y personal técnico

5.3. Entrega de documentación del sistema

### **6. Cierre del Proyecto (versión 1)**

6.1. Informe final de resultados

6.2. Evaluación de cumplimiento de objetivos

6.3. Plan de mejoras y proyección a nuevas fases

Esta EDT servirá de base para el seguimiento del avance del proyecto y la gestión de recursos, riesgos y tiempos durante todo el ciclo de vida de **MARIA**.

## **1.5. Cronograma y resumen del presupuesto.**

### **1.5.1. Cronograma.**

Fecha estimada de finalización: 15 de julio de 2025.

**CRONOGRAMA PENDIENTE**

### 1.5.2. Presupuesto.

El proyecto **MARIA** se ejecuta con recursos institucionales propios, optimizando al máximo el talento humano interno y herramientas de código abierto. A continuación, se presenta un resumen de los costos estimados por fase, teniendo en cuenta tiempo de dedicación, infraestructura, soporte técnico y otros recursos necesarios:

ANALISIS DE PRESUPUESTO		
Fase / Actividad	Costo estimado (COP)	Observaciones
Planificación y análisis de requerimientos		Honorarios, reuniones, análisis funcional y técnico
Diseño de base de datos y arquitectura		Modelado, documentación técnica y estructura relacional
Desarrollo backend		Programación de lógica, control de usuarios y roles
Desarrollo frontend		Interfaces gráficas, usabilidad, conexión con backend
Integración de módulos y pruebas internas		QA, revisión cruzada, testing de módulos
Migración de datos desde SIC		Limpieza, depuración, importación y validación de registros
Pruebas de aceptación y puesta en marcha		Test con usuarios, ajustes finales
Capacitación y documentación		Manuales, talleres, formación institucional
Infraestructura tecnológica (uso actual)		Hosting, respaldo, dominios, certificados digitales
Contingencias y soporte		Incidentes técnicos, mantenimientos menores

**Total, estimado del proyecto: \$....**

Este presupuesto está sujeto a revisión y actualización según las necesidades institucionales y la evolución del proyecto.



2. Documentos de referencia.

2.1. Referencias.

A continuación, se listan los documentos y fuentes de referencia utilizadas como soporte para la planificación, diseño e implementación del sistema MARIA. Estas referencias comprenden estándares técnicos, documentación institucional y bibliografía especializada que fundamenta las decisiones tomadas en el proyecto:

Referencia	Título	Autor	Año
[R1]	Informe de Auditoría Forense – Base de Datos BD_SYS_SISEDUCATIVO	Alexis Cadrasco García	2024
[R2]	Lineamientos estratégicos de digitalización de INCATEC (2024-2026)	INCATEC	2024
[R3]	Ingeniería de Software: Un enfoque práctico (8.ª ed.). McGraw-Hill. Guía de referencia sobre procesos de desarrollo, análisis de riesgos y aseguramiento de calidad.	Pressman, R. S., & Maxim, B. R.	2015
[R3]	IEEE Recommended Practice for Software Requirements Specifications. Estándar para la redacción estructurada de especificaciones de requerimientos de software.	IEEE Std 830-1998	1998

2.2. Estándares.

Estándar	Título	Autor	Año
[E1]	IEEE Std 830-1998 - Recommended Practice for Software Requirements Specifications	IEEE (Institute of Electrical and Electronics Engineers)	1998
[E2]	ISO/IEC 25010:2011 - Systems and software Quality Requirements and Evaluation (SQuaRE)	ISO/IEC Joint Technical Committee	2011
[E3]	OWASP Top 10 - Top 10 Web Application Security Risks	Open Web Application Security Project	2023
[E4]	ISO/IEC 12207 - Systems and Software Engineering – Software Life Cycle Processes	International Organization for Standardization	2008

3. Definiciones y acrónimos.

3.1. Definiciones.

Sistema MARIA

Plataforma de gestión académica, administrativa y financiera desarrollada para INCATEC.

SIC

Sistema de Información de Control anteriormente utilizado en INCATEC, reemplazado por MARIA.

Base de datos relacional:

Modelo estructurado de almacenamiento de datos que utiliza tablas relacionadas mediante claves primarias y foráneas.

**Integridad referencial:**

Propiedad de la base de datos que garantiza que las relaciones entre tablas permanezcan consistentes.

**Backend:**

Parte del sistema que gestiona la lógica de negocio, acceso a datos y control de seguridad.

**Frontend:**

Interfaz visual con la que interactúan los usuarios para acceder a las funcionalidades del sistema.

**Full Stack:**

Perfil de desarrollador que trabaja tanto en el frontend como en el backend de una aplicación.

**Control de accesos:**

Mecanismo que regula qué usuarios pueden acceder a qué funcionalidades dentro del sistema.

**Pruebas de aceptación:**

Evaluaciones realizadas por los usuarios finales para validar que el sistema cumple con sus necesidades.

**Migración de datos:**

**Capacitación:**

Proceso de formación para asegurar que los usuarios conozcan y manejen adecuadamente el sistema.

**4. Resumen del trabajo requerido.**

El desarrollo del sistema **MARIA** implica la creación de una solución tecnológica integral para la gestión académica, administrativa y financiera del Instituto Técnico de Administración y Salud **INCATEC**. Este trabajo surge como respuesta a las limitaciones del sistema anterior (SIC) y se basa en una serie de suposiciones, restricciones y objetivos institucionales claramente definidos.

**Suposiciones y consideraciones:**

- El equipo técnico cuenta con los conocimientos necesarios en desarrollo de software, modelado de bases de datos y despliegue en entornos de producción.
- **INCATEC** proporcionará los recursos técnicos y logísticos mínimos necesarios para ejecutar el proyecto dentro del entorno institucional.
- Los datos del sistema anterior (SIC) estarán disponibles y podrán ser transformados para su migración al nuevo sistema.
- Los usuarios clave participarán en procesos de validación, pruebas y retroalimentación del sistema durante su desarrollo.

### Restricciones del proyecto:

- **Cronograma:** El desarrollo del sistema debe concluir a más tardar en julio de 2025, con una duración total de aproximadamente 9 meses.
- **Presupuesto:** El proyecto cuenta con una asignación aproximada de \$..., financiado con recursos institucionales.
- **Recursos humanos:** El equipo está conformado por tres miembros principales, cada uno con roles específicos (desarrollo, base de datos, frontend).
- **Tecnología:** Se emplearán tecnologías de código abierto como Django, PostgreSQL, HTML, CSS y JavaScript, con preferencia por soluciones escalables y de bajo costo.

### Objetivos de desarrollo:

- Crear un sistema modular y escalable que permita gestionar procesos académicos, administrativos y financieros con integridad y seguridad.
- Garantizar una interfaz de usuario amigable, intuitiva y eficiente para todos los perfiles institucionales.
- Implementar un modelo de base de datos normalizado que permita trazabilidad total y reportes confiables.
- Migrar de forma controlada los datos históricos del sistema SIC al nuevo entorno.

### Productos para entregar:

- Sistema **MARIA** operativo, con backend y frontend funcionales.
- Base de datos relacional optimizada y segura.
- Panel de gestión de usuarios, roles y permisos.
- Módulos funcionales para matrícula, pagos, seguimiento académico y reportes.

- Herramientas de auditoría, logs y seguridad.
- Documentación técnica y manuales de usuario.
- Plan de capacitación para el personal institucional.
- Informe final de cierre y proyección de mejora continua.

**Métodos de validación:**

- Pruebas internas de integración y funcionalidad.
- Pruebas de aceptación con usuarios institucionales.
- Evaluación de cumplimiento de requerimientos.
- Verificación contra estándares de calidad del software (ISO/IEC 25010).
- Registro de incidencias y retroalimentación de usuarios en entorno controlado.

## 5. Organización y control del proyecto.

El proyecto **MARIA** cuenta con una organización técnica definida que permite una gestión clara y eficiente de los recursos humanos involucrados. El equipo de desarrollo está conformado por profesionales con roles específicos que aseguran la ejecución de cada fase del proyecto, desde la planificación hasta la puesta en marcha.

Nombre y Apellidos	Rol en el Proyecto	Responsabilidades
<b>MARIA ANTONIA DIAZ MENDOZA</b>	PRODUCT MANAGER	impulsa el desarrollo y el éxito de MARIA, definiendo su estrategia, hoja de ruta y características, actuando como puente entre los clientes, las partes interesadas y los equipos de desarrollo.
<b>JHON ANTHONY URIBE AHUMADA</b>	DESARROLLADOR FULL-STACK	Participa en el desarrollo tanto del backend como del frontend, garantizando la integración funcional del sistema. conectividad con la base de datos y diseño de interfaces.
<b>MIGUEL ALBERTO MEDRANO PEREZ</b>	DESARROLLADOR FULL-STACK	Desarrolla interfaces gráficas funcionales y adaptativas, así como la lógica necesaria para la interacción eficiente del sistema. Apoya la integración de módulos y la optimización del flujo de usuario.
<b>JUAN MANUEL ARENAS PERDOMO</b>	ADMINISTRADOR DE BASE DE DATOS	Diseña, crea y mantiene la base de datos del sistema <b>MARIA</b> , asegurando su integridad, seguridad y rendimiento. Es responsable del soporte en migraciones, respaldos y configuración avanzada de datos, Implementa la lógica del negocio.

La coordinación del equipo se lleva a cabo mediante reuniones técnicas, seguimiento de hitos, y uso de herramientas colaborativas de gestión de proyectos. Esta estructura organizacional permite una comunicación efectiva, asignación clara de responsabilidades y cumplimiento de los objetivos del sistema MARIA.

## 6. Proceso de desarrollo de software.

### 6.1. Entorno del proceso de software.

Para el desarrollo del sistema **MARIA**, se ha seleccionado el modelo de ciclo de vida Incremental Iterativo, debido a su idoneidad para proyectos de mediana a alta complejidad, como es el caso de un sistema integral académico-administrativo. Este modelo permite una evolución progresiva del software a través de iteraciones, facilitando la incorporación de retroalimentación continua y el ajuste a requerimientos cambiantes.

### Fases del proceso de desarrollo y sus actividades

FASE	ACTIVIDADES PRINCIPALES
1. Recolección de requerimientos	Identificación de necesidades institucionales, entrevistas, revisión del sistema SIC.
2. Análisis y diseño	Modelado de base de datos, arquitectura del sistema, diagramas UML, definición de roles.
3. Desarrollo incremental	Programación por módulos funcionales: backend (lógica), frontend (interfaz), integraciones.
4. Pruebas internas	Pruebas unitarias, pruebas de integración, revisión de flujos y validación por parte del equipo técnico.
5. Pruebas de aceptación	Validación del sistema por usuarios institucionales, reporte de incidencias, correcciones.
6. Puesta en producción	Despliegue en entorno real, configuraciones finales, verificación de desempeño.
7. Capacitación y cierre	Formación a usuarios clave, entrega de manuales, evaluación del cumplimiento de objetivos.

### Estrategia de desarrollo y herramientas utilizadas

**Estrategia metodológica:** Enfoque ágil y modular, permitiendo entregas parciales y revisiones periódicas.

**Lenguaje de programación:** Python (Django), JavaScript, HTML5, CSS3.

**Gestión de base de datos:** PostgreSQL.

**Control de versiones:** Git, alojado en repositorios privados.

**Modelado y diseño:** UML con herramientas como Draw.io y PlantUML.

**Pruebas:** Unitarias (Pytest, Django TestCase).

De integración y aceptación (manuales y automatizadas).

**Herramientas colaborativas:** Trello para gestión de tareas, Google Drive para documentación, y Reuniones semanales para seguimiento.

Este entorno garantiza un desarrollo organizado, flexible, seguro y **transparente**, con espacios definidos para la mejora continua, revisión por pares y evaluación constante de calidad.

## 6.2. Diseño de software.

### 6.2.1. Definición de roles o actores.

A continuación, se identifican los roles o actores del sistema MARIA, entendidos como entidades que interactúan con el sistema desde el contexto de las necesidades institucionales:

Nombre del Rol	Descripción
<b>Colaborador</b>	Es el personal administrativo institucional que necesita gestionar procesos internos de matrícula, recaudo, control académico y generación de reportes. Su necesidad surge de la falta de trazabilidad, errores frecuentes y duplicación de tareas presentes en el sistema anterior.
<b>Estudiante</b>	Es el usuario beneficiario de los servicios académicos de INCATEC, cuya experiencia institucional se ha visto afectada por información errónea o demoras en trámites. Requiere acceder a información clara, oportuna y confiable.
<b>Docente</b>	Actor encargado de procesos de formación, seguimiento académico y evaluación de estudiantes. En el sistema anterior, carecía de acceso adecuado a la información y debía depender de procesos manuales.
<b>Administrador del Sistema</b>	Usuario con control total sobre la configuración técnica y operativa del sistema. Surge como figura necesaria para mantener la integridad del sistema, gestionar roles y supervisar su funcionamiento general.
<b>Empresa Coformadora</b>	Representa a las entidades externas que colaboran en la formación práctica de los estudiantes. Requieren acceso controlado a información académica para seguimiento, evaluación y validación de procesos en formación dual.
<b>Auditor Interno</b>	Encargado de la verificación, control y evaluación del cumplimiento normativo y de los procesos institucionales. Requiere acceso a registros confiables para análisis y toma de decisiones correctivas.

#### 6.2.2. Análisis de requerimientos de software.

Para el desarrollo del sistema **MARIA**, se utilizó un enfoque mixto de recolección de requerimientos que permitió identificar de forma precisa las necesidades funcionales y no funcionales del software. Esta etapa fue crucial para garantizar que el sistema respondiera adecuadamente a las condiciones reales de la institución y superara las limitaciones del sistema anterior (SIC).

**Técnica utilizada:** Recolección participativa por medio de entrevistas estructuradas y análisis documental

##### 1. Entrevistas estructuradas

Se realizaron entrevistas con usuarios clave de las áreas académica, administrativa y financiera de INCATEC, incluyendo colaboradores, docentes y personal directivo. Estas entrevistas siguieron una guía prediseñada basada en el análisis de procesos críticos institucionales.

##### 2. Análisis documental

Se revisaron documentos institucionales y operativos como:

Informes de auditoría forense (SIC).

Manuales de procedimiento internos.

Reportes operacionales del sistema anterior.

Requerimientos funcionales históricos.

Normativas del Subsistema de Formación para el Trabajo.

### 3. Observación directa de procesos

Se llevaron a cabo sesiones de observación de actividades institucionales reales, como inscripción de estudiantes, matrículas, recaudos, y generación de certificados, para validar las necesidades tecnológicas desde la operación misma.

### 4. Sesiones de validación

Posteriormente, se organizaron mesas de trabajo para validar los requerimientos identificados con los mismos actores entrevistados. Esto permitió priorizar funcionalidades y confirmar expectativas sobre el sistema.

### Resultados

Esta combinación de técnicas permitió recopilar una lista robusta de requerimientos clasificados por módulo, tipo de usuario y criticidad. Los detalles completos de los requerimientos se anexan como parte del documento técnico entregable del proyecto.

#### 6.2.3. Requerimientos funcionales.

Liste en la siguiente tabla los requerimientos del sistema, es decir lo que el sistema va a hacer y los requerimientos del usuario, es decir lo que podrá hacer con el sistema.

Referencia	Requerimiento
RF01	
RF02	
RF03	

Nota: teniendo como referencia la norma ISO/IEC/IEEE 29148 de 2011 Ingeniería del Software Proceso del Ciclo de Vida Ingeniería de Requerimientos, deberá utilizar la siguiente estructura recomendada para su escritura:

[Condición] [Sujeto] [Acción] [Objeto] [Restricción]

[Condición] [Acción o restricción] [Valor]

[Sujeto] [Acción] [Valor]

Nota: Se puede utilizar el tipo de requerimientos mediante Historias de Usuario para los primeros semestres.

Plantilla:

Yo como <Rol del Usuario> deseo <Objetivo/Funcionalidad> para <Resultado/Objetivo>

#### 6.2.4. Requerimientos no funcionales.

Elabore una lista de las restricciones o servicios ofrecidos por el software relacionada con aspectos del producto de software, la organización y aspectos externos. Debe generar mínimo dos requerimientos no funcionales para cada uno de los tipos establecidos (usabilidad, eficiencia, seguridad, aspectos éticos, aspectos regulatorios y legislativos).



Referencia	Tipo	Descripción
RNF-01		
RNF-02		
RNF-03		

### 6.3. Modelado de software.

#### 6.3.1. Diagramas de Casos de uso.

Elabore cada uno de los diagramas de casos de uso que cubra todos los requerimientos funcionales descritos en el numeral 6.2.2. El diagrama debe estar basado en la notación UML y en función de los diferentes roles del sistema.

Para cada diagrama de casos de uso deberán hacer una descripción utilizando la siguiente plantilla:

Caso de Uso	Nombre del Caso de Uso		
Versión	V1	Referencia	CU#
Autor	Nombre de quien lo describe	Fecha	D/M/A
Actores	Liste los actores que intervienen en el caso de uso.		
Referencias	Liste la referencia de los requerimientos funcionales del punto 6.2.2. que intervienen en el caso de uso.		
Precondiciones	Describa las condiciones necesarias que se deben cumplir para poder ejecutar el caso de uso.		
Pos condición	Describe el estado del sistema cuando se cumple con éxito el caso de uso.		
Flujo Normal (Deberá escribir en cada línea paso a paso la interacción entre lo que hace el actor y el sistema durante la ejecución del caso de uso)			
Paso	Actor	Descripción	
Flujos alternos	Describa lo que pasa en la eventualidad que no se cumpla con el flujo normal de interacción.		
Excepciones	Escriba las excepciones que pueden ocurrir durante el caso de uso.		

#### 6.3.2. Diagrama de secuencia.

Coloque aquí un diagrama de secuencia por cada diagrama de caso de uso, aplicando la notación UML mediante el uso de una herramienta digital.

#### 6.3.3. Diagrama de actividades.

Coloque aquí un diagrama de actividad de cada uno de los procesos del software más representativos, aplicando la notación UML mediante el uso de una herramienta digital.

#### 6.3.4. Diagramas de clases.

Coloque aquí el diagrama de clases con su notación respectiva la cual incluye, sus atributos y métodos, así como las relaciones y cardinalidad respectiva.

### 6.4. Diseño de la interfaz.

#### 6.4.1. Arquitectura de información.

Realice el mapa de la estructura de información (Mapa del Sitio Web) con base en los diagramas de diseño de software.

Con base en el mapa, explique cuáles son los elementos de navegación principal y describa que tipo de información encontrará el usuario en cada uno de ellos.

#### **6.4.2. Wireframe.**

Mediante el uso de una herramienta tecnológica, desarrollo y coloque en este punto los wireframe más representativos del producto de software, con su respectiva descripción.

### **6.5. Diseño de datos.**

#### **6.5.1. Método de acceso a la base de datos.**

De acuerdo con el sistema de gestión de base de datos utilizada, explique la forma de tener acceso a la base de datos para realizar las transacciones.

#### **6.5.2. Definición de las entidades de datos.**

Enuncie y describa cada una de las entidades que pertenecen al diseño de la base de datos, mediante el uso de la siguiente tabla.

Entidad	Descripción

#### **6.5.3. Diagrama entidad relación.**

Coloque en este punto el diagrama realizado mediante el uso de una herramienta digital en formato imagen.

#### **6.5.4. Diccionario de datos.**

El diccionario de datos hace referencia a las características de los datos utilizados por el sistema. En este punto deberá colocar cada una de las entidades con sus respectivos campos entre los cuales están: nombre, tipo de datos, longitud, valores específicos si es obligatorio y una descripción. De igual forma se debe especificar la integridad a través de las llaves primarias y foráneas.

#### **6.5.5. Restricciones de seguridad e integridad.**

Describa los aspectos de integridad utilizados en las relaciones entre las diferentes entidades del modelo de datos.

### **6.6. Diseño de la arquitectura de Software.**

#### **6.6.1. Patrón de diseño.**

Describa brevemente los métodos para definir y describir la arquitectura del sistema de software.

#### **6.6.2. Diagrama de componentes.**

Realice y coloque el diagrama de componentes que se utiliza para modelar los aspectos físicos del sistema, mediante la unión de componentes, interfaces y relaciones.

#### **6.6.3. Diagrama de despliegue.**

Realice y coloque el diagrama que se utiliza para para modelar la configuración de funcionamiento del sistema incluyendo el software y el hardware.

## **6.7. Construcción e implementación de software.**

### **6.7.1. Integración del sistema de software.**

Describa los elementos de software como unidades, componentes o módulos que se integrarán en un esfuerzo de desarrollo coordinado para cumplir con los objetivos de costo, programación y rendimiento del sistema de software.

### **6.7.2. Verificación y validación del sistema de software.**

Explique cómo los requisitos, incluidos los no técnicos, como la seguridad y la protección, se verificarán y validarán para el sistema de software.

Tenga en cuenta que existen pruebas de tipo unitario, de integración, de funcionalidad y de usuario final.

### **6.7.3. Integración de software en hardware.**

Describa cómo los elementos de Software que se integrarán con los elementos de Hardware relevantes para realizar un sistema integral.

### **6.7.4. Transición de software y gestión de lanzamientos.**

Describa las actividades de control de versiones de software, distribución de software y transición de software entre versiones.

### **6.7.5. Mantenimiento de software.**

Describa el enfoque para mejorar el software y corregir los defectos una vez que se haya lanzado el software.

### **6.7.6. Gestión de defectos y problemas de software.**

Explique cómo los defectos de software y los problemas técnicos se identificarán, registran y resolverán (acción correctiva).

Describa las actividades para el aseguramiento de la calidad y la mejora de los procesos.

## **7. Aspectos éticos del proyecto.**

### **7.1. Consideraciones éticas en la planeación del proyecto.**

En esta sección, se abordan de manera resumida las consideraciones éticas clave que guiarán la planificación del proyecto, asegurando la integridad y responsabilidad en todas las fases.

### **7.2. Consideraciones éticas en el desarrollo del proyecto.**

Aquí se presentan de forma concisa las consideraciones éticas fundamentales durante la ejecución del proyecto, destacando principios y prácticas que promueven la conducta ética en todas las actividades.

### **7.3. Consideraciones éticas en la etapa de producción y mantenimiento de la solución aportada por el producto desarrollado por el proyecto.**

En esta sección se resumen las consideraciones éticas esenciales que deben observarse durante la producción y el mantenimiento de la solución ofrecida por el producto desarrollado, garantizando prácticas éticas a lo largo del ciclo de vida del proyecto.

## **8. Conclusiones.**

### **8.1. Conclusiones del estudiante.**

Reflexión del estudiante en la que explica que aprendió durante el desarrollo del proyecto.

### **8.2. Conclusión del tutor en sitio del proyecto.**

Opinión del tutor sobre los resultados logrados con el proyecto. Párrafo redactado por el tutor

### **8.3. Lecciones aprendidas**

Recomendaciones para el desarrollo de nuevos proyectos, que evitan o eliminan problemas que se resolvieron en el desarrollo del proyecto actual.

**Anexos.**

Incluya cualquier documento o gráfico que considere necesario para dar soporte a alguno de los puntos descritos en el documento, recuerde hacer referencia al anexo.