

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



## Sistema de Analítica en Prevención de Riesgos

Planificación - Casino Luckia Arica S.A

**Autores:** Andrew Campos Seguel  
Gustavo Ríos Álvarez

**Asignatura:** Proyecto IV 2025

**Profesor:** Diego Aracena Pizarro

**Cliente:** Edgardo Flores Alarcón



# Índice

1. Definición y justificación del proyecto	5
1.1. Contexto	5
1.2. Problemática	5
1.3. Solución	5
2. Alcance y objetivos	6
2.1. Objetivo General	6
2.2. Objetivos Específicos	6
2.3. Carta Gantt	7
3. Requerimientos del Proyecto	8
3.1. Funcionales	8
3.2. No Funcionales	10
4. Metodología	11
5. Modelos del sistema	12
5.1. Diagrama de contexto	12
5.2. Análisis del Diagrama de contexto	13
5.3. Modelo de caso de uso	14
5.4. Análisis del Diagrama de Casos de Uso	15
8.1. Modelo Entidad-Relación de la base de datos	16
8.2. Modelo de Tablas Relacionales de la base de datos	17
8.3. Modelo de proceso de negocio	18
9. Visuales del Proyecto	1
10. Herramientas para el desarrollo	1
10.1. Visual Studio Code	1
10.2. Angular Material UI	1
10.3. GitHub	1
10.4. Figma	1
10.5. Vercel	1
10.6. Tailwind CSS	1
11. Framework	1
11.1. Angular	1
11.2. NestJs	1
12. Herramientas de almacenamiento	1
12.1. Google Drive	1
12.2. Free SQL Database	1
13. Modelado de Datos	1
13.1. Base de datos	1
14. Descripción de Modelos de análisis	1
15. Conclusión	1
16. Referencias	1



## Índice de figuras

Figura 1: Carta Gantt	7
Figura 2: Metodología Prototipo descartable	11
Figura 3: Diagrama de contexto	12
Figura 4: Diagrama de caso de uso	14
Figura 6: Modelo de Tablas Relacionales	17
Figura 7: modelo de negocio registrar incidente	18
Figura 8: modelo de negocio generar Dashboard e Informe	18
Figura 8: Modo nocturno vista Registro de incidentes	19
Figura 11: Modo Diurno vista Registro de incidente	20
Figura 12: Vista Dashboard	21
Figura 13: Diagrama de modelo de análisis	26



## Índice de Tablas

Tabla 1: Objetivos Específicos	6
Tabla 2: Requerimientos funcionales	8
Tabla 3: Requerimientos no funcionales	10
Tabla 4: Análisis diagrama de contexto	13
Tabla 5: Descripción casos de uso	15

# 1. Definición y justificación del proyecto

## 1.1. Contexto

La información sobre seguridad y salud en el trabajo, así como los reportes de siniestros (incidentes, accidentes con o sin tiempo perdido, cuasi accidentes, horas trabajadas, días perdidos, capacitaciones, inspecciones, uso de EPP, mantenimientos y observaciones de seguridad) del Casino Luckia Arica S.A se registran en un sistema propio.

Estos registros se consolidan en reportes individuales, informes mensuales y anuales en formato PDF, elaborados manualmente. Cada informe incluye datos sobre los siniestros, las personas afectadas, la ubicación, las fechas y representaciones gráficas. El objetivo de este proceso es analizar los eventos ocurridos y desarrollar campañas y capacitaciones que contribuyan a reducir su frecuencia.

## 1.2. Problemática

Estos registros se consolidan manualmente informes mensuales y anuales en formato PDF, lo que genera una alta carga operativa y una probabilidad considerable de errores o pérdida de datos. La ausencia de un sistema automatizado de análisis impide transformar los datos recopilados en información útil para la toma de decisiones, dificultando la detección de tendencias, la evaluación de causas raíz y la implementación de medidas preventivas efectivas.

Como resultado, los informes generados presentan limitaciones en cuanto a su precisión, oportunidad y valor predictivo, reduciendo la efectividad de las campañas y capacitaciones diseñadas para disminuir la frecuencia de siniestros. Esta situación afecta la integridad de la información, el control de indicadores de desempeño en seguridad laboral y la capacidad del área de prevención para desarrollar estrategias basadas en evidencia.

## 1.3. Solución

Diseñar e implementar un sistema integral y automatizado de gestión y análisis de seguridad como de salud en el trabajo, que permita registrar incidentes, accidentes y actividades preventivas, además, debe generar informes precisos, oportunos y con valor predictivo, así como apoyar la toma de decisiones basada en datos para planificar campañas preventivas y capacitaciones orientadas a reducir la frecuencia de siniestros.

## 2. Alcance y objetivos

### 2.1. Objetivo General

Diseñar e implementar un sistema integral y automatizado de gestión y análisis de seguridad, que permita registrar incidentes, accidentes y actividades preventivas, además, debe generar informes precisos, oportunos y con valor predictivo, así como apoyar la toma de decisiones basada en datos para planificar campañas preventivas y capacitaciones orientadas a reducir la frecuencia de siniestros.

### 2.2. Objetivos Específicos

Tabla 1: Objetivos Específicos

Objetivos específicos	Título	Descripción
OE-01	Crear y actualizar carta gantt	Elaborar y mantener actualizada la planificación del proyecto mediante una carta Gantt que permita gestionar los plazos, responsables y entregables.
OE-02	Establecer Requerimientos del sistema	Realizar el levantamiento de información relevante a través de entrevistas, revisión documental y análisis del proceso actual de gestión de riesgos.
OE-03	Diseñar Modelos del sistema	Diseñar el modelo de datos del sistema, considerando entidades, relaciones, atributos y restricciones, como base para la posterior implementación de la base de da
OE-04	Diseñar e implementar una base de datos	partir del modelo diseñado, creando las tablas, relaciones, claves primarias/foráneas y estructuras necesarias para almacenar y gestionar la información del sistema.
OE-05	Diseñar diagramas y modelos del prototipo	Desarrollar modelos y diagramas del sistema, tales como: diagrama de contexto, flujos de datos, casos de uso, arquitectura general y prototipo de interfaz, para asegurar la coherencia entre requisitos y solución propuesta
OE-06	Presentar avances al cliente	Validar los avances del proyecto mediante revisiones con el usuario clave, aplicando correcciones sobre los prototipos, modelos o base de datos según sea necesario.

OE-07	Documentar procesos y resultados	Documentar detalladamente todas las etapas del proyecto, diagnóstico, análisis, diseño, implementación y evaluación
-------	----------------------------------	---

## 2.3. Carta Gantt

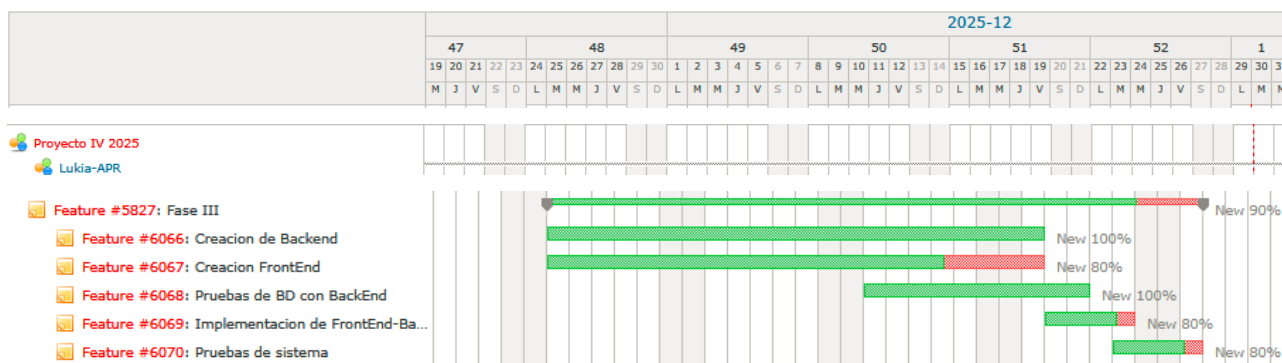


Figura 1: Carta Gantt

## 3. Requerimientos del Proyecto

### 3.1. Funcionales

Tabla 2: Requerimientos funcionales

CÓDIGO	NOMBRE	DESCRIPCIÓN	PRIORIDAD
LUAPR-01	Registro de incidentes y accidentes	El sistema debe permitir registrar incidentes, accidentes y cuasi accidentes; indicando fecha y hora; tipo de suceso; área del trabajador; descripción; edad; y otros datos del afectado.	Alta
LUAPR-02	Gestión de actividades preventivas	El sistema debe permitir registrar capacitaciones, inspecciones, mantenimientos, observaciones de seguridad y uso de EPP.	Media
LUAPR-03	Generación automática de informes	El sistema debe generar informes mensuales y anuales en formato PDF con gráficos, tablas y resúmenes estadísticos de tendencias y reportes individuales.	Alta
LUAPR-04	Control de indicadores de SSL	El sistema debe calcular automáticamente indicadores de seguridad y salud laboral (tasa de frecuencia, gravedad, días perdidos, etc.)	Alta
LUAPR-05	Consulta y análisis histórico	El sistema debe permitir consultar y filtrar registros por Rango de Fecha, Tipo de evento, Área de trabajo y por Persona.	Alta
LUAPR-06	Dashboard interactivo	El sistema debe incluir un panel visual con gráficos e indicadores clave para la toma de decisiones.	Alta
LUAPR-07	Alertas y notificaciones	El sistema debe generar alertas automáticas ante repetición de eventos o indicadores fuera de rango.	Baja



<b>LUAPR-08</b>	Exportación y respaldo de datos	El sistema debe permitir exportar la información a formatos Excel o CSV y realizar copias de seguridad automáticas.	Alta
<b>LUAPR-9</b>	Carga masiva de datos históricos	Debe permitir importar registros previos desde archivos Excel o similares.	Media
<b>LUAPR-10</b>	Registro de Usuarios	Debe permitir guardar los datos del afectado en la BD para uso posteriores	Alta
<b>LUAPR-11</b>	Gestión de usuarios y roles	Debe permitir la administración de usuarios con diferentes permisos (administrador, prevencionista y encargado de reporte sustituto).	Media
<b>LUAPR-12</b>	Función de Administrador	El administrador tiene la función de gestionar roles para los usuarios en el sistema	Alta
<b>LUAPR-13</b>	Función de Prevencionista	El prevencionista tiene las funciones de gestionar reportes, generar informes y acceso al dashboard para realizar análisis a partir de los gráficos visualizados	Alta
<b>LUAPR-14</b>	Función de Encargado de reporte sustituto	El encargado de reporte sustituto tiene la función de generar reportes de incidentes.	Media

## 3.2. No Funcionales

Tabla 3: Requerimientos no funcionales

CÓDIGO	NOMBRE	DESCRIPCIÓN	PRIORIDAD
LUAPR-15	Usabilidad	La interfaz debe ser intuitiva, con navegación sencilla y accesible para usuarios no técnicos.	Alta
LUAPR-16	Seguridad de la información	Los datos deben almacenarse cifrados y protegidos por autenticación segura.	Alta
LUAPR-17	Integridad de datos	El sistema debe evitar duplicación o pérdida de registros durante las operaciones.	Alta
LUAPR-18	Rendimiento	Las consultas y generación de reportes deben ejecutarse en menos de 5 segundos promedio.	Media
LUAPR-19	Disponibilidad	El sistema debe estar disponible el 99.9% del tiempo.	Alta
LUAPR-20	Compatibilidad	El sistema debe ser accesible desde navegadores modernos y dispositivos móviles (diseño responsive).	Media
LUAPR-21	Mantenibilidad	Debe permitir actualizaciones y mejoras sin interrumpir el servicio ni comprometer la información.	Media
LUAPR-22	Respaldo y recuperación	Debe contar con copias automáticas de seguridad y procedimientos de restauración ante fallos.	Alta
LUAPR-23	Escalabilidad	El sistema debe soportar el crecimiento de datos y usuarios sin pérdida de rendimiento.	Media
LUAPR-24	Cumplimiento legal	Debe cumplir con la Ley 19.628 de Protección de la Vida Privada (modificada por la Ley 21.719), y normativas laborales vigentes.	Alta

## 4. Metodología

Para el desarrollo del sistema se ha elegido la metodología de cascada (o waterfall), esto debido a que se adapta a las características del proyecto enfocado en un desarrollo en secuencia ordenada por cada fase, además de esto integrar a este modelo lo que sería la retroalimentación controlada dándonos mayor flexibilidad al momento de solucionar errores con esto le daremos el nombre de “Prototipado Descartable”.

Donde tendríamos el clásico desarrollo en cascada, Análisis, Diseño , Desarrollo , Pruebas, Despliegue, agregando a esta secuencia ciclos controlados en cada fase con condiciones específicas dotando la secuencia de flexibilidad de retorno pudiendo retroceder una cantidad definida de etapas para la corrección de errores o modificación de algún apartado sin la necesidad de volver a empezar la secuencia desde el inicio, dando condiciones de retroceso con lo que se deberían de cumplir para poder realizar este retroceso sin afectar el proceso de desarrollo principal.

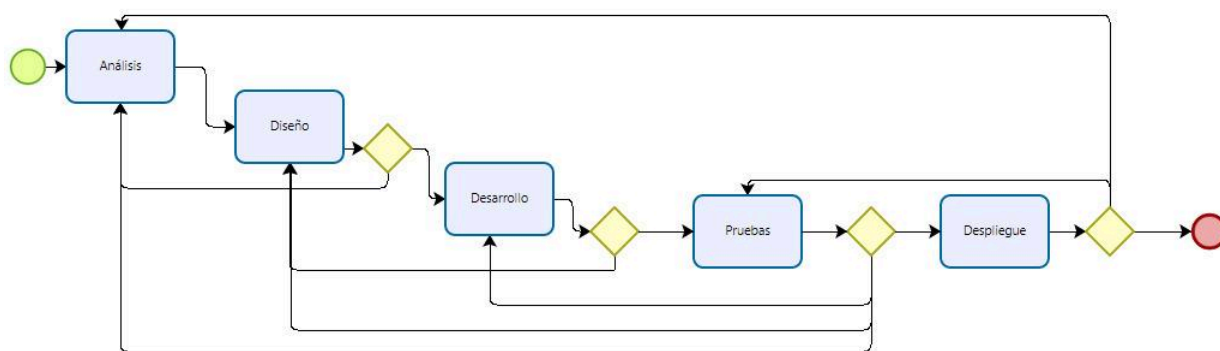


Figura 2: Metodología Prototipo descartable

La elección de esta metodología se justifica porque el proyecto cuenta con requerimientos bien definidos desde el inicio, lo que permite planificar y ejecutar las tareas de manera estructurada. Además, el modelo en cascada facilita la documentación formal de cada fase, el seguimiento del progreso y la verificación de entregables, aspectos esenciales para un entorno académico y para un cliente institucional como el *Casino Luckia Arica S.A.*

Asimismo, este enfoque ofrece claridad en la asignación de roles y responsabilidades dentro del equipo, asegurando un flujo de trabajo organizado y controlado. Su aplicación permitirá garantizar la trazabilidad, cumplimiento de plazos y calidad de los resultados, contribuyendo a la confiabilidad y éxito del sistema propuesto.

## 5. Modelos del sistema

### 5.1. Diagrama de contexto

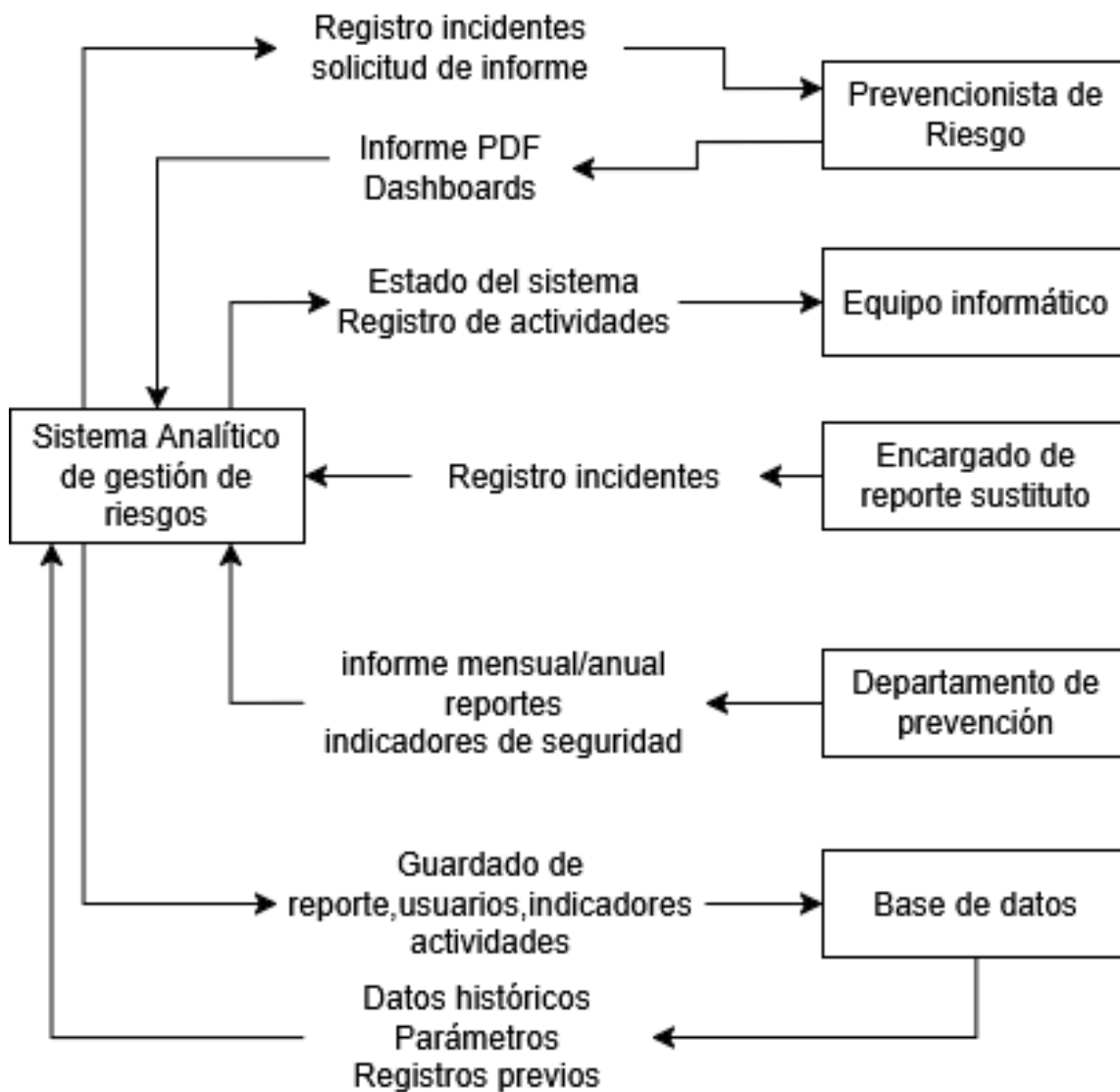


Figura 3: Diagrama de contexto

## 5.2. Análisis del Diagrama de contexto

Tabla 4: Análisis diagrama de contexto

Entidad Externa	Tipo de Flujo	Flujo de Datos	Rol en Sistema
<b>Prevencionista de Riesgo</b>	Entrada y Salida	<b>Reporte Incidentes</b> (Input) & <b>Genera Estadísticas</b> (Output)	Es el usuario operativo principal que ingresa datos y consume el análisis.
<b>Encargado Reportes Sust.</b>	Entrada	<b>Reporte Incidentes</b> (Input)	Es una fuente adicional de datos de incidentes, probablemente como apoyo al prevencionista.
<b>Equipo Informático</b>	Entrada	<b>Gestión Usuarios</b> (Input)	Es responsable de administrar las cuentas, accesos y permisos de los usuarios del sistema.
<b>Departamento Prevención</b>	Salida	<b>reportes mensuales anuales</b> (Output)	Es el receptor de los resultados consolidados y periódicos del sistema para la toma de decisiones.
<b>BD (Base de Datos)</b>	Salida	<b>Almacena Datos Incidentes</b> (Output)	Es el repositorio de almacenamiento persistente donde el sistema guarda todos los datos de incidentes.

### 5.3. Modelo de caso de uso

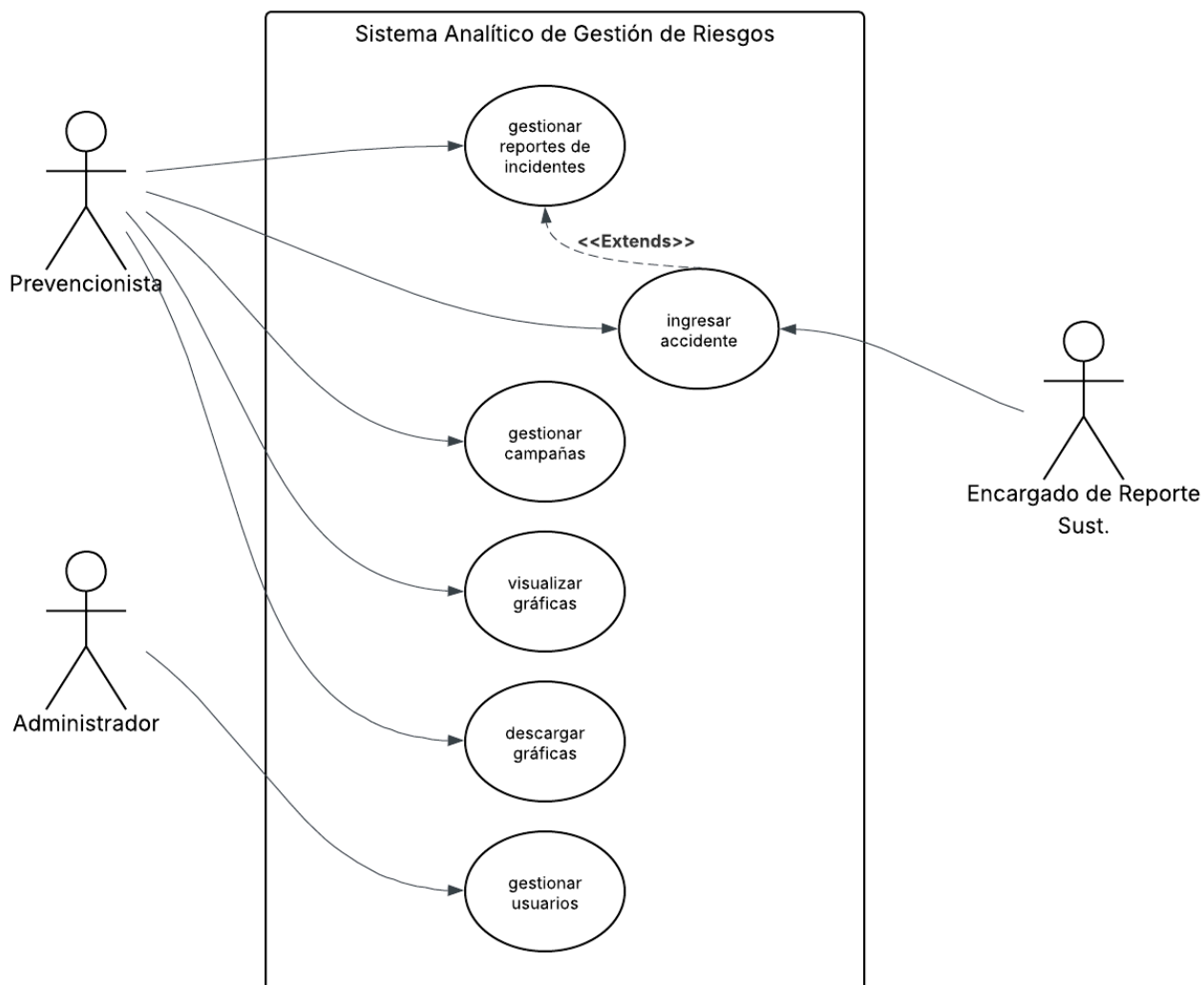


Figura 4: Diagrama de caso de uso

A partir del diagrama de casos de uso presentado, se entiende la funcionalidad y las interacciones clave dentro de un Sistema Analítico de Gestión de Riesgos.

## 5.4. Análisis del Diagrama de Casos de Uso

El diagrama define el alcance del sistema y cómo los diferentes actores (usuarios) interactúan con sus casos de uso (funcionalidades).

### 6. Sistema Analítico de Gestión de Riesgos

Proporciona herramientas para la administración, seguimiento, análisis y reporte de incidentes, accidentes, campañas y usuarios relacionados con la gestión de riesgos.

### 7. Actores (Usuarios del Sistema)

Hay tres actores principales definidos:

- **Prevencionista:** Es el actor principal con la mayor cantidad de interacciones. Es el responsable de las funciones operacionales clave de la gestión de riesgos.
- **Administrador:** Es el actor responsable de las funciones de gestión de acceso al sistema.
- **Encargado de Reporte Sustituto:** Un actor con una función única relacionada con el ingreso de reportes de incidentes, funciona como apoyo para el usuario Prevencionista.

### 8. Descripción de los Casos de Uso

Tabla 5: Descripción casos de uso

Caso de Uso	Actor(es) Involucrado(s)	Descripción
Gestionar reportes de incidentes	Prevencionista	Crear, editar, consultar y eliminar (CRUD) informes de incidentes
Ingresar accidente	Prevencionista, Encargado de reportes sustituto	Registrar información detallada sobre un accidente ocurrido
Gestionar campañas	Prevencionista	Crear, editar, consultar y eliminar campañas de prevención o seguridad
Visualizar gráficas	Prevencionista	Ver datos analíticos (probablemente sobre incidentes, accidentes o campañas) en formato gráfico
Descargar gráficas	Prevencionista	Exportar o guardar los datos analíticos para su uso externo o en presentaciones
Gestionar usuarios	Administrador	Crear, modificar, asignar roles y eliminar cuentas de usuario del sistema

```

    erDiagram
        Company ||--|| Area : tiene
        Company ||--|| Area : crea
        Area ||--|| Employee : trabaja
        Area ||--|| Report : involucra
        Documento ||--|| Report : adjunta
        Encargado de Rep. Sus. ||--|| Report : realiza
        Prevencionista ||--|| Report : realiza
        Report ||--|| Event : se registra
        Report }|--|| Employee : está en

        Company {
            string company_id PK
            string companyName
        }
        Area {
            string area_id PK
            string areaName
        }
        Employee {
            string employee_id PK
            string eName
        }
        Report {
            string report_id PK
            string eventDate
            string report_id FK
            string lostDays
            string DIAT
            string DIET
            string dischargeDate
        }
        Documento {
            string doc_id PK
            string rute
            string type
            string fecha_subida
        }
        Usuario {
            string nombre
            string correo
            string rut
        }
        Encargado de Rep. Sus. {
            string presentationDate
        }
        Prevencionista {
        }
        Event {
            string event_id PK
            string type
            string description
        }
  
```

16



## 8.2. Modelo de Tablas Relacionales de la base de datos

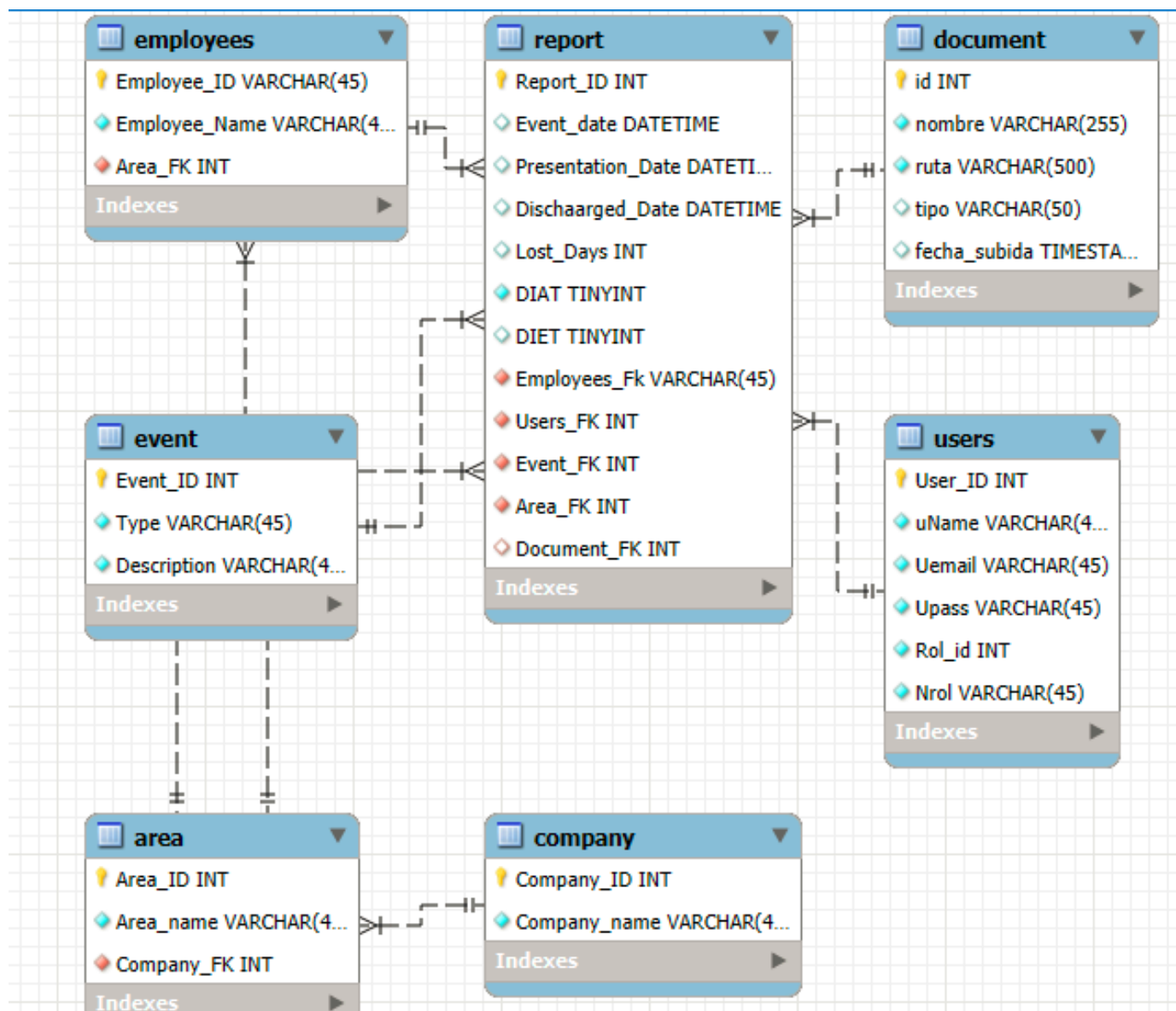


Figura 6: Modelo de Tablas Relacionales

### 8.3. Modelo de proceso de negocio

Procesos de negocio referente a las funcionalidades principales de sistema

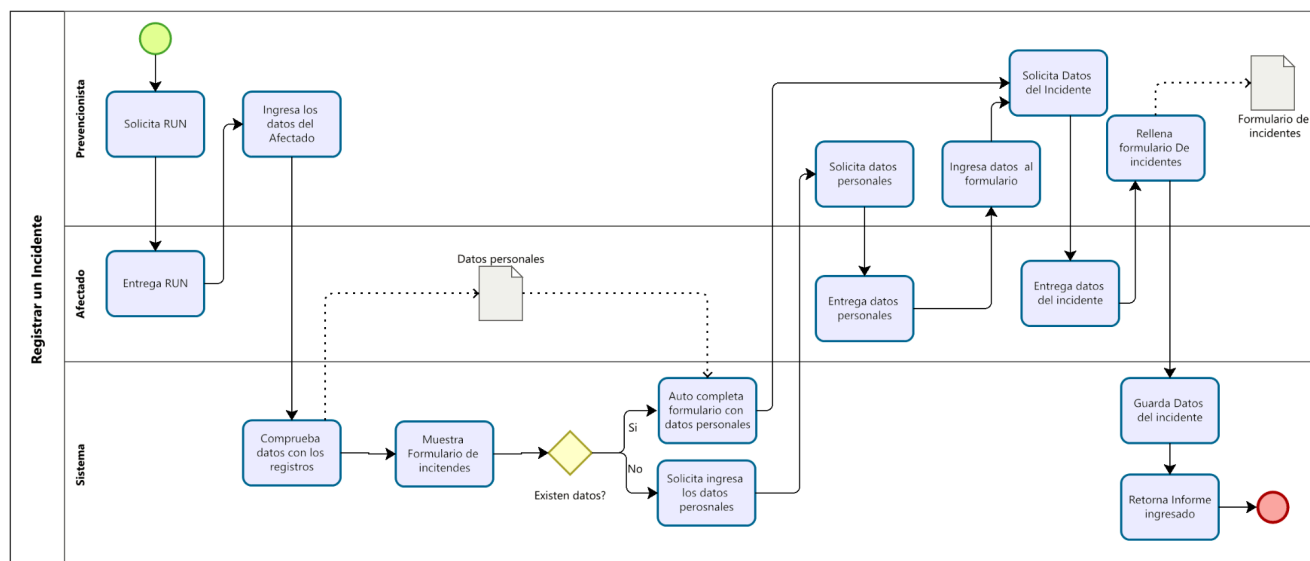


Figura 7: modelo de negocio registrar incidente

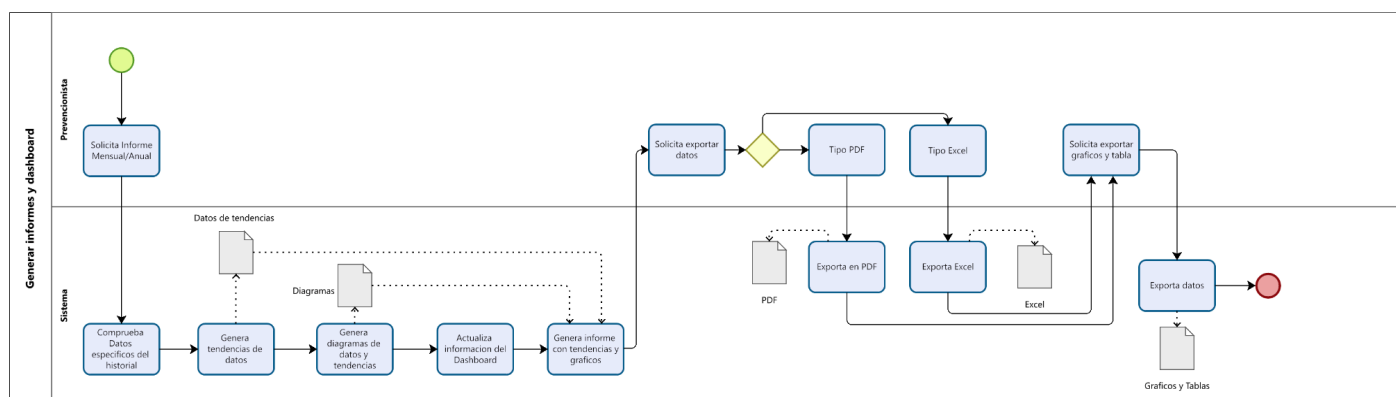


Figura 8: modelo de negocio generar Dashboard e Informe

## 9. Prototipos de Visuales del Proyecto



Figura 8: (Modo nocturno) vista Registro de incidentes

## Plataforma de Gestión de Incidentes

Sistema de registro y análisis de incidentes laborales

[Dashboard](#) [Incidentes](#) [Análisis](#)



### Registrar Incidente

Complete el formulario para registrar un nuevo incidente laboral

Nombre del Empleado

Juan Pérez

RUT

12.345.678-9

Empresa

Seleccione empresa

Área

Seleccione área

Fecha del Incidente

dd-mm-aaaa



Fecha Inicio Licencia

dd-mm-aaaa



Fecha Fin Licencia

dd-mm-aaaa



Registrar Incidente

Figura 11: Modo Diurno vista Registro de incidente

## Plataforma de Gestión de Incidentes

Sistema de registro y análisis de incidentes laborales

Dashboard

Incidentes

Análisis



### Filtros de Análisis

Seleccione el período para visualizar las estadísticas

Año

2025

Mes

Todos los meses

Mostrando 5 incidentes de 2025

### Incidentes por Empresa

Distribución de incidentes entre empresas



### Incidentes por Área

Distribución de incidentes por área de trabajo

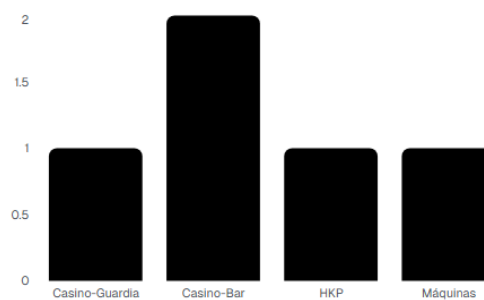


Figura 12: Vista Dashboard

## 10. Herramientas para el desarrollo

### 10.1. Visual Studio Code

Visual Studio Code (conocido también como “VS Code”) es un editor de código fuente gratuito, potente y ligero de código abierto, desarrollado por Microsoft. Es muy popular y funciona en Windows, macOS y Linux. Ofrece funciones como resaltado de sintaxis, autocompletado, depuración y control de versiones (Git). Su amplia gama de extensiones permite personalizarlo y añadir soporte para casi cualquier lenguaje de programación y tecnología, lo que será útil para la mejora constante en el desarrollo del producto. Este será el editor de código utilizado para programar la plataforma, útil debido a sus múltiples extensiones de uso libre y funciones de acceso rápido con los cuales se puede hacer más rápido la implementación de los módulos necesarios.

### 10.2. Angular Material UI

Es una biblioteca de componentes de interfaz de usuario (UI) que se integra con el framework de desarrollo web Angular. Permite a los desarrolladores crear aplicaciones visualmente atractivas y consistentes rápidamente al proporcionar una colección de componentes reutilizables (como botones, tarjetas, campos de entrada) que siguen las pautas de Material Design de Google. Seleccionado para utilizar su colección de componentes en el diseño de la plataforma.

### 10.3. GitHub

Es una plataforma en la nube que sirve como servicio de alojamiento para proyectos de software, utilizando el sistema de control de versiones Git para facilitar la colaboración entre desarrolladores. Permite a los usuarios almacenar, compartir y trabajar en código de forma remota, siguiendo y administrando los cambios a lo largo del tiempo. Se llevará el front-end y back-end del sistema a esta plataforma para controlar las versiones de la fase de desarrollo, mejoras, correcciones e implementaciones de nuevos módulos.

### 10.4. Figma

Plataforma de diseño y prototipado basada en la nube que permite a los equipos crear, compartir y probar diseños de interfaces digitales, como sitios web y aplicaciones. Su principal característica es la colaboración en tiempo real, ya que varias personas pueden trabajar en el mismo proyecto simultáneamente, lo que agiliza el proceso de diseño. Utilizada principalmente para realizar los diseños bosquejo de la interfaz del sistema, con el objetivo de que se pueda seguir una estructura de diseño y ser expuesto al cliente, para corregir errores de diseño, propuestas y funcionalidades pertinentes.

## 10.5. Vercel

Plataforma en la nube que permite a los desarrolladores crear, desplegar y escalar sitios web y aplicaciones de alto rendimiento. Se enfoca en simplificar el proceso de desarrollo frontend y backend, ofrece alojamiento con escalado automático y es conocida por ser la creadora del framework Next.js. Sus características incluyen implementaciones instantáneas, SSL automático y funciones sin servidor. Elegida para desplegar el front-end de la plataforma.

## 10.6. Tailwind CSS

Tailwind es un framework de CSS de código abierto que utiliza clases de utilidad (utility-first) para ayudar a los desarrolladores a diseñar interfaces web de manera eficiente. Tailwind ofrece clases de bajo nivel para cada propiedad CSS (como margen, color de texto, flexbox, etc.) que se aplican directamente en el HTML. Este es útil para evitar el uso de archivos de estilo para la plataforma, además de simplificar el diseño del HTML con la utilización de clases.

# 11. Framework

## 11.1. Angular

Framework de desarrollo web de código abierto, escrito en TypeScript y mantenido por Google, que se utiliza para crear aplicaciones web de una sola página (SPA). Facilita la construcción de aplicaciones rápidas, fiables y escalables mediante un conjunto de herramientas, APIs y bibliotecas. Su arquitectura está basada en componentes, lo que permite crear aplicaciones modulares y reutilizables, el cual será la base para el sistema del lado del cliente.

## 11.2. NestJs

NestJS es un framework de Node.js para crear aplicaciones backend eficientes y escalables, que utiliza TypeScript por defecto para facilitar el desarrollo de aplicaciones profesionales. Está construido sobre Express (o Fastify) y se inspira en la arquitectura de Angular, organizando el código en módulos, controladores y servicios para una estructura clara y mantenible. Su diseño facilita la integración de módulos populares, la escritura de pruebas y el uso de herramientas para automatizar tareas repetitivas, utilizado para la implementación del lado de servidor en el sistema.

## 12. Herramientas de almacenamiento

### 12.1. Google Drive

Es un servicio de almacenamiento en la nube que permite guardar archivos de forma segura, organizarlos en carpetas y acceder a ellos desde cualquier dispositivo. También facilita compartir archivos con otros usuarios y colaborar en documentos, hojas de cálculo y presentaciones en tiempo real. Esta herramienta tiene el objetivo de almacenar toda la documentación digital relacionada al proyecto, ya sea, informes de planificación y de avances, diagramas, recursos para la documentación, enlaces, fuentes, etc.

### 12.2. Free SQL Database

Proporciona una opción gratuita para bases de datos MySQL en la nube, adecuada para proyectos de demostración y pruebas. Permite conectarse utilizando herramientas como MySQL Workbench o PHPMysqlAdmin. Free SQL Database tiene como objetivo ser el espacio de almacenamiento de las tablas, entidades y registros de la base de datos del desarrollo del proyecto.



## 13. Modelado de Datos

### 13.1. Base de datos

```
CREATE TABLE `employees` (
  `Employee_ID` varchar(45) NOT NULL,
  `Employee_Name` varchar(45) NOT NULL,
  `Area_FK` int NOT NULL,
  PRIMARY KEY (`Employee_ID`),
  KEY `fk_Employees_Area1_idx` (`Area_FK`),
  CONSTRAINT `fk_Employees_Area1` FOREIGN KEY
(`Area_FK`) REFERENCES `area` (`Area_ID`)
)
```

```
CREATE TABLE `report` (
  `Report_ID` int NOT NULL,
  `Event_date` datetime DEFAULT NULL,
  `Presentation_Date` datetime DEFAULT NULL,
  `Discharged_Date` datetime DEFAULT NULL,
  `Lost_Days` int DEFAULT NULL,
  `DIAT` tinyint NOT NULL,
  `DIET` tinyint DEFAULT NULL,
  `Employees_Fk` varchar(45) NOT NULL,
  `Users_FK` int NOT NULL,
  `Event_FK` int NOT NULL,
  `Area_FK` int NOT NULL,
  `Document_FK` int DEFAULT NULL,
  PRIMARY KEY (`Report_ID`),
  UNIQUE KEY `Document_FK` (`Document_FK`),
  KEY `fk_Report_Employees1_idx` (`Employees_Fk`),
  KEY `fk_Report_Users1_idx` (`Users_FK`),
  KEY `fk_Report_Event1_idx` (`Event_FK`),
  KEY `fk_Report_Area1_idx` (`Area_FK`),
  CONSTRAINT `fk_Report_Area1` FOREIGN KEY
(`Area_FK`) REFERENCES `area` (`Area_ID`),
  CONSTRAINT `fk_report_document` FOREIGN KEY
(`Document_FK`) REFERENCES `document` (`id`) ON
DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT `fk_Report_Employees1` FOREIGN KEY
(`Employees_Fk`) REFERENCES `employees`
(`Employee_ID`),
  CONSTRAINT `fk_Report_Event1` FOREIGN KEY
(`Event_FK`) REFERENCES `event` (`Event_ID`),
  CONSTRAINT `fk_Report_Users1` FOREIGN KEY
(`Users_FK`) REFERENCES `users` (`User_ID`)
)
```

```
CREATE TABLE `users` (
  `User_ID` int NOT NULL,
  `uName` varchar(45) NOT NULL,
  `Uemail` varchar(45) NOT NULL,
  `Upass` varchar(45) NOT NULL,
  `Rol_id` int NOT NULL,
  `Nrol` varchar(45) NOT NULL,
  PRIMARY KEY (`User_ID`)
)
```

```
CREATE TABLE `document` (
  `id` int NOT NULL AUTO_INCREMENT,
  `nombre` varchar(255) NOT NULL,
  `ruta` varchar(500) NOT NULL,
  `tipo` varchar(50) DEFAULT NULL,
  `fecha_subida` timestamp NULL DEFAULT
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
)
```

```
CREATE TABLE `area` (
  `Area_ID` int NOT NULL,
  `Area_name` varchar(45) NOT NULL,
  `Company_FK` int NOT NULL,
  PRIMARY KEY (`Area_ID`),
  KEY `fk_Area_Company_idx` (`Company_FK`),
  CONSTRAINT `fk_Area_Company` FOREIGN KEY
(`Company_FK`) REFERENCES `company`
(`Company_ID`)
)
```

```
CREATE TABLE `company` (
  `Company_ID` int NOT NULL,
  `Company_name` varchar(45) NOT NULL,
  PRIMARY KEY (`Company_ID`)
)
```

```
CREATE TABLE `event` (
  `Event_ID` int NOT NULL,
  `Type` varchar(45) NOT NULL,
  `Description` varchar(45) NOT NULL,
  PRIMARY KEY (`Event_ID`)
)
```

## 14. Descripción de Modelos de análisis

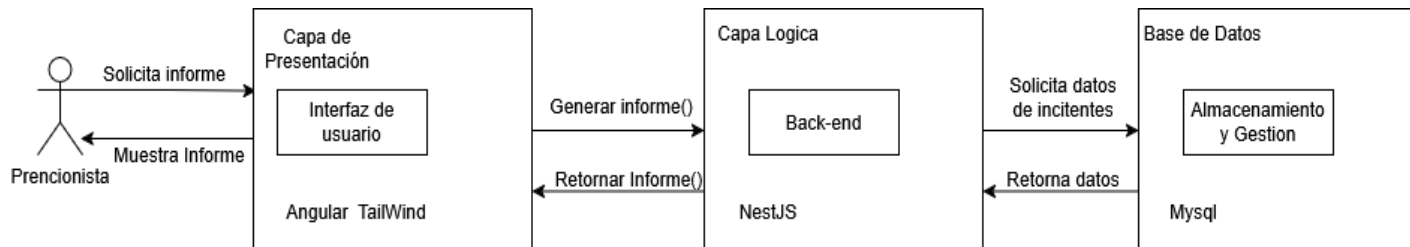


Figura 13: Diagrama de modelo de análisis

En este diagrama donde el prevencionista actúa como el usuario final realiza una solicitud al sistema para solicitar el informe (mensual/anual) donde esta pasara por la Capa de presentación (FrontEnd, en Angular con Tailwind) que recibe la solicitud realizando la petición a la capa lógica (BackEnd, en NestJS) procesando esta solicitud y obteniendo la información desde la BD(Base de datos) retornando la información por el BackEnd donde se ordenar y dará formato para que el FrontEnd lo estructure a una plantilla de informe

## 15. Conclusión

El desarrollo del **Sistema de Analítica en Prevención de Riesgos** permitió cumplir con el objetivo de transformar un proceso manual y fragmentado en una solución integrada, orientada al análisis y uso efectivo de la información. A través del registro estructurado de incidentes, accidentes y actividades preventivas, el sistema facilita el control, la trazabilidad y la consulta histórica de los datos, aportando mayor claridad al trabajo del área de prevención.

En términos generales, el proyecto deja una plataforma funcional y preparada para su despliegue, la cual puede ser ajustada y validada en conjunto con el cliente para su uso operativo. Esta solución no solo aporta valor al Casino Luckia Arica S.A., sino que también representa una experiencia significativa de aplicación práctica de conocimientos, consolidando el trabajo realizado y proyectándose hacia su implementación definitiva.

## 16. Referencias

1. *El modelo en cascada en el desarrollo de software*. (11 octubre). IONOS Digital Guide.  
<https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
2. *MySQL documentation*. (25 octubre). [https://docs.oracle.com/cd/E17952\\_01/index.html](https://docs.oracle.com/cd/E17952_01/index.html)
3. *Angular documentación*. (07 noviembre). <https://angular.dev/>
4. *Angular Material Design components* (07 noviembre).  
<https://material.angular.dev/components/categories>
5. *Official tailwind UI components and templates*. (07 noviembre). <https://tailwindcss.com/plus>
6. *Github, project repository Luckia - APR*. (07 noviembre).  
<https://github.com/DreuxTer/Luckia-PAPR>