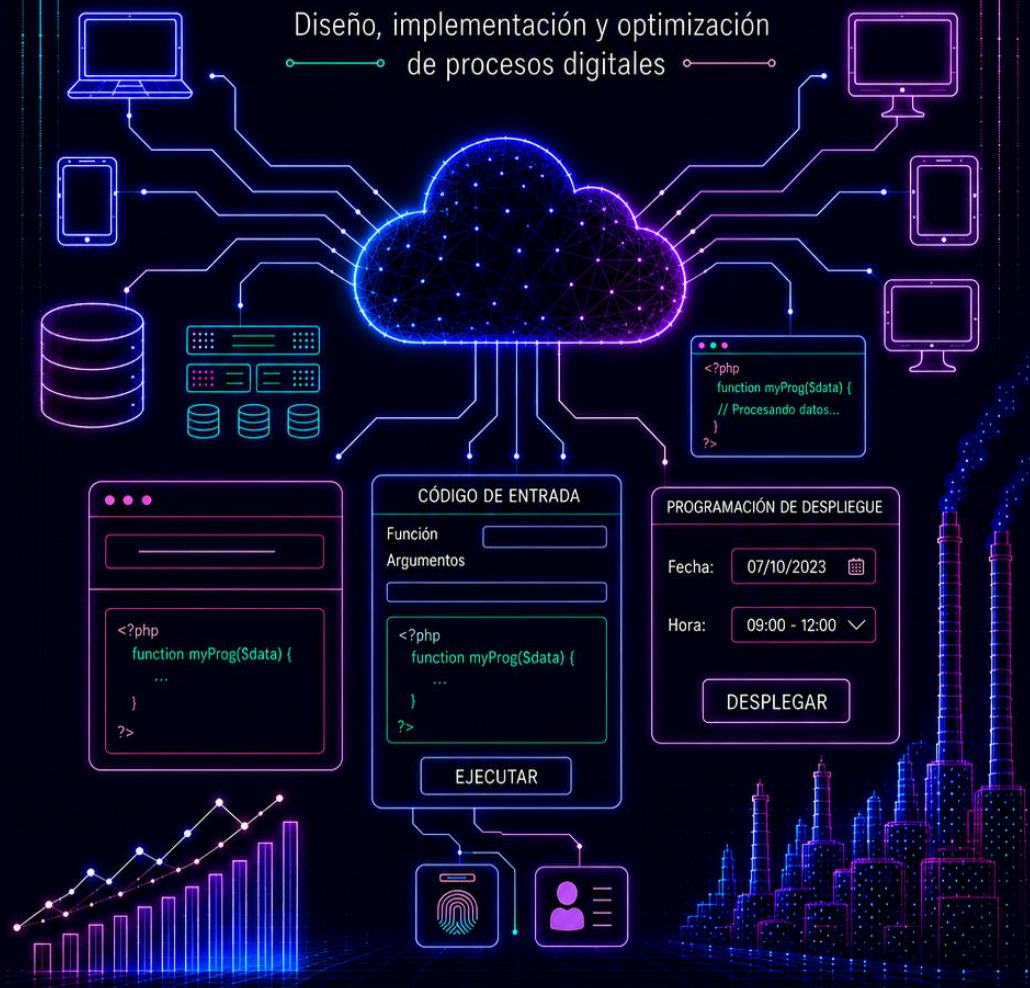


Desarrollo de Aplicaciones Web para Ingeniería Industrial

Diseño, implementación y optimización
de procesos digitales



JULIO CÉSAR ÁLVAREZ REYES

Desarrollo de Aplicaciones web para Ingeniería industrial

Diseño, Implementación y optimización de proceso digitales

© **Julio César Alvarez Reyes**

<https://orcid.org/0000-0001-8465-8907>

Editada por:

© Professionals On Line SAC. (FEPOL) – Fondo Editorial.

Dirección Av. General José de San Martín N° 790, Dpto. 402,

Urb. El Carmen - Pueblo Libre, Lima, Perú.

professionalsonline.net@gmail.com

Teléf. móvil: +51 981 105 611

Web: <https://professionals.pe/>

Primera edición digital: Junio de 2026

Libro digital disponible en <https://editorialfondo.com/>

Depósito Legal en la Biblioteca Nacional del Perú N° 2026-06277

ISBN: 978-612-99409-2-2

DOI: <https://doi.org/10.47422/>

Diseño y Diagramación: Gráfica “Imagen”

Professionals On Line

professionalsonline.net@gmail.com

Sello editorial: Fondo Editorial (978-612-48981)



Licencia No Comercial Creative Commons

<https://creativecommons.org/licenses/by-nc/4.0/deed.es>

Professionals On Line (FEPOL)

ISSN: 2955-8549

Este libro está dedicado a:

A mis estudiantes de Ingeniería Industrial, presentes y futuros, porque en sus preguntas y desafíos encontré la razón para escribir este libro. Que estas páginas les devuelvan, al menos en parte, la curiosidad y el entusiasmo que ellos me regalan cada semestre.

A la memoria de mi padre, escritor prolijo donde los hubo, que me enseñó que cada palabra debe ser cuidadosamente elegida y que un texto bien construido puede transformar realidades. Ojalá estuvieras aquí para corregir estos párrafos con tu lápiz rojo.

A mi madre, por sus enseñanzas silenciosas pero profundas: la constancia, la honestidad y la convicción de que el conocimiento es el único tesoro que nadie puede arrebatar.

A mi esposa e hijos, motor y motivo de cada amanecer. Ellos son la razón por la que busco ser mejor cada día y el puerto al que siempre regreso después de navegar entre líneas de código y capítulos interminables.

Prefacio

La ingeniería industrial se encuentra en una encrucijada histórica. Durante décadas, nuestra disciplina se ha enfocado en la optimización de tiempos, movimientos y recursos físicos dentro de los procesos productivos. Sin embargo, en el contexto de la Industria 4.0, la eficiencia ya no se mide únicamente en la planta de producción, sino también en la velocidad, calidad y precisión con la que fluye la información a través de la organización. Hoy en día, un ingeniero industrial que no comprende la arquitectura de los datos generados por sus procesos está, en cierta medida, operando a ciegas.

Este libro surge de una necesidad observada tanto en las aulas universitarias como en el sector productivo: cerrar la brecha entre la gestión de los procesos y el desarrollo de soluciones tecnológicas. En muchos casos, las soluciones de software comerciales resultan rígidas o excesivamente costosas, mientras que las hojas de cálculo tradicionales terminan convirtiéndose en islas de información difíciles de mantener y propensas al error. Entre ambos extremos existe un espacio donde el ingeniero, con las herramientas adecuadas, puede diseñar y construir sus propias soluciones digitales, ágiles, escalables y alineadas con la lógica de los procesos que gestiona.

Esta obra no es un manual de informática dirigido a programadores. Es, más bien, un mapa de ruta para ingenieros industriales que desean comprender cómo los procesos pueden transformarse en sistemas de información. A lo largo de estas páginas, el lector no solo aprenderá

la sintaxis de lenguajes como PHP o la construcción de consultas SQL, sino que aprenderá a modelar la realidad operativa dentro de una estructura digital, conectando procesos, datos y decisiones.

El caso práctico central de esta obra el sistema de gestión SSOMA de la Minera ZET no fue elegido al azar. Representa un entorno industrial donde la gestión de la información es crítica. En estos contextos, una adecuada administración de los datos no solo impacta en la eficiencia operativa, sino también en la seguridad de las personas y en la sostenibilidad de las operaciones.

Mi objetivo con este texto es ofrecer al estudiante y al profesional una perspectiva diferente sobre el papel de la tecnología en la ingeniería industrial. Más que depender exclusivamente de herramientas externas, el ingeniero puede desarrollar la capacidad de transformar procesos manuales en sistemas de información estructurados y robustos. La tecnología no debe ser una caja negra, sino un espacio donde la ingeniería industrial proyecte su capacidad analítica, su creatividad y su visión de mejora continua.

Te invito a recorrer este camino con curiosidad, disciplina y espíritu crítico. En el mundo de la transformación digital, el código se convierte en una de las herramientas más poderosas que un ingeniero puede utilizar para comprender, mejorar y transformar los procesos de la industria.

Tabla de contenido

PARTE I. Fundamentos para Ingenieros Industriales	7
CAPÍTULO 1. Digitalización de procesos en la ingeniería industrial.....	7
CAPÍTULO 2. Fundamentos de las aplicaciones web	14
CAPÍTULO 3. Modelado del proceso antes de programar	16
PARTE II – Bases de datos para procesos industriales	25
CAPÍTULO 4. Diseño de Base de datos para aplicaciones industriales.....	25
PARTE III. Desarrollo Web con PHP para sistemas industriales	44
CAPÍTULO 5. Desarrollo de Interfaces Web para Sistemas Industriales ..	44
CAPÍTULO 6. Fundamentos de Programación en PHP.....	49
CAPÍTULO 7. Conexión entre PHP y MySQL.....	54
CAPÍTULO 8. Desarrollo del sistema SSOMA de la minera ZET	57
CAPÍTULO 9. Consultas para reportes e indicadores	73
CAPÍTULO 10. Proyecto integrado final.....	82

Foro
Editorial

PARTE I. Fundamentos para Ingenieros Industriales

CAPÍTULO 1. Digitalización de procesos en la ingeniería industrial

1.1 La nueva realidad de la ingeniería industrial

Durante décadas, el ingeniero industrial ha sido reconocido como el profesional especializado en la optimización de procesos productivos, la mejora de la eficiencia y la gestión de recursos. Sin embargo, el entorno empresarial actual exige algo más: la capacidad de integrar procesos físicos con sistemas digitales.

Hoy, la competitividad no depende únicamente de producir más o reducir costos, sino de contar con información precisa, oportuna y estructurada que permita tomar decisiones basadas en datos.

En este contexto, la digitalización de procesos se convierte en una competencia estratégica del ingeniero industrial moderno.

1.2 Procesos tradicionales vs procesos digitalizados

1.2.1 Procesos tradicionales

En muchas organizaciones aún predominan:

- Registros en papel
- Uso excesivo de hojas de cálculo no integradas
- Información fragmentada
- Duplicidad de datos
- Retrasos en consolidación de información

Estos sistemas generan:

- Errores humanos
- Falta de trazabilidad
- Dificultad para medir indicadores
- Decisiones tardías



Fig. 1 Procesos manuales vs procesos digitales

1.2.2 Procesos digitalizados

La digitalización implica convertir la información operativa en datos estructurados almacenados en sistemas informáticos.

Un proceso digitalizado permite:

- Registro en tiempo real
- Base de datos centralizada
- Generación automática de indicadores

- Reducción de errores
- Acceso remoto y control integrado

No se trata solo de “usar computadoras”, sino de estructurar la información de manera sistemática.

1.3 Rol del ingeniero industrial en el desarrollo de aplicaciones web

El ingeniero industrial no necesariamente debe convertirse en un programador profesional, pero sí debe:

- Comprender la lógica de los sistemas
- Modelar correctamente los procesos
- Identificar los datos críticos
- Diseñar soluciones alineadas a la operación

Cuando un ingeniero industrial entiende cómo funciona una aplicación web, puede:

- Diseñar mejores requerimientos
- Supervisar desarrollos tecnológicos
- Implementar soluciones internas
- Reducir dependencia externa

1.4 ¿Qué es una aplicación web industrial?

Una aplicación web es un sistema informático que:

- Funciona a través de un navegador
- Se conecta a una base de datos

Desarrollo de Aplicaciones web para Ingeniería industrial

- Permite registrar, consultar y procesar información

En el contexto industrial, puede servir para:

- Control de inventarios
- Registro de producción
- Seguimiento de mantenimiento
- Gestión de seguridad y salud
- Control de calidad

Una aplicación web industrial no es solo tecnología: es una herramienta de gestión.

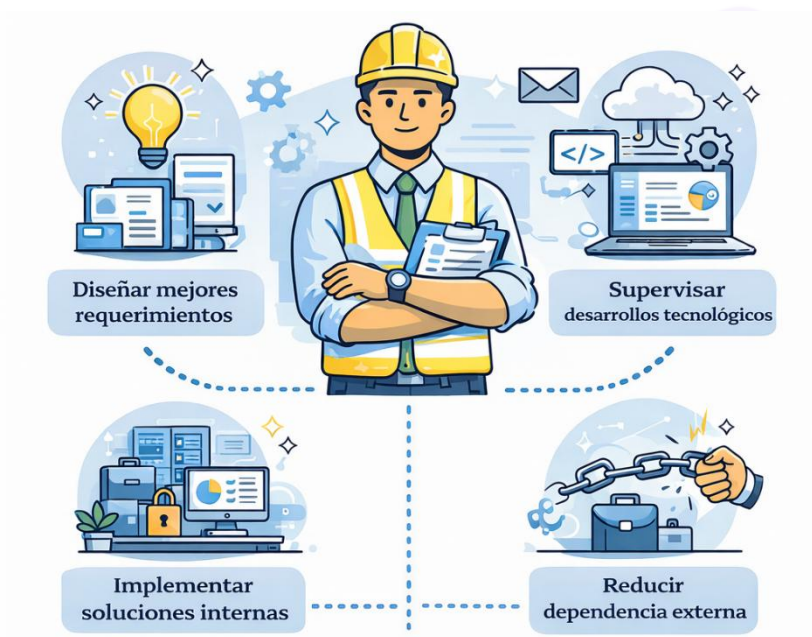


Fig. 2 Ingeniería industrial y transformación digital

1.5 Proyecto integrado

A lo largo del desarrollo de este libro se construirá progresivamente un caso de estudio integral que permitirá aplicar los conceptos de modelado, diseño de base de datos y desarrollo web en un contexto industrial real.

El caso se centra en la Minera ZET, una empresa de mediana escala dedicada a la extracción y procesamiento de minerales metálicos. Debido a la naturaleza de sus operaciones, la organización trabaja bajo estrictos estándares de seguridad y salud ocupacional, ya que enfrenta condiciones de alto riesgo tales como:

- Exposición constante a riesgos físicos y mecánicos
- Trabajo en altura
- Operaciones con maquinaria pesada
- Manipulación de explosivos

La empresa cuenta con:

- 600 trabajadores (entre personal directo y contratistas)
- Cuatro áreas críticas: Perforación, Voladura, Transporte y Planta
- Una Gerencia de Operaciones
- Una Jefatura de Seguridad, Salud Ocupacional y Medio Ambiente (SSOMA)

En el marco de su estrategia de mejora continua y transformación digital, la Minera ZET ha decidido modernizar su sistema de gestión

de accidentes laborales mediante el desarrollo de una aplicación web interna que permita registrar, analizar y monitorear la información de seguridad en tiempo real.

Este proyecto integrado será el eje práctico del libro y acompañará al estudiante desde el modelado del proceso hasta la implementación completa del sistema.

1.6 Ejercicios prácticos. Selección del caso de estudio

Como parte del enfoque aplicado del libro, los estudiantes deberán conformar equipos de trabajo y seleccionar un caso de estudio que será desarrollado progresivamente a lo largo de todo el libro.

El caso elegido será el eje integrador del proyecto académico y permitirá aplicar los contenidos de modelado de procesos, diseño de base de datos, desarrollo con PHP y MySQL, e implementación de indicadores.

En esta primera etapa, el equipo deberá presentar lo siguiente:

- Descripción general de la empresa u organización (actividad económica, tamaño, estructura básica y área seleccionada).
- Identificación de los principales problemas operativos o de gestión relacionados con el proceso que se desea digitalizar.
- Justificación técnica del proyecto, explicando por qué la implementación de una aplicación web contribuiría a mejorar la eficiencia, el control y la toma de decisiones.

El objetivo de esta actividad es que el estudiante comprenda que todo sistema de información debe responder a una necesidad real y estar alineado con un problema concreto de gestión.

Sugerencias de casos para el proyecto

Los estudiantes podrán seleccionar, entre otros, los siguientes tipos de procesos:

- Sistema de control de inventarios en una empresa comercial o industrial.
- Sistema de registro y análisis de accidentes laborales (SSOMA).
- Sistema de control de producción diaria.
- Sistema de gestión de mantenimiento de equipos.
- Sistema de control de asistencia del personal.
- Sistema de seguimiento de pedidos y despacho.
- Sistema de gestión de capacitaciones internas.
- Sistema de control de calidad y no conformidades.

Se recomienda que el caso seleccionado:

- Tenga un flujo de proceso claramente identificable.
- Involucre múltiples registros y usuarios.
- Permita generar indicadores de gestión.

CAPÍTULO 2. Fundamentos de las aplicaciones web

2.1 Arquitectura básica cliente – servidor

En este libro se utiliza PHP y MySQL porque permiten desarrollar aplicaciones web funcionales de manera clara, estructurada y accesible para estudiantes de pregrado en Ingeniería Industrial.

PHP facilita la comprensión de la lógica del servidor y el procesamiento de datos, mientras que MySQL permite diseñar bases de datos relacionales que estructuran la información de forma organizada y confiable. Esta combinación ayuda al estudiante a entender cómo un proceso industrial se transforma en datos, cómo esos datos se almacenan y cómo se convierten en información útil para la toma de decisiones.

Además, PHP y MySQL son tecnologías ampliamente utilizadas, fáciles de implementar en entornos académicos y suficientes para construir sistemas reales como control de inventarios, producción o gestión de seguridad.

La elección tecnológica no busca formar programadores especializados, sino ingenieros industriales capaces de integrar procesos, datos y soluciones digitales de manera práctica y aplicada.

Toda aplicación web se estructura bajo el modelo arquitectónico cliente-servidor, el cual permite distribuir responsabilidades y optimizar el procesamiento de la información.

Desarrollo de Aplicaciones web para Ingeniería industrial

- **Cliente:** Es la interfaz de interacción con el usuario, generalmente un navegador web. Se encarga de enviar solicitudes al servidor y mostrar las respuestas generadas por el sistema.
- **Servidor:** Es el entorno donde se ejecuta la lógica del negocio. En este caso, el código desarrollado en PHP procesa las solicitudes recibidas, valida la información, aplica reglas operativas y gestiona la comunicación con la base de datos.
- **Base de datos:** Constituye el sistema de almacenamiento estructurado, como MySQL, donde se registran, organizan y resguardan los datos de manera persistente, garantizando integridad, seguridad y disponibilidad de la información.

Flujo básico:

Usuario → Navegador → Servidor (PHP) → Base de datos →
Respuesta al usuario



Fig. 3 Arquitectura cliente - servidor

CAPÍTULO 3. Modelado del proceso antes de programar

3.1 Importancia del modelado

Uno de los errores más comunes en desarrollo de sistemas es programar sin entender el proceso.

El ingeniero industrial debe:

- Analizar el flujo operativo
- Identificar entradas y salidas
- Definir responsabilidades
- Determinar información crítica

Sin modelado, no hay sistema eficiente.

3.2 Levantamiento de requerimientos

Se deben responder preguntas como:

- ¿Qué datos se registran?
- ¿Quién los registra?
- ¿Con qué frecuencia?
- ¿Qué indicadores se necesitan?
- ¿Qué decisiones dependen de esa información?

3.3 Identificación de entidades

A partir del proceso se identifican:

- Productos
- Órdenes

- Operarios
- Incidentes
- Movimientos

Estas entidades luego se convertirán en tablas en la base de datos.

3.4 Definición de indicadores

El sistema debe diseñarse pensando en generar estos indicadores automáticamente.

- Eficiencia de producción
- Nivel de inventario
- Tasa de incidentes
- Tiempo promedio de reparación

3.5 Desarrollo caso de estudio Minera ZET

En la Minera ZET, el mapeo del proceso de gestión de accidentes, permite comprender cómo se detecta un evento, cómo se registra en el sistema, quién lo investiga, qué medidas se implementan y cómo se realiza el seguimiento hasta su cierre formal.

3.5.1 Mapa de proceso

Mapear procesos no solo describe lo que ocurre, sino que permite:

- Detectar ineficiencias y cuellos de botella
- Identificar riesgos operativos
- Definir responsabilidades claras
- Estandarizar procedimientos

- Facilitar la automatización mediante una aplicación web

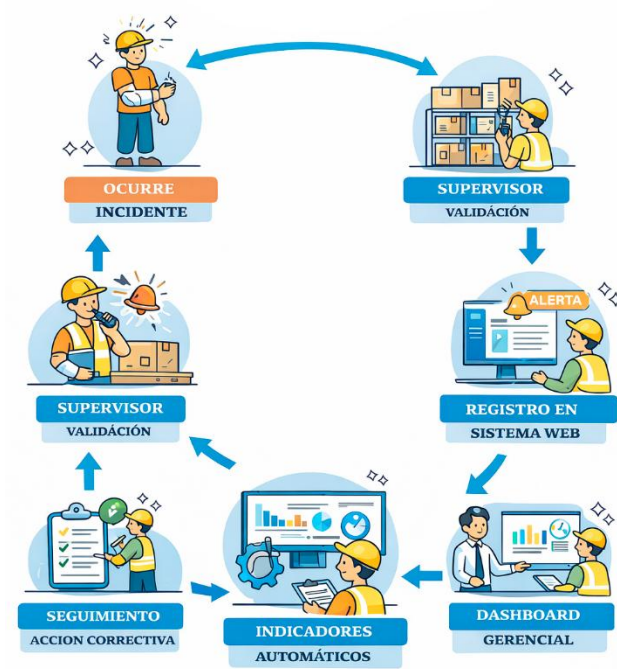


Fig. 4 Mapa de procesos seguimiento de accidentes Minera ZET

Subprocesos sistema SSOMA Minera ZET

Detección del evento

- Ocurre incidente o accidente
- Supervisor identifica situación

Registro del evento

- Ingreso en la aplicación web
- Clasificación (Incidente / Accidente)

- Registro preliminar

Notificación automática

- Alerta a Jefatura SSOMA
- Notificación a Gerencia de Operaciones

Investigación

- Asignación de responsable
- Identificación de causas
- Determinación de gravedad

Definición de medidas

- Correctivas
- Preventivas
- Capacitación relacionada

Seguimiento y cierre

- Validación de ejecución
- Verificación de eficacia
- Cierre formal del evento

3.5.2 Lista de entidades

El análisis de entidades constituye una etapa previa esencial al modelado relacional, ya que traduce el proceso operativo en componentes estructurados de información. Esta abstracción

sistemática permite que el diseño de la base de datos refleje fielmente la realidad organizacional.

- **Área.** Representa las unidades operativas o departamentos dentro de la organización (por ejemplo, perforación, voladura, transporte o planta). Permite clasificar la información y analizar la accidentabilidad por sector operativo.
- **Trabajador.** Contiene la información personal y laboral del empleado, incluyendo identificación, puesto y área asignada. Es la entidad central para vincular accidentes, capacitaciones, EPP y acciones correctivas.
- **Accidente.** Registra el evento no deseado ocurrido en la operación. Incluye fecha, hora, descripción y su clasificación causal. Constituye el núcleo del sistema de gestión de incidentes.
- **Tipo de accidente.** Catálogo que clasifica los accidentes según su naturaleza (caída, golpe, atrapamiento, etc.), permitiendo análisis estadísticos por tipo de evento.
- **Categoría.** Entidad que agrupa causas raíz bajo categorías generales, facilitando la organización y análisis estructurado de los factores que originan los accidentes.
- **Causa raíz.** Registra la causa fundamental que originó el accidente, identificada durante la investigación. Es clave para el enfoque preventivo y la implementación de acciones correctivas efectivas.

- Turno. Define la franja horaria de trabajo (mañana, tarde, noche). Permite analizar la incidencia de accidentes según el horario operativo.
- Gravedad. Clasifica el nivel de severidad del accidente (leve, incapacitante, fatal), lo que permite calcular indicadores como el Índice de Gravedad.
- Condición peligrosa. Catálogo que identifica condiciones físicas o ambientales inseguras presentes en el entorno laboral al momento del accidente.
- Acto inseguro. Registra conductas o acciones incorrectas realizadas por el trabajador que contribuyeron al accidente. Es parte del análisis causal del evento.
- Acción correctiva. Representa las medidas implementadas para eliminar o mitigar las causas del accidente. Incluye responsable, fechas y estado, cerrando el ciclo de mejora continua.
- EPP. Catálogo de Equipos de Protección Personal utilizados en la organización (casco, arnés, respirador, etc.). Permite controlar asignación, vigencia y cumplimiento normativo.
- Capacitación. Registra los eventos formativos en seguridad y salud ocupacional. Permite verificar si el trabajador recibió capacitación previa y evaluar su relación con la ocurrencia de accidentes.

3.5.3 Lista de indicadores esperados

Indicadores de frecuencia

- Tasa de Frecuencia de Accidentes (TF)
- Número de incidentes por área
- Accidentes por cada 100 trabajadores
- Eventos por tipo (Incidente vs Accidente)

Indicadores de Gravedad

- Tasa de Gravedad (TG)
- Total de días perdidos
- Índice de Severidad

Indicadores de Gestión

- % eventos investigados dentro de plazo
- Tiempo promedio de cierre
- % medidas ejecutadas en fecha
- % medidas eficaces

Indicadores Preventivos

- Trabajadores capacitados por área
- Accidentes antes y después de capacitación
- Área con mayor recurrencia
- Tipo de causa más frecuente

Indicadores Estratégicos (Gerencia)

- Ranking de áreas críticas
- Tendencia mensual de accidentabilidad
- Costo estimado por accidentes
- Nivel de cumplimiento SSOMA

3.6 Entregable del capítulo

En función del caso de estudio seleccionado en el Capítulo 1(sección 1.6), el estudiante deberá presentar los siguientes elementos como resultado del análisis desarrollado en este capítulo:

- Mapa del proceso, representando gráficamente el flujo actual o propuesto del sistema a digitalizar.
- Lista de entidades identificadas, derivadas del análisis conceptual previo al modelado de datos.
- Lista de indicadores esperados, que el sistema deberá generar para apoyar la toma de decisiones.

Este entregable tiene como propósito validar que el estudiante ha comprendido el proceso antes de aplicar técnicas formales de modelado y desarrollo, asegurando que el diseño del sistema responda a necesidades reales de gestión.

Resumen de la Parte I. Fundamentos para Ingenieros Industriales

Esta primera parte establece el marco conceptual que justifica la participación del ingeniero industrial en proyectos de digitalización. Se explica por qué la transformación digital es hoy una competencia estratégica, contrastando los procesos tradicionales basados en papel con los digitalizados que permiten registro en tiempo real y generación automática de indicadores. Se introducen los fundamentos de las aplicaciones web y la arquitectura cliente-servidor, justificando el uso de PHP y MySQL por su claridad y aplicabilidad en contextos industriales. El núcleo de esta parte es el modelado del proceso antes de programar, donde se enseña a levantar requerimientos, identificar entidades y definir indicadores. Todo se aplica al caso de estudio de la Minera ZET y su sistema de gestión de accidentes, sentando las bases para que los estudiantes analicen su propio caso.

PARTE II – Bases de datos para procesos industriales

CAPÍTULO 4. Diseño de Base de datos para aplicaciones industriales

4.1 La base de datos como núcleo del sistema

En toda aplicación industrial moderna, la base de datos constituye el núcleo estructural del sistema de información. No se trata simplemente de un repositorio donde se almacenan datos, sino del componente que organiza, relaciona y preserva la información crítica que sostiene la operación empresarial.

En un entorno industrial, los procesos generan datos de manera constante: registros de producción, controles de calidad, reportes de incidentes, mantenimientos, inventarios, capacitaciones, indicadores de desempeño y trazabilidad operativa. Sin una estructura adecuada para gestionarlos, estos datos pierden coherencia, confiabilidad y valor estratégico.

Desde la perspectiva de la Ingeniería Industrial, la base de datos cumple cinco funciones esenciales:

4.1.1 Estructuración de la información

Permite organizar los datos bajo un modelo lógico que refleje la realidad del proceso productivo. Cada entidad (trabajador, máquina, evento, orden, área) y cada relación entre ellas debe estar correctamente definida para evitar redundancias y asegurar integridad.

4.1.2 Soporte a la operación

Los sistemas web o aplicaciones internas dependen directamente de la base de datos para registrar transacciones, validar información y ejecutar reglas de negocio. Si la estructura de datos es deficiente, la aplicación será inestable o inconsistente.

4.1.3 Garantía de integridad y trazabilidad

En procesos industriales, la trazabilidad es crítica. Por ejemplo:

- ¿Qué trabajador estuvo involucrado en un accidente?
- ¿Qué máquina presentó fallas recurrentes?
- ¿Qué lote de producción tuvo desviaciones?

La base de datos permite reconstruir la historia operativa con precisión.

4.1.4 Fuente para la analítica e indicadores

Los indicadores de gestión (KPIs) no se generan desde la interfaz visual, sino desde la estructura de datos. Una base correctamente diseñada facilita el cálculo de:

- Tasa de accidentabilidad
- Productividad por turno
- Cumplimiento de mantenimiento
- Índices de calidad

Sin diseño adecuado, la analítica se vuelve limitada o poco confiable.

4.1.5 Sostenibilidad del sistema en el tiempo

Un sistema industrial debe escalar. Nuevas áreas, más trabajadores, mayores volúmenes de información. Una base de datos bien diseñada permite crecimiento ordenado sin necesidad de reconstruir el sistema desde cero.

4.2 La base de datos como reflejo del proceso

Antes de programar cualquier aplicación industrial, es imprescindible mapear el proceso. El modelo de datos no debe inventar la realidad, sino representarla estructuralmente.

La base de datos es, por tanto, la traducción estructurada del proceso organizacional.

4.3 Diseño orientado a procesos industriales

En aplicaciones industriales, el diseño de la base de datos debe considerar:

- Relaciones uno a muchos (1:N)
- Relaciones muchos a muchos (N:M)
- Integridad referencial
- Restricciones obligatorias
- Campos auditables (fecha_registro, usuario_registro)
- Estados del proceso
- Históricos y seguimiento

Además, el diseño debe estar alineado con principios de normalización, evitando duplicidad de información y asegurando consistencia operativa.

4.4 Error común en proyectos industriales

Uno de los errores más frecuentes en proyectos de transformación digital es iniciar la programación de la aplicación web sin haber diseñado previamente la base de datos.

Esto provoca:

- Tablas improvisadas
- Datos duplicados
- Dificultad para generar reportes
- Inconsistencias en reglas de negocio
- Reprogramaciones costosas

En ingeniería, primero se diseña la estructura; luego se construye el sistema.

4.5 Base de datos como activo estratégico

En la actualidad, la información es uno de los principales activos de la organización. Una base de datos industrial bien diseñada:

- Reduce riesgos operativos
- Mejora la toma de decisiones
- Facilita auditorías
- Permite análisis predictivo

- Contribuye a la mejora continua

La base de datos no es un componente técnico aislado; es el eje que conecta operación, control y estrategia.

4.6 Referencia a profundización técnica

Los aspectos técnicos detallados sobre:

- Modelo Entidad–Relación
- Normalización (1FN, 2FN, 3FN)
- Diseño conceptual, lógico y físico
- Claves primarias y foráneas
- Integridad referencial
- Implementación en MySQL
- Buenas prácticas de modelado

Se desarrollan de manera exhaustiva en:

Alvarez, J. (2026). Bases de datos para la Ingeniería Industrial (1ª ed.).
Professionals On line S.A.C.

<https://isbn.bnpgob.pe/catalogo.php?mode=detalle&nt=166544>

Se recomienda al lector revisar dicha obra para comprender en profundidad la construcción estructurada de una base de datos aplicada a contextos industriales.

4.7 Desarrollo modelo de datos caso de estudio Minera ZET

El diseño del modelo de datos constituye una de las etapas más críticas en el desarrollo de cualquier sistema de información, ya que define la estructura lógica sobre la cual se soportará toda la aplicación. En el caso de la Minera ZET, el objetivo no fue únicamente almacenar información relacionada con accidentes laborales, sino diseñar una arquitectura relacional integral que permita articular trabajadores, eventos de seguridad, equipos de protección personal, capacitaciones y acciones correctivas dentro de una plataforma coherente, segura y analíticamente robusta.

El modelo desarrollado es el resultado del análisis previo del proceso operativo, la identificación de requerimientos funcionales y el estudio del marco normativo aplicable al sector minero. Sin embargo, su alcance no se limita al componente operativo del sistema SSOMA, sino que incorpora también un módulo de seguridad y control de acceso basado en roles, garantizando autenticación, autorización y trazabilidad de las operaciones realizadas dentro del sistema.

De esta manera, el modelo no solo estructura los datos operativos de seguridad y salud ocupacional, sino que establece una base institucional sólida para el desarrollo del sistema web SSOMA que será implementado en las siguientes etapas del libro.

El script SQL para la creación de la base de datos del sistema SSOMA se incluye en el *Anexo 1*, permitiendo al lector reproducir la estructura completa del modelo de datos desarrollado en este libro.

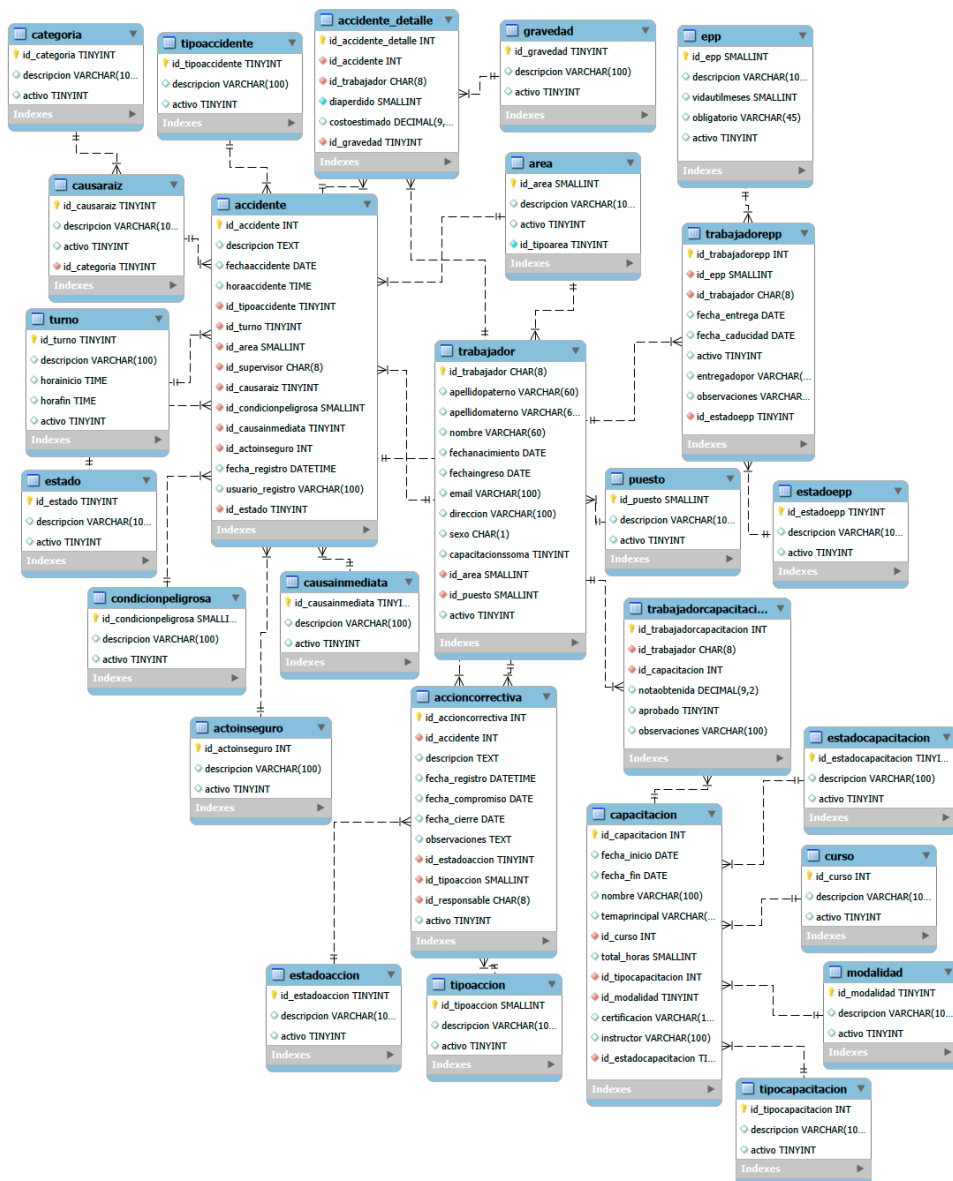


Fig. 5 Modelo de datos SSOMA Minera ZET

4.7.1 Núcleo del sistema: registro y análisis de accidentes

El eje central del modelo es el módulo de gestión de accidentes, el cual concentra la información crítica relacionada con los eventos no deseados ocurridos en la operación minera.

Tabla accidente

Esta entidad constituye el registro estructural del evento y responde a tres preguntas fundamentales de análisis:

- ¿Qué ocurrió? (descripción del evento)
- ¿Cuándo ocurrió? (fecha y hora)
- ¿Dónde ocurrió? (área, turno, supervisor responsable)

Adicionalmente, el accidente se vincula con:

- Causa raíz
- Causa inmediata
- Acto inseguro
- Condición peligrosa
- Tipo de accidente

Esta estructura permite no solo documentar el evento, sino clasificarlo bajo criterios técnicos de investigación de accidentes.

Tabla accidente_detalle

La incorporación de esta entidad fortalece significativamente el modelo, ya que permite gestionar la relación uno-a-muchos entre un accidente y los trabajadores afectados.

Cada registro en accidente_detalle incluye:

- Trabajador involucrado
- Días perdidos
- Costo estimado
- Gravedad específica

Este diseño evita redundancia de información general del evento y permite realizar análisis individualizados por trabajador, mejorando la precisión estadística del sistema.

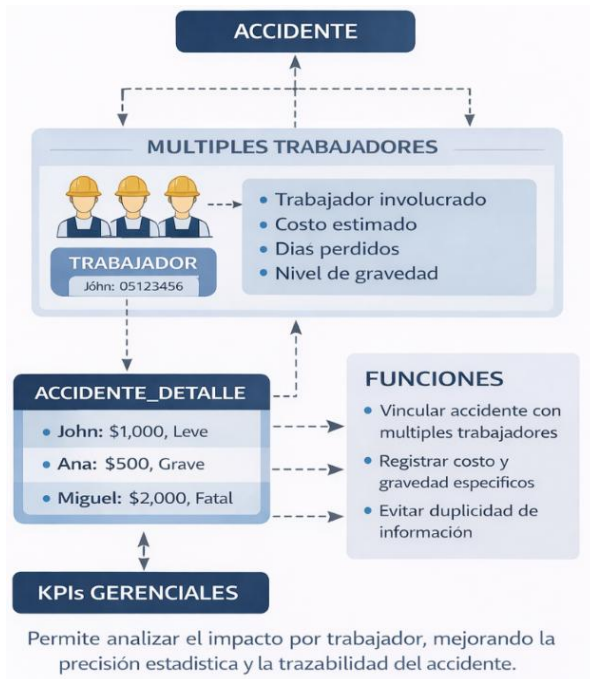


Fig. 6 Análisis de la tabla accidente_detalle

Catálogos de análisis causal

Las tablas que constituyen el marco de clasificación técnica del accidente:

- *causaraiz*
- *causainmediata*
- *actoinseguro*
- *condicionpeligrosa*
- *gravedad*
- *tipoaccidente*

La separación de estos catálogos permite:

- Estandarización de criterios bajo metodologías internacionales de investigación.
- Análisis estadístico estructurado.
- Identificación de patrones recurrentes.
- Reducción de subjetividad en el registro.

Desde el punto de vista del diseño relacional, esta descomposición cumple con principios de normalización y facilita consultas analíticas complejas.

4.7.2 Gestión del talento humano y estructura organizativa

La entidad *trabajador* representa el componente humano del sistema y permite vincular la accidentabilidad con variables laborales y demográficas.

Tabla trabajador

Contiene información como:

- Documento de identidad
- Nombres y apellidos
- Sexo
- Fecha de ingreso
- Área
- Puesto

Esto permite realizar análisis tales como:

- Accidentes por antigüedad.
- Accidentes por puesto crítico.
- Accidentabilidad por área operativa.
- Exposición diferencial por perfil laboral.

Estructura Organizativa

Las siguientes tablas permiten contextualizar el accidente dentro de la organización:

- Puesto
- Turno
- Área

Esta segmentación es fundamental para:

- Evaluar desempeño por departamento.

- Analizar accidentes por turno (ej. turno noche).
- Identificar áreas de mayor riesgo operativo.

Desde la perspectiva de gestión industrial, esta estructura facilita la toma de decisiones focalizadas.

4.7.3 Módulo de capacitación y cumplimiento normativo

Este bloque cumple una función preventiva y de cumplimiento legal.

Tabla *capacitacion*

Registra los eventos formativos incluyendo:

- Curso asociado
- Instructor
- Fechas de ejecución
- Modalidad
- Tipo de capacitación

Permite evidenciar cumplimiento de programas formativos obligatorios.

Tabla *trabajadorcapacitacion*

Registra el historial individual del trabajador:

- Capacitación recibida
- Nota obtenida
- Condición de aprobación

Este diseño permite responder preguntas clave como:

- ¿El trabajador accidentado contaba con capacitación vigente?
- ¿Había aprobado la formación específica para su puesto?
- ¿Existe correlación entre accidentes y falta de capacitación?

Desde una perspectiva de auditoría, esta información es esencial para demostrar diligencia preventiva.

4.7.4 Control de EPP y gestión de acciones correctivas

Gestión de EPP

La tabla *trabajadorepp* permite registrar:

- Entrega de equipo.
- Fecha de caducidad.
- Estado del EPP.
- Responsable de entrega.

Este módulo permite:

- Controlar vigencia de equipos.
- Programar reposiciones preventivas.
- Analizar accidentes asociados a EPP no adecuado.

Se fortalece así el enfoque preventivo del sistema.

Gestión de acciones correctivas

La tabla *accioncorrectiva* cierra el ciclo de mejora continua bajo el enfoque PHVA (Planificar–Hacer–Verificar–Actuar).

Permite:

- Asignar responsables.
- Establecer fechas de compromiso.
- Controlar estado de ejecución.
- Registrar fecha de cierre.

Este componente transforma el sistema en una herramienta activa de gestión, ya que no solo documenta el accidente, sino que impulsa acciones concretas para evitar su recurrencia.

4.7.5 Consideraciones estratégicas para el Ingeniero Industrial

Desde la perspectiva de la optimización de procesos y transformación digital, este modelo ofrece ventajas significativas:

a. Generación de Indicadores (KPIs)

El sistema permite automatizar el cálculo de:

- Índice de Frecuencia (IF).
- Índice de Severidad (IS).
- Días perdidos acumulados.
- Costo total por accidentes.
- % acciones correctivas cerradas.
- Accidentes por área o puesto.

b. Toma de decisiones basada en datos

La integración entre accidentes, capacitación, EPP y acciones correctivas permite análisis cruzados que apoyan decisiones estratégicas preventivas.

c. Escalabilidad

El modelo está estructurado para crecer sin comprometer integridad referencial, permitiendo incorporar nuevos catálogos o indicadores.

4.7.6 Conclusión

El modelo de datos del sistema SSOMA de la Minera ZET no se limita al registro de accidentes, sino que articula una arquitectura integral orientada a:

- Investigación estructurada.
- Prevención basada en evidencia.
- Cumplimiento normativo.
- Mejora continua.

Desde la óptica del ingeniero industrial, este modelo representa la transformación de un proceso tradicional en papel hacia un sistema digital capaz de generar información estratégica para la gestión organizacional.

4.8 Modelado de seguridad y control de acceso

Una vez definido el modelo funcional del sistema SSOMA, es necesario incorporar una capa adicional de diseño orientada a la seguridad de la información y el control de acceso.

Todo sistema web que gestiona información sensible —como datos personales, registros de accidentes laborales y acciones correctivas—

debe garantizar que el acceso a la información esté restringido según el perfil y responsabilidad del usuario.

En esta etapa, el modelado ya no se centra únicamente en el proceso operativo, sino en la protección y gobernanza del sistema.

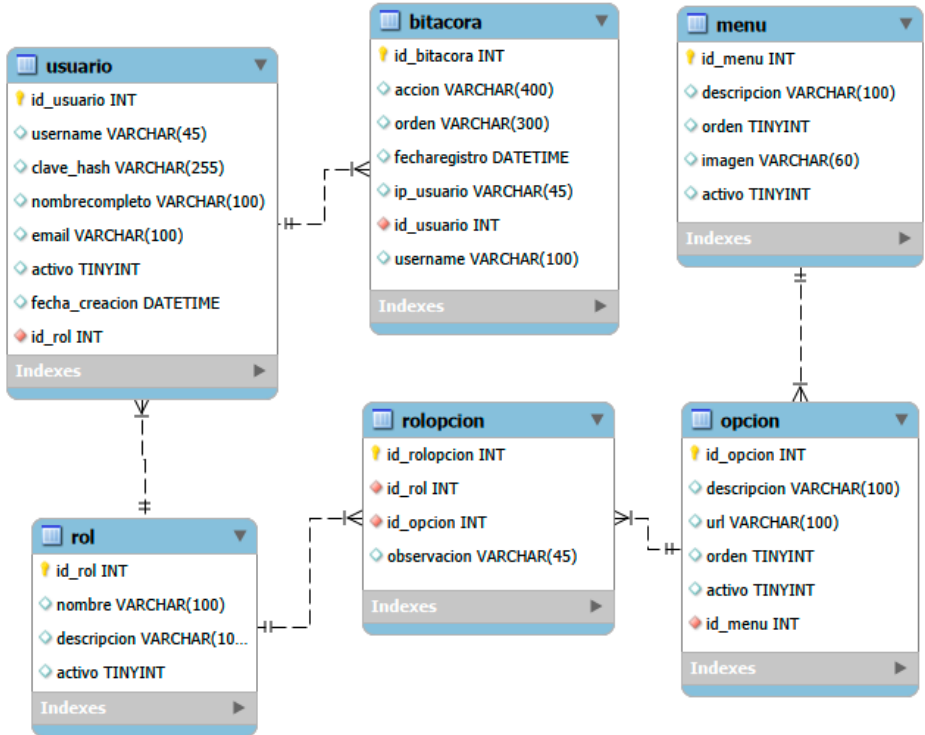


Fig. 7 Modelo de datos del control de usuarios SSOMA Minera ZET

4.8.1 Modelado de seguridad

El sistema SSOMA de la Minera ZET maneja información crítica:

- Datos personales de trabajadores.

- Información de accidentes.
- Costos asociados a incidentes.
- Acciones correctivas estratégicas.

Por lo tanto, es indispensable implementar un modelo de control de acceso basado en:

- Autenticación (¿quién es el usuario?)
- Autorización (¿qué puede hacer?)
- Trazabilidad (¿qué acciones realizó?)

4.8.2 Análisis del modelo de control de acceso y seguridad del sistema SSOMA de la Minera ZET

Este modelo implementa un esquema de Control de Acceso Basado en Roles (RBAC), permitiendo una administración granular de los permisos dentro de la aplicación web industrial.

a. Gestión de identidad y autenticación

Tabla *usuario*: Es el repositorio central de credenciales. Almacena datos críticos como el nombreusuario, el email y, fundamentalmente, la clave_hash. El uso de hashes en lugar de texto plano es vital para la seguridad de la información en la empresa.

Estado de Cuenta: El campo activo permite deshabilitar el acceso de forma inmediata (por ejemplo, en caso de cese del trabajador) sin necesidad de borrar sus registros históricos de auditoría.

b. Estructura de roles y permisos

El sistema no asigna permisos directamente a los usuarios, sino que utiliza una capa intermedia de abstracción para facilitar la escalabilidad:

Tabla rol: Define los perfiles de usuario dentro del sistema SSOMA (ej. "administrador", "supervisor de planta", "analista de seguridad").

Tabla opcion y menu: Representan las funcionalidades de la aplicación web. Las opciones se agrupan en menús (ej. Menú "Reportes" con opciones como "Lista de Accidentes" o "Kardex de EPP") para organizar la interfaz de usuario de forma lógica.

Tabla rolopcion: Es la tabla relacional que define exactamente qué puede hacer cada rol. Por ejemplo, define que el rol "Supervisor" tiene acceso a la opción "Registrar Accidente", pero no a la de "Configuración del Sistema".

c. Trazabilidad y auditoría

Tabla bitacora: En un entorno industrial, la trazabilidad es un requisito legal y normativo. Esta tabla registra cada acción realizada por los usuarios, capturando la fecha y hora exacta (fecharegistro) y la dirección IP desde la cual se realizó la operación (ip_usuario). Esto permite saber quién, cuándo y desde dónde se modificó un dato sensible del modelo SSOMA.

Resumen de Parte II: Bases de datos para procesos industriales

Esta parte aborda la base de datos como el núcleo estratégico del sistema, no como un mero repositorio técnico. Se explica su función esencial: estructurar la información, soportar la operación, garantizar trazabilidad, alimentar indicadores y permitir escalabilidad. A partir del caso Minera ZET, se desarrolla un modelo relacional completo que integra la gestión de accidentes, trabajadores, áreas, capacitaciones, EPP y acciones correctivas. Se muestra cómo las entidades identificadas en el análisis del proceso se convierten en tablas interrelacionadas que reflejan fielmente la realidad operativa. Además, se incorpora un módulo de seguridad basado en roles (RBAC) con usuarios, permisos y bitácora, transformando el modelo en una base institucional sólida que permite automatizar el cálculo de indicadores de gestión.

PARTE III. Desarrollo Web con PHP para sistemas industriales**CAPÍTULO 5. Desarrollo de Interfaces Web para Sistemas Industriales****5.1 Introducción**

Una aplicación web no solo debe funcionar correctamente desde el punto de vista lógico y estructural, sino también presentar una interfaz clara, usable y alineada con el entorno operativo de la organización.

En sistemas industriales, la interfaz no es un elemento decorativo: es un componente estratégico que impacta directamente en la eficiencia del usuario, la calidad del registro de datos y la toma de decisiones.

En este capítulo se desarrollan los fundamentos del diseño de interfaces utilizando HTML y CSS, integrándolos posteriormente con PHP.

5.2 Arquitectura de una aplicación web

Antes de diseñar la interfaz, es importante comprender la separación de capas:

- HTML → Estructura del contenido
- CSS → Presentación y diseño visual
- PHP → Lógica del servidor
- MySQL → Almacenamiento de datos

Esta separación permite construir sistemas mantenibles y escalables.

5.3 Fundamentos de HTML

HTML (HyperText Markup Language) define la estructura de la página web.

Elementos básicos

HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Sistema SSOMA - Minera ZET</title>
</head>
<body>
  <h1>Registro de Accidentes</h1>
  <form>
    <label>Fecha:</label>
    <input type="date">
    <label>Descripción:</label>
    <textarea></textarea>
    <button type="submit">Guardar</button>
  </form>
</body>
</html>
```

Componentes clave para sistemas industriales

- Formularios estructurados
- Tablas de datos
- Botones de acción

- Campos obligatorios
- Selectores dinámicos

5.4 Diseño visual con CSS

CSS (Cascading Style Sheets) controla la apariencia del sistema.

Ejemplo básico

CSS

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f6f9;  
}  
h1 {  
    color: #1f3c88;  
}  
form {  
    background: white;  
    padding: 20px;  
    border-radius: 8px;  
}
```

5.5 Principios de diseño para sistemas industriales

En entornos industriales se recomienda:

- Simplicidad. Evitar sobrecarga visual.
- Claridad. Campos bien etiquetados.
- Jerarquía visual. Titulares, subtítulos y secciones diferenciadas.

- Validación visible. Mensajes claros de error.
- Diseño responsivo. Acceso desde tablet o móvil en campo.

5.6 Uso de frameworks CSS

En el desarrollo de interfaces web modernas, es recomendable utilizar frameworks CSS que faciliten la construcción de diseños consistentes, responsivos y visualmente profesionales. Estos frameworks proporcionan componentes predefinidos, sistemas de rejilla (grid system) y utilidades que aceleran el proceso de desarrollo.

Entre los más utilizados se encuentran:

- Bootstrap
- Tailwind CSS

5.6.1 Bootstrap

Bootstrap es un framework CSS ampliamente adoptado que incluye estilos preconfigurados, componentes reutilizables y un sistema de diseño responsivo basado en rejillas. Permite desarrollar interfaces uniformes sin necesidad de diseñar todos los estilos desde cero.

Ejemplo con Bootstrap:

HTML

```
<button class="btn btn-primary">Guardar</button>
```

En este ejemplo, las clases `btn` y `btn-primary` aplican automáticamente estilos prediseñados, incluyendo color, espaciado y comportamiento visual.

5.6.2 Beneficios del uso de frameworks CSS

- La incorporación de frameworks como Bootstrap o Tailwind CSS aporta ventajas significativas en el desarrollo de sistemas industriales:
- Mejora estética: Interfaces visualmente profesionales y coherentes.
- Responsividad: Adaptación automática a diferentes dispositivos (computadoras, tablets y móviles).
- Productividad: Reducción del tiempo de desarrollo mediante componentes reutilizables.
- Consistencia visual: Uniformidad en botones, formularios y tablas.
- Escalabilidad: Facilita el mantenimiento y expansión futura del sistema.

5.7 Buenas prácticas de interfaz

- Uso de colores coherentes (verde = seguro, rojo = alerta).
- Minimizar pasos para registrar un evento.
- Separar módulos por funcionalidad.
- Respetar consistencia visual.

CAPÍTULO 6. Fundamentos de Programación en PHP

6.1 ¿Qué es PHP y cómo funciona?

PHP (Hypertext Preprocessor) es un lenguaje de programación interpretado del lado del servidor, diseñado específicamente para el desarrollo de aplicaciones web dinámicas. Su ejecución se realiza en el servidor web (por ejemplo, Apache o Nginx con PHP-FPM), donde el código es procesado antes de enviarse el resultado al navegador del cliente en forma de HTML.

Una de sus principales características es que es un lenguaje embebido en HTML, lo que permite insertar bloques de código PHP dentro de documentos HTML mediante delimitadores especiales (`<?php ... ?>`). Esta integración facilita la combinación entre la lógica de negocio y la presentación, permitiendo generar contenido dinámico en función de datos provenientes de formularios, sesiones de usuario o bases de datos como MySQL.

Desde el punto de vista arquitectónico, PHP actúa como capa intermedia entre la interfaz (HTML/CSS) y la base de datos, gestionando la validación de datos, el procesamiento de reglas de negocio y la comunicación con el sistema gestor de bases de datos.

Flujo básico:

1. El usuario envía información desde un formulario.
2. El servidor procesa la información con PHP.
3. PHP interactúa con MySQL.

4. Se genera una respuesta dinámica.

6.2 Estructura básica de un archivo PHP

Ejemplo:

PHP
<pre><?php echo "Bienvenido al Sistema SSOMA de Minera ZET"; ?> Todo el código PHP se escribe entre: <?php // código ?></pre>

6.3 Variables y tipos de datos

Ejemplo:

PHP
<pre><?php \$producto = "Harina"; \$cantidad = 100; \$precio = 45.50; ?></pre>

Buenas prácticas:

- Usar nombres descriptivos.
- Evitar variables ambiguas.

6.4 Estructuras Condicionales

Ejemplo:

PHP

```
<?php
$stock = 10;
if ($stock < 20) {
    echo "Stock bajo";
} else {
    echo "Stock suficiente";
}
?>
```

6.5 Bucles

Ejemplo:

PHP

```
<?php
for ($i = 1; $i <= 5; $i++) {
    echo "Orden número: " . $i . "<br>";
}
?>
```

Se usan para:

- Mostrar listados
- Procesar registros
- Generar reportes

6.6 Integración HTML + PHP

Ejemplo:

PHP

```
<select name="area">

<?php
$sql = "SELECT * FROM area WHERE activo = 1";
$resultado = $conn->query($sql);
```

```

while($fila = $resultado->fetch_assoc())
{
    echo          "<option          value='\".$fila['id_area'].\"'>
\".$fila['descripcion'].\"</option>\";
}
?>

</select>

```

6.7 Tablas dinámicas de consulta

PHP
<pre> <table> <tr> <th>Fecha</th> <th>Área</th> <th>Acción</th> </tr> <?php while(\$fila = \$resultado->fetch_assoc()){ echo "<tr>"; echo "<td>\".\$fila['fechaaccidente'].\"</td>\"; echo "<td>\".\$fila['area'].\"</td>\"; echo "<td>Ver</t d>\"; echo "</tr>\"; } ?> </table> </pre>

6.8 Formularios HTML y Método POST

Ejemplo

HTML

```
<select name="area">

<form method="POST" action="procesar.php">
    Producto: <input type="text" name="producto">
    Cantidad: <input type="number" name="cantidad">
    <button type="submit">Registrar</button>
</form>
```

PHP (archivo: procesar.php)

```
<?php
$producto = $_POST['producto'];
$cantidad = $_POST['cantidad'];
echo "Producto: " . $producto . "<br>";
echo "Cantidad: " . $cantidad;
?>
```

Fond
Editorial

CAPÍTULO 7. Conexión entre PHP y MySQL

7.1 Conexión a la base de datos

Archivo conexion.php:

PHP
<pre><?php \$host = "localhost"; \$usuario = "root"; \$password = ""; \$base_datos = "minera_ZET"; \$conn = new mysqli(\$host, \$usuario, \$password, \$base_datos); if (\$conn->connect_error) { die("Error de conexión: " . \$conn->connect_error); } ?></pre>

Buenas prácticas:

Centralizar la conexión en un solo archivo.

Incluirlo en cada módulo.

7.2 Gestionar transacciones

7.2.1 Insertar registros

PHP
<pre><?php include("conexion.php"); /* Recibir datos del formulario */ \$fecha = \$_POST['fechaaccidente']; \$descripcion = \$_POST['descripcion']; \$id_area = \$_POST['id_area']; /* Insertar accidente */ \$sql = "INSERT INTO accidente (fechaaccidente, descripcion, id_area) VALUES (?, ?, ?)"; \$stmt = \$conn->prepare(\$sql);</pre>

```

$stmt->bind_param("ssi", $fecha, $descripcion,
$id_area);
if($stmt->execute()){
    echo "Accidente registrado correctamente";
}else{
    echo "Error: " . $conn->error;
}
?>

```

7.2.2 Mostrar registros

PHP

```

<?php
include("conexion.php");
$sql = "SELECT
        ac.fechaaccidente, ac.descripcion,
        a.descripcion AS area
        FROM accidente ac
        INNER JOIN area a
        ON ac.id_area = a.id_area";
$resultado = $conn->query($sql);
while ($fila = $resultado->fetch_assoc()) {
    echo $fila['fechaaccidente'] . " - " .
        $fila['area'] . " - " .
        $fila['descripcion'] . "<br>";
}
?>

```

7.2.3 Editar registros

PHP

```

<?php
include("conexion.php");
/* Datos a actualizar */
$id_accidente = 1;
$id_estado = 0;
/* Actualizar estado del accidente */
$sql = "UPDATE accidente
        SET id_estado = ?
        WHERE id_accidente = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ii", $id_estado, $id_accidente);

```

```
$stmt->execute();  
echo "Accidente actualizado correctamente";  
?>
```

7.2.4 Eliminar registros

PHP

```
<?php  
include("conexion.php");  
/* Id del accidente a eliminar */  
$id_accidente = 5;  
/* Eliminar registro */  
$sql = "DELETE FROM accidente  
        WHERE id_accidente = ?";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("i", $id_accidente);  
$stmt->execute();  
echo "Accidente eliminado correctamente";  
?>
```

FEPOL
Fond
Editorial

CAPÍTULO 8. Desarrollo del sistema SSOMA de la minera ZET

El proyecto se desarrolla bajo un enfoque pedagógico orientado a la comprensión estructural del sistema. Por ello, el diseño visual y la programación se presentan en un nivel básico controlado, suficiente para demostrar la funcionalidad del modelo, pero sin incorporar aún estándares empresariales avanzados de arquitectura, seguridad o escalabilidad.

En etapas posteriores de formación profesional, el sistema podría evolucionar hacia una solución de mayor complejidad técnica.

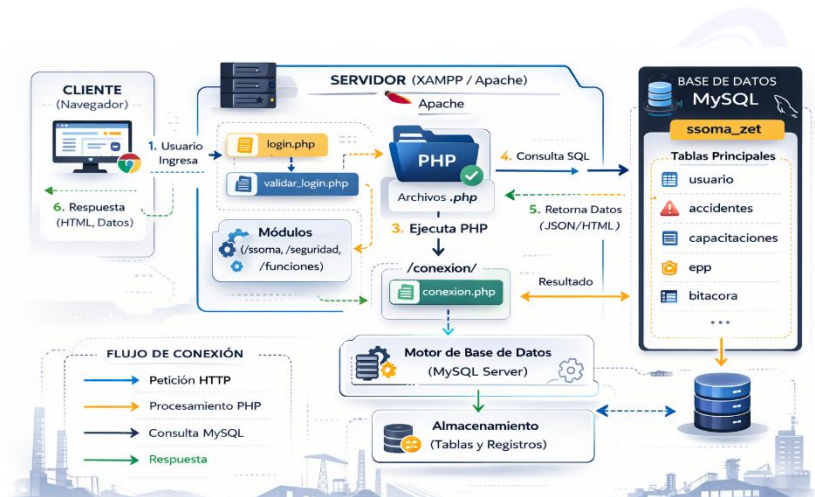


Fig. 8 Conexión PHP / Mysql dentro del servidor minera ZET

8.1 Organización del proyecto

Una adecuada organización del proyecto es un elemento clave en el desarrollo de cualquier sistema web. Más allá de la programación, la estructura de carpetas y archivos determina la mantenibilidad, escalabilidad y claridad del código.

En el caso del sistema SSOMA de la Minera ZET, se propone una organización modular que permita separar responsabilidades y facilitar futuras ampliaciones del sistema.



Fig. 9 Estructura de carpetas del proyecto

/css

Gestión de estilos y presentación visual. Este directorio almacena los archivos de hojas de estilo (CSS) que controlan la apariencia del sistema.

/funciones

Biblioteca de funciones generales del sistema. Este directorio centraliza las funciones reutilizables que pueden ser invocadas desde distintos módulos del sistema. Su propósito es evitar duplicación de código y promover una programación más ordenada y mantenible.

/imagenes

Recursos gráficos del sistema. Contiene los elementos visuales utilizados en la interfaz.

/layout

Plantillas y componentes reutilizables de la interfaz del sistema. Este directorio contiene los archivos que definen la estructura común de las páginas, como el encabezado (header), el menú lateral (sidebar), el pie de página (footer) y otros elementos de diseño compartidos. Su objetivo es mantener una presentación uniforme en todas las páginas del sistema y facilitar el mantenimiento del código, evitando la duplicación de elementos visuales en cada archivo.

/conexion

Configuración de acceso a base de datos. Este directorio almacena los archivos relacionados con la conexión a la base de datos.

/seguridad

Módulo de autenticación y control de acceso. Este directorio contiene los archivos responsables de gestionar el acceso al sistema y garantizar la protección de la información. Aquí se implementan los mecanismos de autenticación, autorización y auditoría, alineados con el modelo de roles previamente diseñado en la base de datos.

/ssoma

Módulo funcional del sistema. Este directorio contiene los archivos principales del sistema SSOMA.

8.2 Implementación del menú del sistema

El menú no es solo navegación. Es una representación estructural del proceso digitalizado.

Cuando el menú refleja correctamente la organización del proceso SSOMA, el sistema se vuelve intuitivo y alineado con la realidad operativa.



Fig. 10 Menú sistema SSOMA Minera Zet

A continuación, se generarán las opciones funcionales del sistema a través de un script SQL ejecutado en MySQL. Este procedimiento permitirá poblar las tablas correspondientes a módulos, menús o permisos, sentando las bases para la configuración del control de acceso y la estructura de navegación del sistema.

El código completo para poblar la data de roles, menú, opciones y opciones por rol puede consultarse en el *Anexo 2*.

8.3 Implementación del inicio de sesión

El inicio de sesión constituye el mecanismo de autenticación del sistema SSOMA de la minera ZET y representa el primer punto de control de seguridad. Su finalidad es verificar la identidad del usuario antes de permitir el acceso a los módulos internos del sistema.

Desde una perspectiva técnica, este proceso implica la validación de credenciales almacenadas en la base de datos mediante la comparación segura de contraseñas cifradas. Desde una perspectiva organizacional, garantiza que cada usuario acceda únicamente a las funcionalidades permitidas según su rol dentro de la empresa.

Con fines de validación funcional y pruebas integrales, se configurará un usuario con rol de Administrador, dotado de privilegios totales sobre el sistema. Este perfil permitirá evaluar la correcta interacción entre los módulos, comprobar la asignación de permisos y asegurar la coherencia operativa del sistema SSOMA en su fase académica de desarrollo.

Con el fin de facilitar las pruebas del sistema, se creará un usuario inicial mediante un script SQL, el cual tendrá asignado el rol de Administrador. Este rol posee acceso a todas las opciones del sistema, lo que permitirá explorar y validar el funcionamiento completo de los módulos desarrollados.

MySQL
<pre>##crear usuario INSERT INTO usuario (username, clave_hash, nombrecompleto, email, activo, fecha_creacion, id_rol) VALUES ('juliozet','\$2y\$10\$ptIep9Clgp6vjmMzu9bEqOLjCYGmrWc/iETjvs8RQ nwnZNEwnfjsG', 'Julio César Álvarez Reyes', 'juliozet@gmail.com', 1, NOW(), 1); ## usuario: juliozet ##password: admin123</pre>

8.3.1 Estructura del módulo de seguridad

El proceso de autenticación implementado sigue las siguientes etapas:

- El usuario ingresa su nombre de usuario y contraseña en el formulario login.php.
- El formulario envía los datos mediante el método POST al archivo validar_login.php.
- El servidor ejecuta una consulta preparada para buscar el usuario activo en la base de datos.
- Se utiliza la función password_verify() para comparar la contraseña ingresada con el hash almacenado.

- Si las credenciales son correctas:
 - Se crea una sesión con `$_SESSION`.
 - Se almacenan los datos básicos del usuario (id, nombre, rol).
 - Se redirige a la pagina inicial del sistema `dashboard.php`
- Si las credenciales son incorrectas:
 - Se retorna al formulario con un mensaje de error.

Este flujo permite separar claramente la capa de presentación (formulario) de la lógica de validación.

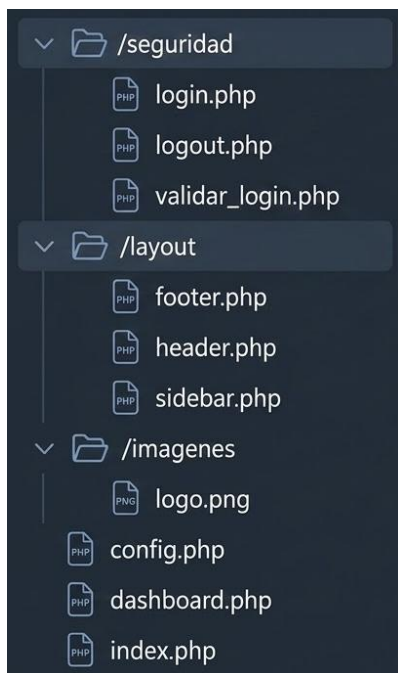


Fig. 11 Estructura del módulo de seguridad

Para fines de consulta y análisis, los archivos que conforman el módulo de seguridad del sistema se presentan en el *Anexo 3*. En dicho anexo se incluyen los archivos encargados de gestionar el inicio de sesión, la validación de usuarios y el cierre de sesión del sistema.

8.3.2 Seguridad implementada

La implementación incluye buenas prácticas fundamentales:

- Uso de `password_hash()` para almacenar contraseñas cifradas.
- Uso de `password_verify()` para validar credenciales.
- Uso de consultas preparadas (Prepared Statements) para evitar inyección SQL.
- Validación de usuarios activos.
- Manejo de sesiones mediante `session_start()`.

Estas medidas reducen significativamente los riesgos de vulnerabilidades comunes en aplicaciones web básicas.

8.3.3 Manejo de sesiones

Una vez autenticado, el sistema mantiene la información del usuario en variables de sesión:

PHP
<pre>\$_SESSION['id_usuario'] \$_SESSION['username'] \$_SESSION['rol']</pre>

Esto permite:

- Restringir acceso a páginas internas.
- Mostrar menús dinámicos según el rol.
- Registrar acciones en la bitácora.
- Mantener coherencia en la navegación.

8.3.4 Control de acceso a páginas internas

Cada módulo del sistema debe verificar que exista una sesión activa antes de ejecutarse:

PHP
<pre>if(!isset(\$_SESSION['id_usuario'])){\n header("Location: ../seguridad/login.php");\n exit();\n}</pre>



Fig. 12 Interfaz de inicio al Sistema SSOMA de la minera ZET

8.4 Interfaz principal

La interfaz principal del sistema constituye el punto de acceso central a las funcionalidades de la aplicación web desarrollada para la gestión de seguridad y salud ocupacional en el caso de estudio de la Minera ZET. Esta interfaz ha sido diseñada siguiendo principios de usabilidad y organización modular, permitiendo que los usuarios naveguen de manera sencilla entre los distintos módulos del sistema.

Como se observa la figura, la estructura de la interfaz se compone de tres elementos principales. En la parte superior se encuentra la barra de navegación o topbar, donde se muestra la identidad del sistema y la información del usuario autenticado. En el lado izquierdo se ubica el menú lateral dinámico, el cual presenta las opciones disponibles según el rol del usuario dentro del sistema. Finalmente, en la sección central se encuentra el área de contenido, donde se cargan las distintas páginas funcionales del sistema, tales como el registro de accidentes, el control de equipos de protección personal, las capacitaciones y los reportes de indicadores.

Esta arquitectura de interfaz es común en aplicaciones empresariales modernas, ya que permite separar la navegación, la identidad del sistema y el contenido operativo, facilitando la escalabilidad y el mantenimiento de la aplicación. Además, el menú lateral se genera dinámicamente a partir de los permisos definidos en la base de datos, lo que garantiza que cada usuario visualice únicamente las funcionalidades que le corresponden de acuerdo con su rol.

Desarrollo de Aplicaciones web para Ingeniería industrial

De esta manera, la interfaz principal actúa como el entorno de trabajo desde el cual los usuarios interactúan con los distintos procesos digitales implementados en el sistema.



Fig. 13 Interfaz principal del Sistema SSOMA Minera Zet

8.5 Desarrollo de formularios para tablas de catálogo y tablas de proceso

En las aplicaciones web empresariales es necesario desarrollar diferentes tipos de formularios para gestionar la información almacenada en la base de datos. En términos generales, los formularios pueden clasificarse en dos categorías principales: formularios para tablas de catálogo y formularios para tablas de proceso. Cada uno cumple una función distinta dentro del sistema y presenta características particulares en su diseño e implementación.

Con fines pedagógicos y con el propósito de ilustrar la implementación práctica de los componentes del sistema, en este libro se desarrollarán dos formularios clave: el formulario de catálogo de gestión de roles, orientado a la administración de perfiles y permisos de usuario, y el formulario de registro de accidentes, destinado al registro y seguimiento de incidentes. Asimismo, para facilitar las pruebas y el funcionamiento del sistema, las tablas de la base de datos serán pobladas mediante un script de carga de datos, el cual se presenta en el *Anexo 4*.

Formularios para la administración de tablas de catálogo

Los formularios asociados a tablas de catálogo tienen como objetivo permitir la administración de datos de referencia utilizados por distintos módulos del sistema. Estos datos suelen cambiar con poca frecuencia y son necesarios para estandarizar la información registrada en los procesos del sistema.

Entre las principales operaciones que se realizan sobre las tablas de catálogo se encuentran:

- Registrar nuevos valores del catálogo.
- Modificar registros existentes.
- Eliminar registros existentes.
- Visualizar los datos registrados.

Por ejemplo, en el sistema SSOMA desarrollado para el caso de estudio de la Minera ZET, existen tablas de catálogo como tipo de

accidente, área, gravedad o tipo de equipo de protección personal. Para administrar esta información se desarrolla un formulario sencillo que permite registrar o modificar los valores del catálogo.

Este tipo de formulario generalmente contiene campos básicos como:

- Identificador del registro
- Descripción del elemento del catálogo
- Estado del registro (activo o inactivo)

Debido a que los datos de catálogo son relativamente simples, la interfaz de estos formularios suele ser compacta y orientada principalmente al mantenimiento de información.

Para facilitar la comprensión del desarrollo del sistema, el código fuente del formulario de gestión de roles se presenta íntegramente en el *Anexo 5*, permitiendo al lector analizar su estructura y funcionamiento.

Sistema SSOMA

Administración

Gestión de accidentes

Capacitaciones

Control de EPPs

Acciones correctivas

Gestión de Roles

Nuevo Rol

ID	Nombre	Descripción	Estado	Acciones
1	Administrador	Tiene acceso a todas las opciones.	Activo	
2	Reportes	Sólo tiene acceso a los reportes del sistema	Inactivo	
3	Supervisor	Tiene acceso a la gestión de accidentes	Activo	

Fig. 14 Formulario para tabla de catálogo “roles”

Formularios para tablas de proceso

A diferencia de los formularios de catálogo, los formularios asociados a tablas de proceso se utilizan para registrar las operaciones o eventos que ocurren dentro de la organización. Estos formularios suelen ser más complejos, ya que deben integrar información proveniente de diversas tablas de catálogo y capturar múltiples datos relacionados con el proceso que se está registrando.

En el caso del sistema SSOMA, uno de los procesos más importantes es el registro de accidentes laborales. Este proceso requiere capturar información detallada sobre el evento ocurrido, lo que implica la utilización de diversos campos y la selección de valores provenientes de diferentes tablas de catálogo.

El formulario de Gestión de Accidentes permite administrar los accidentes registrados en el sistema SSOMA. En la parte superior se muestra un filtro por año, que permite visualizar únicamente los accidentes ocurridos en el período seleccionado, y un botón “Nuevo Accidente” para registrar un nuevo evento.

En la parte central se presenta una tabla con la lista de accidentes, donde se muestran el Id, la fecha, la descripción del accidente y su estado dentro del proceso de gestión.

Finalmente, en la columna Acciones se incluyen tres botones representados por iconos que permiten editar el accidente, gestionar los trabajadores involucrados y eliminar el registro. De esta manera,

Desarrollo de Aplicaciones web para Ingeniería industrial

el formulario facilita el mantenimiento y control de los accidentes registrados en el sistema.

Para fines de revisión y aprendizaje, el código fuente relacionado con la gestión de accidentes se presenta en el *Anexo 6*.

Sistema SSOMA juliozet | Salir

Administración
Gestión de accidentes
Capacitaciones
Control de EPPs
Acciones correctivas
Reporte

Gestión de Accidentes

Filtrar por Año
2026 Nuevo Accidente

Id	Fecha	Descripción	Estado	Acciones
7	08-03-2026	Caída de escalera	Registrado	
3	15-01-2026	Corte con metal	En investigación	
2	12-01-2026	Golpe con herramienta	Registrado	
1	10-01-2026	Caída por piso mojado	Registrado	

Fig. 15 Formulario principal de gestión de accidentes

Sistema SSOMA juliozet | Salir

Administración
Gestión de accidentes
Capacitaciones
Control de EPPs
Acciones correctivas
Reporte

Registro de Accidente

Fecha: 08/03/2026 Hora: 12:59:00 p.m. Área: Producción

Tipo de Accidente: Caída de altura Turno: Turno mañana Estado: Registrado

Descripción del accidente
Caída de escalera

Causas del accidente
Causa raíz: Falta de entrenamiento Condición peligrosa: Piso resbaloso
Causa inmediata: Falta de capacitación Acto inseguro: No usar EPP
Trabajador que informó el accidente: Vargas Luis

Guardar Cancelar

Fig. 16 Formulario de edición de accidentes

CAPÍTULO 9. Consultas para reportes e indicadores

9.1 Importancia de los reportes en los sistemas de información

Una de las principales finalidades de un sistema de información es transformar los datos operativos registrados en el sistema en información útil para la toma de decisiones. En el contexto de la seguridad y salud ocupacional, los reportes permiten analizar la ocurrencia de accidentes, identificar patrones de riesgo y evaluar el desempeño de las acciones preventivas implementadas en la organización.

Los reportes pueden mostrar información como el número de accidentes ocurridos en un periodo determinado, las áreas donde se presentan con mayor frecuencia, los tipos de accidentes más comunes o el impacto económico generado por estos eventos. Esta información resulta fundamental para que los responsables de la gestión de seguridad puedan implementar medidas correctivas y preventivas basadas en evidencia.

En este capítulo se presentan algunas consultas SQL que permiten generar reportes básicos e indicadores de gestión a partir de los datos almacenados en el sistema desarrollado.

Para efectos ilustrativos, las consultas presentadas en este capítulo utilizan como referencia los accidentes registrados durante el año 2026. El uso de este periodo tiene únicamente fines de ejemplo,

pudiendo adaptarse fácilmente a otros años o rangos de fechas según las necesidades de análisis de la organización.

9.1.1 Accidentes registrados

Una consulta frecuente consiste en conocer el número total de accidentes ocurridos durante el año 2026.

MySQL
<pre>SELECT COUNT(*) AS total_accidentes FROM accidente WHERE YEAR(fechaaccidente) = 2026;</pre>

9.1.2 Accidentes por área de trabajo

Este reporte permite identificar las áreas donde se presentan más accidentes.

MySQL
<pre>SELECT a.descripcion AS area, COUNT(ac.id_accidente) AS total_accidentes FROM accidente ac INNER JOIN area a ON ac.id_area = a.id_area WHERE YEAR(ac.fechaaccidente) = 2026 GROUP BY a.descripcion ORDER BY total_accidentes DESC;</pre>

9.1.3 Accidentes por tipo de accidente

Este análisis permite conocer cuáles son los tipos de accidentes más frecuentes.

MySQL

```
SELECT
t.descripcion AS tipo_accidente, COUNT(ac.id_accidente)
AS total
FROM accidente ac INNER JOIN tipoaccidente t ON
ac.id_tipoaccidente = t.id_tipoaccidente
WHERE YEAR(ac.fechaaccidente) = 2026
GROUP BY t.descripcion
ORDER BY total DESC;
```

9.1.4 Accidentes por estado

Esta consulta permite conocer el estado actual de los accidentes registrados.

MySQL

```
SELECT
e.descripcion AS estado,
COUNT(ac.id_accidente) AS total
FROM accidente ac
INNER JOIN estado e
ON ac.id_estado = e.id_estado
WHERE YEAR(ac.fechaaccidente) = 2026
GROUP BY e.descripcion;
```

9.1.5 Trabajadores involucrados en accidentes

Esta consulta permite identificar qué trabajadores han estado involucrados en accidentes durante el año.

MySQL

```
SELECT
t.apellidopaterno, t.nombre, COUNT(d.id_accidente) AS
total_accidentes
FROM accidente_detalle d INNER JOIN trabajador t
ON d.id_trabajador = t.id_trabajador
INNER JOIN accidente ac ON d.id_accidente =
ac.id_accidente
WHERE YEAR(ac.fechaaccidente) = 2026
GROUP BY t.id_trabajador
ORDER BY total_accidentes DESC;
```

9.1.6 Días perdidos por accidentes

Uno de los indicadores más importantes en la gestión de seguridad es el número de días perdidos a causa de accidentes laborales.

MySQL

```
SELECT
SUM(diaperdido) AS total_dias_perdidos
FROM accidente_detalle d
INNER JOIN accidente ac
ON d.id_accidente = ac.id_accidente
WHERE YEAR(ac.fechaaccidente) = 2026;
```

9.1.7 Costo estimado de los accidentes

El sistema también permite estimar el impacto económico de los accidentes ocurridos.

MySQL

```
SELECT
SUM(costoestimado) AS costo_total_accidentes
FROM accidente_detalle d
INNER JOIN accidente ac
ON d.id_accidente = ac.id_accidente
WHERE YEAR(ac.fechaaccidente) = 2026;
```

9.2 Indicadores de seguridad y salud ocupacional

Además de los reportes descriptivos presentados anteriormente, los sistemas de información también permiten calcular indicadores de gestión, los cuales son métricas que ayudan a evaluar el desempeño de la organización en materia de seguridad y salud en el trabajo.

Entre los indicadores más utilizados en la gestión de seguridad industrial se encuentran el Índice de Frecuencia, el Índice de Severidad y la Tasa de Incidencia. Estos indicadores permiten analizar la ocurrencia de accidentes y su impacto en la organización.

9.2.1 Índice de Frecuencia (IF)

El Índice de Frecuencia mide el número de accidentes ocurridos por cada millón de horas trabajadas durante un periodo determinado. Este

indicador permite evaluar la frecuencia con la que ocurren accidentes dentro de la organización.

La fórmula utilizada es:

$$IF = \frac{\text{Nº de accidentes} \times 1,000,000}{\text{Horas} - \text{hombre trabajadas}}$$

Si se considera, por ejemplo, que durante el año 2026 se trabajaron 200000 horas, el cálculo puede realizarse mediante la siguiente consulta:

MySQL

SELECT

(COUNT(*) * 1000000) / 200000 AS indice_frecuencia

FROM accidente

WHERE YEAR(fechaaccidente) = 2026;

Este indicador permite comparar el nivel de seguridad entre diferentes periodos o entre distintas áreas de la organización.

9.2.2 Índice de Severidad (IS)

El Índice de Severidad mide la severidad de los accidentes ocurridos, considerando el número de días perdidos como consecuencia de estos eventos.

La fórmula utilizada es:

$$IS = \frac{\text{Días perdidos} \times 1,000,000}{\text{Horas} - \text{hombre trabajadas}}$$

La consulta para calcular este indicador sería la siguiente:

MySQL
<pre>SELECT (SUM(diaperdido) * 1000) / 200000 AS indice_gravedad FROM accidente_detalle d INNER JOIN accidente ac ON d.id_accidente = ac.id_accidente WHERE YEAR(ac.fechaaccidente) = 2026;</pre>

Este indicador permite evaluar el impacto que los accidentes tienen en la productividad de la organización.

9.2.3 Tasa de Incidencia (TI)

La Tasa de Incidencia mide la cantidad de trabajadores que han sufrido accidentes en relación con el total de trabajadores de la organización.

La fórmula utilizada es:

$$TI = \frac{\text{Número de trabajadores accidentados} \times 100}{\text{Número total de trabajadores}}$$

La consulta SQL para calcular este indicador es:

MySQL
<pre>SELECT (COUNT(DISTINCT d.id_trabajador) * 100) / (SELECT COUNT(*) FROM trabajador WHERE activo = 1)</pre>

```
AS tasa_incidencia
FROM accidente_detalle d
INNER JOIN accidente ac
ON d.id_accidente = ac.id_accidente
WHERE YEAR(ac.fechaaccidente) = 2026;
```

Este indicador permite conocer el porcentaje de trabajadores que han estado involucrados en accidentes durante el periodo analizado.

9.3 Importancia de los indicadores en la gestión de seguridad

Los indicadores presentados permiten evaluar de manera cuantitativa el desempeño de la organización en materia de seguridad y salud ocupacional. A partir de estos indicadores es posible:

- Evaluar la evolución de los accidentes a lo largo del tiempo
- Identificar tendencias en la ocurrencia de incidentes
- Medir el impacto de las acciones preventivas implementadas
- Comparar el desempeño entre distintas áreas o periodos

En este sentido, los sistemas de información desempeñan un papel fundamental al permitir registrar, procesar y analizar los datos necesarios para la generación de estos indicadores, contribuyendo así a la mejora continua de la gestión de seguridad en la organización.

9.4 Uso de herramientas de inteligencia de negocios para la visualización de indicadores

Si bien las consultas SQL permiten generar reportes directamente desde la base de datos, en muchos casos es necesario presentar la información de manera más visual y dinámica. Para ello se utilizan herramientas de Inteligencia de Negocios (Business Intelligence) como Power BI o Tableau, las cuales permiten transformar los datos en tableros interactivos (dashboards) que facilitan el análisis de la información.

Estas herramientas están especialmente diseñadas para trabajar con grandes volúmenes de datos, permitiendo explorar, resumir y visualizar la información de manera eficiente. Para lograrlo, suelen utilizar modelos de datos analíticos, distintos a los modelos transaccionales utilizados en los sistemas operativos. Mientras que las bases de datos operacionales se orientan al registro y mantenimiento de la información, los modelos utilizados en herramientas de inteligencia de negocios se enfocan en facilitar el análisis y la generación de indicadores, generalmente mediante estructuras optimizadas para consultas analíticas.

CAPÍTULO 10. Proyecto integrado final

Como parte final del desarrollo presentado en este libro, se propone la elaboración de un proyecto integrado, cuyo propósito es aplicar de manera práctica los conceptos estudiados a lo largo de los capítulos anteriores. Este proyecto permitirá integrar conocimientos relacionados con el análisis de procesos, modelado de datos, desarrollo de sistemas de información y generación de reportes e indicadores.

El proyecto que cada grupo de estudiantes desarrollará estará basado en el proceso organizacional que fue seleccionado y analizado en el Capítulo 1. A partir de dicho proceso, el grupo deberá diseñar una solución informática que permita registrar la información relevante, organizar los datos generados por el proceso y generar reportes o indicadores que apoyen la toma de decisiones dentro de la organización. De esta manera, el proyecto busca que los estudiantes apliquen de forma integrada los conceptos estudiados a lo largo del libro, trabajando de manera colaborativa en el análisis, diseño y demostración del sistema propuesto.

La presentación del proyecto debe evidenciar la comprensión del proceso seleccionado y demostrar cómo los conceptos abordados en el libro se aplican para construir un sistema de información que contribuya a mejorar la gestión del proceso analizado.

Para ello, el proyecto deberá incluir los siguientes elementos.

- Explicación del proceso modelado
- Diagrama entidad–relación
- Demostración funcional del sistema
- Reportes e indicadores generados
- Beneficios esperados

Resumen Parte III: Desarrollo web con PHP para sistemas industriales

Esta última parte materializa todo lo aprendido en la construcción efectiva del sistema web. Se recorren los fundamentos de HTML, CSS y PHP, así como la conexión con MySQL mediante operaciones CRUD y consultas preparadas. Con esta base, se desarrolla paso a paso el sistema SSOMA de la Minera ZET: organización del proyecto, implementación del menú dinámico, módulo de autenticación segura con sesiones y control de acceso. Se construyen formularios tanto para catálogos como para procesos complejos, mostrando cómo integrar múltiples tablas y gestionar transacciones. Finalmente, se enseña a extraer valor de los datos mediante consultas SQL que generan reportes y calculan indicadores clave como índices de frecuencia y severidad. La obra culmina con un proyecto integrado donde los estudiantes desarrollan su propio sistema completo.

Anexo 1: Script de creación de la base de datos*MySQL*

```

/*
SQLyog Ultimate v11.11 (64 bit)
MySQL - 5.5.5-10.4.32-MariaDB : Database - ssoma_zet
*****
*****
*/

/*!40101 SET NAMES utf8 */;
/*!40101 SET SQL_MODE='*/;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
/*Table structure for table `accidente` */

DROP TABLE IF EXISTS `accidente`;

CREATE TABLE `accidente` (
  `id_accidente` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` text DEFAULT NULL,
  `fechaaccidente` date DEFAULT NULL,
  `horaaccidente` time DEFAULT NULL,
  `id_estado` tinyint(4) DEFAULT NULL,
  `id_tipoaccidente` tinyint(4) NOT NULL,
  `id_turno` tinyint(4) NOT NULL,
  `id_area` smallint(6) NOT NULL,
  `id_supervisor` char(8) NOT NULL,
  `id_causaraiz` tinyint(4) NOT NULL,
  `id_condicionpeligrosa` smallint(6) NOT NULL,
  `id_causainmediata` tinyint(4) NOT NULL,
  `id_actoinseguro` int(11) NOT NULL,
  `fecha_registro` datetime DEFAULT current_timestamp(),
  `usuario_registro` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_accidente`),
  KEY `id_turno` (`id_turno`),
  KEY `id_tipoaccidente` (`id_tipoaccidente`),
  KEY `id_causaraiz` (`id_causaraiz`),
  KEY `id_condicionpeligrosa` (`id_condicionpeligrosa`),
  KEY `id_causainmediata` (`id_causainmediata`),
  KEY `id_actoinseguro` (`id_actoinseguro`),
  KEY `id_area` (`id_area`),
  KEY `id_estado` (`id_estado`),
  KEY `id_supervisor` (`id_supervisor`),
  CONSTRAINT `accidente_ibfk_10` FOREIGN KEY
(`id_supervisor`) REFERENCES `trabajador` (`id_trabajador`),

```

```

CONSTRAINT `accidente_ibfk_2` FOREIGN KEY (`id_turno`)
REFERENCES `turno` (`id_turno`),
CONSTRAINT `accidente_ibfk_3` FOREIGN KEY
(`id_tipoaccidente`) REFERENCES `tipoaccidente`
(`id_tipoaccidente`),
CONSTRAINT `accidente_ibfk_4` FOREIGN KEY (`id_causaraiz`)
REFERENCES `causaraiz` (`id_causaraiz`),
CONSTRAINT `accidente_ibfk_5` FOREIGN KEY
(`id_condicionpeligrosa`) REFERENCES `condicionpeligrosa`
(`id_condicionpeligrosa`),
CONSTRAINT `accidente_ibfk_6` FOREIGN KEY
(`id_causainmediata`) REFERENCES `causainmediata`
(`id_causainmediata`),
CONSTRAINT `accidente_ibfk_7` FOREIGN KEY
(`id_actoinseguro`) REFERENCES `actoinseguro`
(`id_actoinseguro`),
CONSTRAINT `accidente_ibfk_8` FOREIGN KEY (`id_area`)
REFERENCES `area` (`id_area`),
CONSTRAINT `accidente_ibfk_9` FOREIGN KEY (`id_estado`)
REFERENCES `estado` (`id_estado`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

```

```

/*Table structure for table `accidente_detalle` */

```

```

DROP TABLE IF EXISTS `accidente_detalle`;

```

```

CREATE TABLE `accidente_detalle` (
  `id_accidente_detalle` int(11) NOT NULL AUTO_INCREMENT,
  `id_accidente` int(11) NOT NULL,
  `id_trabajador` char(8) NOT NULL,
  `id_gravedad` tinyint(4) NOT NULL,
  `diaperdido` smallint(6) NOT NULL,
  `costoestimado` decimal(9,2) DEFAULT NULL,
  PRIMARY KEY (`id_accidente_detalle`),
  KEY `id_gravedad` (`id_gravedad`),
  KEY `id_accidente` (`id_accidente`),
  KEY `id_trabajador` (`id_trabajador`),
  CONSTRAINT `accidente_detalle_ibfk_3` FOREIGN KEY
(`id_gravedad`) REFERENCES `gravedad` (`id_gravedad`),
  CONSTRAINT `accidente_detalle_ibfk_4` FOREIGN KEY
(`id_accidente`) REFERENCES `accidente` (`id_accidente`),
  CONSTRAINT `accidente_detalle_ibfk_5` FOREIGN KEY
(`id_trabajador`) REFERENCES `trabajador` (`id_trabajador`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

```

```

/*Table structure for table `accioncorrectiva` */

```

```

DROP TABLE IF EXISTS `accioncorrectiva`;

```

```

CREATE TABLE `accioncorrectiva` (

```

```

`id_accioncorrectiva` int(11) NOT NULL AUTO_INCREMENT,
`id_accidente` int(11) DEFAULT NULL,
`descripcion` text DEFAULT NULL,
`fecha_registro` datetime DEFAULT NULL,
`fecha_compromiso` date DEFAULT NULL,
`fecha_cierre` date DEFAULT NULL,
`observaciones` text DEFAULT NULL,
`id_estadoaccion` tinyint(4) NOT NULL,
`id_tipoaccion` smallint(6) NOT NULL,
`id_responsable` char(8) NOT NULL,
`activo` tinyint(4) DEFAULT NULL,
PRIMARY KEY (`id_accioncorrectiva`),
KEY `id_accidente` (`id_accidente`),
KEY `id_responsable` (`id_responsable`),
KEY `id_estadoaccion` (`id_estadoaccion`),
KEY `id_tipoaccion` (`id_tipoaccion`),
CONSTRAINT `accioncorrectiva_ibfk_1` FOREIGN KEY
(`id_accidente`) REFERENCES `accidente` (`id_accidente`),
CONSTRAINT `accioncorrectiva_ibfk_2` FOREIGN KEY
(`id_responsable`) REFERENCES `trabajador` (`id_trabajador`),
CONSTRAINT `accioncorrectiva_ibfk_3` FOREIGN KEY
(`id_estadoaccion`) REFERENCES `estadoaccion`
(`id_estadoaccion`),
CONSTRAINT `accioncorrectiva_ibfk_4` FOREIGN KEY
(`id_tipoaccion`) REFERENCES `tipoaccion` (`id_tipoaccion`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `actoinseguro` */

DROP TABLE IF EXISTS `actoinseguro`;

CREATE TABLE `actoinseguro` (
  `id_actoinseguro` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_actoinseguro`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `area` */

DROP TABLE IF EXISTS `area`;

CREATE TABLE `area` (
  `id_area` smallint(6) NOT NULL AUTO INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  `id_tipoarea` tinyint(4) NOT NULL,
  PRIMARY KEY (`id_area`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

```

```

/*Table structure for table `bitacora` */

DROP TABLE IF EXISTS `bitacora`;

CREATE TABLE `bitacora` (
  `id_bitacora` int(11) NOT NULL AUTO_INCREMENT,
  `accion` varchar(400) DEFAULT NULL,
  `orden` varchar(300) DEFAULT NULL,
  `fecharegistro` datetime DEFAULT current_timestamp(),
  `ip_usuario` varchar(45) DEFAULT NULL,
  `id_usuario` int(11) NOT NULL,
  PRIMARY KEY (`id_bitacora`),
  KEY `id_usuario` (`id_usuario`),
  CONSTRAINT `bitacora_ibfk_1` FOREIGN KEY (`id_usuario`)
REFERENCES `usuario` (`id_usuario`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `capacitacion` */

DROP TABLE IF EXISTS `capacitacion`;

CREATE TABLE `capacitacion` (
  `id_capacitacion` int(11) NOT NULL AUTO_INCREMENT,
  `fecha_inicio` date DEFAULT NULL,
  `fecha_fin` date DEFAULT NULL,
  `nombre` varchar(100) DEFAULT NULL,
  `temaprincipal` varchar(100) DEFAULT NULL,
  `id_curso` int(11) NOT NULL,
  `total_horas` smallint(6) DEFAULT NULL,
  `id_tipocapacitacion` int(11) NOT NULL,
  `id_modalidad` tinyint(4) NOT NULL,
  `id_estadocapacitacion` tinyint(4) NOT NULL,
  `certificacion` varchar(100) DEFAULT NULL,
  `instructor` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_capacitacion`),
  KEY `id_modalidad` (`id_modalidad`),
  KEY `id_tipocapacitacion` (`id_tipocapacitacion`),
  KEY `id_curso` (`id_curso`),
  KEY `id_estadocapacitacion` (`id_estadocapacitacion`),
  CONSTRAINT `capacitacion_ibfk_1` FOREIGN KEY
(`id_modalidad`) REFERENCES `modalidad` (`id_modalidad`),
  CONSTRAINT `capacitacion_ibfk_2` FOREIGN KEY
(`id_tipocapacitacion`) REFERENCES `tipocapacitacion`
(`id_tipocapacitacion`),
  CONSTRAINT `capacitacion_ibfk_4` FOREIGN KEY (`id_curso`)
REFERENCES `curso` (`id_curso`),
  CONSTRAINT `capacitacion_ibfk_5` FOREIGN KEY
(`id_estadocapacitacion`) REFERENCES `estadocapacitacion`
(`id_estadocapacitacion`)

```

```

) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `categoria` */

DROP TABLE IF EXISTS `categoria`;

CREATE TABLE `categoria` (
  `id_categoria` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_categoria`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `causainmediata` */

DROP TABLE IF EXISTS `causainmediata`;

CREATE TABLE `causainmediata` (
  `id_causainmediata` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_causainmediata`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `causaraiz` */

DROP TABLE IF EXISTS `causaraiz`;

CREATE TABLE `causaraiz` (
  `id_causaraiz` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  `id_categoria` tinyint(4) NOT NULL,
  PRIMARY KEY (`id_causaraiz`),
  KEY `id_categoria` (`id_categoria`),
  CONSTRAINT `causaraiz_ibfk_1` FOREIGN KEY (`id_categoria`)
REFERENCES `categoria` (`id_categoria`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `condicionpeligrosa` */

DROP TABLE IF EXISTS `condicionpeligrosa`;

CREATE TABLE `condicionpeligrosa` (
  `id_condicionpeligrosa` smallint(6) NOT NULL
  AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,

```



```

PRIMARY KEY (`id_condicionpeligrosa`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `curso` */

DROP TABLE IF EXISTS `curso`;

CREATE TABLE `curso` (
  `id_curso` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_curso`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `epp` */

DROP TABLE IF EXISTS `epp`;

CREATE TABLE `epp` (
  `id_epp` smallint(6) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `vidautilmeses` smallint(6) DEFAULT NULL,
  `obligatorio` char(2) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_epp`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `estado` */

DROP TABLE IF EXISTS `estado`;

CREATE TABLE `estado` (
  `id_estado` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_estado`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `estadoaccion` */

DROP TABLE IF EXISTS `estadoaccion`;

CREATE TABLE `estadoaccion` (
  `id_estadoaccion` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_estadoaccion`)

```

```

) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `estadocapacitacion` */

DROP TABLE IF EXISTS `estadocapacitacion`;

CREATE TABLE `estadocapacitacion` (
  `id_estadocapacitacion` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT 1,
  PRIMARY KEY (`id_estadocapacitacion`),
  KEY `id_estadocapacitacion` (`id_estadocapacitacion`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `estadoepp` */

DROP TABLE IF EXISTS `estadoepp`;

CREATE TABLE `estadoepp` (
  `id_estadoepp` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_estadoepp`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `gravedad` */

DROP TABLE IF EXISTS `gravedad`;

CREATE TABLE `gravedad` (
  `id_gravedad` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_gravedad`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `menu` */

DROP TABLE IF EXISTS `menu`;

CREATE TABLE `menu` (
  `id_menu` int(11) NOT NULL AUTO INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `orden` tinyint(4) DEFAULT NULL,
  `imagen` varchar(60) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT 1,
  PRIMARY KEY (`id_menu`)

```

```

) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `modalidad` */

DROP TABLE IF EXISTS `modalidad`;

CREATE TABLE `modalidad` (
  `id_modalidad` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_modalidad`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `opcion` */

DROP TABLE IF EXISTS `opcion`;

CREATE TABLE `opcion` (
  `id_opcion` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `url` varchar(100) DEFAULT NULL,
  `orden` tinyint(4) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT 1,
  `id_menu` int(11) NOT NULL,
  PRIMARY KEY (`id_opcion`),
  KEY `id_menu` (`id_menu`),
  CONSTRAINT `opcion_ibfk_1` FOREIGN KEY (`id_menu`)
REFERENCES `menu` (`id_menu`)
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `puesto` */

DROP TABLE IF EXISTS `puesto`;

CREATE TABLE `puesto` (
  `id_puesto` smallint(6) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_puesto`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `rol` */

DROP TABLE IF EXISTS `rol`;

CREATE TABLE `rol` (
  `id_rol` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) DEFAULT NULL,

```

```

`descripcion` varchar(100) DEFAULT NULL,
`activo` tinyint(4) DEFAULT NULL,
PRIMARY KEY (`id_rol`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `rolopcion` */

DROP TABLE IF EXISTS `rolopcion`;

CREATE TABLE `rolopcion` (
  `id_rolopcion` int(11) NOT NULL AUTO_INCREMENT,
  `id_rol` int(11) NOT NULL,
  `id_opcion` int(11) NOT NULL,
  `observacion` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_rolopcion`),
  KEY `id_rol` (`id_rol`),
  KEY `id_opcion` (`id_opcion`),
  CONSTRAINT `rolopcion_ibfk_1` FOREIGN KEY (`id_rol`)
REFERENCES `rol` (`id_rol`),
  CONSTRAINT `rolopcion_ibfk_2` FOREIGN KEY (`id_opcion`)
REFERENCES `opcion` (`id_opcion`)
) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `tipoaccidente` */

DROP TABLE IF EXISTS `tipoaccidente`;

CREATE TABLE `tipoaccidente` (
  `id_tipoaccidente` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_tipoaccidente`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `tipoaccion` */

DROP TABLE IF EXISTS `tipoaccion`;

CREATE TABLE `tipoaccion` (
  `id_tipoaccion` smallint(6) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT 1,
  PRIMARY KEY (`id_tipoaccion`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `tipocapacitacion` */

DROP TABLE IF EXISTS `tipocapacitacion`;

```

```

CREATE TABLE `tipocapacitacion` (
  `id_tipocapacitacion` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_tipocapacitacion`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `trabajador` */

DROP TABLE IF EXISTS `trabajador`;

CREATE TABLE `trabajador` (
  `id_trabajador` char(8) NOT NULL,
  `apellidopaterno` varchar(60) DEFAULT NULL,
  `apellidomaterno` varchar(60) DEFAULT NULL,
  `nombre` varchar(60) DEFAULT NULL,
  `fechanacimiento` date DEFAULT NULL,
  `fechaingreso` date DEFAULT NULL,
  `sexo` char(1) DEFAULT NULL,
  `capacitacionssoma` tinyint(4) DEFAULT NULL,
  `id_area` smallint(4) DEFAULT NULL,
  `id_puesto` smallint(4) DEFAULT NULL,
  `celular` varchar(20) DEFAULT NULL,
  `direccion` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_trabajador`),
  KEY `id_puesto` (`id_puesto`),
  KEY `id_area` (`id_area`),
  CONSTRAINT `trabajador_ibfk_1` FOREIGN KEY (`id_puesto`)
REFERENCES `puesto` (`id_puesto`),
  CONSTRAINT `trabajador_ibfk_2` FOREIGN KEY (`id_area`)
REFERENCES `area` (`id_area`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `trabajadorcapacitacion` */

DROP TABLE IF EXISTS `trabajadorcapacitacion`;

CREATE TABLE `trabajadorcapacitacion` (
  `id_trabajadorcapacitacion` int(11) NOT NULL
  AUTO_INCREMENT,
  `id_trabajador` char(8) NOT NULL,
  `id_capacitacion` int(11) NOT NULL,
  `notaobtenida` decimal(9,2) DEFAULT NULL,
  `aprobado` tinyint(4) DEFAULT NULL,
  `observaciones` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_trabajadorcapacitacion`),
  KEY `id_trabajador` (`id_trabajador`),
  KEY `id_capacitacion` (`id_capacitacion`),

```

```

    CONSTRAINT `trabajadorcapacitacion_ibfk_1` FOREIGN KEY
    (`id_trabajador`) REFERENCES `trabajador` (`id_trabajador`),
    CONSTRAINT `trabajadorcapacitacion_ibfk_2` FOREIGN KEY
    (`id_capacitacion`) REFERENCES `capacitacion`
    (`id_capacitacion`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `trabajadorepp` */

DROP TABLE IF EXISTS `trabajadorepp`;

CREATE TABLE `trabajadorepp` (
  `id_trabajadorepp` int(11) NOT NULL AUTO_INCREMENT,
  `id_epp` smallint(6) NOT NULL,
  `id_trabajador` char(8) NOT NULL,
  `fecha_entrega` date DEFAULT NULL,
  `fecha_caducidad` date DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  `entregadopor` varchar(100) DEFAULT NULL,
  `observaciones` varchar(100) DEFAULT NULL,
  `id_estadoepp` tinyint(4) NOT NULL,
  PRIMARY KEY (`id_trabajadorepp`),
  KEY `id_epp` (`id_epp`),
  KEY `id_trabajador` (`id_trabajador`),
  KEY `id_estadoepp` (`id_estadoepp`),
  CONSTRAINT `trabajadorepp_ibfk_1` FOREIGN KEY (`id_epp`)
REFERENCES `epp` (`id_epp`),
  CONSTRAINT `trabajadorepp_ibfk_2` FOREIGN KEY
(`id_trabajador`) REFERENCES `trabajador` (`id_trabajador`),
  CONSTRAINT `trabajadorepp_ibfk_3` FOREIGN KEY
(`id_estadoepp`) REFERENCES `estadoepp` (`id_estadoepp`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `turno` */

DROP TABLE IF EXISTS `turno`;

CREATE TABLE `turno` (
  `id_turno` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) DEFAULT NULL,
  `horainicio` time DEFAULT NULL,
  `horafin` time DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id_turno`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*Table structure for table `usuario` */

DROP TABLE IF EXISTS `usuario`;

```

```

CREATE TABLE `usuario` (
  `id_usuario` int(11) NOT NULL AUTO INCREMENT,
  `username` varchar(45) DEFAULT NULL,
  `password_hash` varchar(255) DEFAULT NULL,
  `nombrecompleto` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `activo` tinyint(4) DEFAULT NULL,
  `fecha_creacion` datetime DEFAULT current_timestamp(),
  `id_rol` int(11) NOT NULL,
  PRIMARY KEY (`id_usuario`),
  KEY `id_rol` (`id_rol`),
  CONSTRAINT `usuario_ibfk_1` FOREIGN KEY (`id_rol`)
REFERENCES `rol` (`id_rol`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_spanish_ci;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

FEPOL
Fond
Editorial

Anexo 2. Script módulo de seguridad

MySQL

```

insert into `menu` (`id_menu`, `descripcion`, `orden`,
`imagen`, `activo`) values (1,'Administración', 1, 'bi bi-
gear',1), (2,'Gestión de accidentes',2,' bi bi-exclamation-
triangle',1), (3, 'Capacitaciones', 3, 'bi bi-journal-
text',1), (4, 'Control de EPPs', 4, 'bi bi-shield-check',1), (5,
'Acciones correctivas', 5, 'bi bi-check2-circle',1), (6,
'Reporte', 6, 'bi bi-bar-chart',1);

/*Data for the table `opcion` */

insert into `opcion`(`id_opcion`, `descripcion`, `url`,
`orden`, `activo`, `id_menu`) values (2,' Gestión de usuarios',
'seg_usuario.php', 1, 1, 1), (3, 'Gestión de roles',
'seg_rol.php', 2, 1, 1), (4, 'Permisos', 'seg_permiso.php', 3,
1, 1), (5, 'Maestros', 'seg_maestro.php', 4, 1, 1), (6,
'Trabajadores', 'seg_trabajador.php', 5, 1, 1), (11,
'Registrar capacitación', 'cap_registro.php', 1, 1, 3), (12,
'Asignar trabajador', 'cap_trabajador.php', 2, 1, 3), (13,
'Historial', 'cap_historial.php', 3, 1, 3), (14, 'Registrar
epp', 'epp_registro.php', 1, 1, 4), (15, 'Asignar epp',
'epp_asigna.php', 2, 1, 4), (16, 'Registrar acción',
'cor_registro.php', 1, 1, 5), (17, 'Seguimiento',
'cor_seguimiento.php', 2, 1, 5), (18, 'Cerrar',
'cor_cerrar.php', 3, 1, 5), (19, 'Reportes generales',
'rep_general.php', 1, 1, 6), (20, 'Indicadores',
'rep_indicador.php', 2, 1, 6), (21, 'Bitácora',
'rep_bitacora.php', 3, 1, 6), (22, 'Registro',
'acc_registro.php', 1, 1, 2), (23, 'Investigación',
'acc_investiga.php', 2, 1, 2);

/*Data for the table `rol` */

insert into `rol`(`id_rol`, `nombre`, `descripcion`, `activo`)
values (1,'Administrador','Tiene acceso a todas las
opciones.',1), (2,'Reportes','Sólo tiene acceso a los reportes
del sistema',0), (3,'Supervisor','Tiene acceso a la gestión de
accidentes',1);

/*Data for the table `rolopcion` */

insert into `rolopcion` (`id_rolopcion`, `id_rol`,
`id_opcion`, `observacion`) values (1, 1, 2, ''), (2, 1, 3,
''), (3, 1, 4, ''), (4, 1, 5, ''), (5, 1, 6, ''), (6, 1, 11,
''), (7, 1, 12, ''), (8, 1, 13, ''), (9, 1, 14, ''),
(10,1,15,''), (11,1,16,''), (12,1,17,''), (13,1,18,''),
(14,1,19,''), (15,1,20,''), (16,1,21,''), (32,1,22,NULL),
(33,1,23,NULL);

```


Anexo 3. Código fuente módulo de seguridad

PHP. index.php

```
<?php
    if (!empty($_SERVER['HTTPS']) && ('on' ==
$_SERVER['HTTPS'])) {
        $uri = 'https://';
    } else {
        $uri = 'http://';
    }
    $uri .= $_SERVER['HTTP_HOST'];
    header('Location:
'. $uri . '/zet_ssoma/seguridad/login.php');
    exit;
?>
```

PHP. config.php

```
<?php
/*
CONFIGURACIÓN GENERAL DEL SISTEMA
Sistema SSOMA - Minera ZET
*/
error_reporting(E_ALL);
ini_set('display_errors',1);

/* -----
   RUTA BASE DEL PROYECTO
----- */
define("ROOT_PATH", dirname(__DIR__));
define('GL_SERVER', 'http://'. $_SERVER['HTTP_HOST']);
define('GL_RAIZ_HTTP', GL_SERVER . '/zet_ssoma/');
define('GL_RAIZ', 'c:/xampp/htdocs/zet_ssoma/');
define('GL_RAIZSSOMA', 'c:/xampp/htdocs/zet_ssoma/ssoma/');

/* -----
   CONFIGURACIÓN BASE DE DATOS
----- */
define("DB_HOST", "localhost");
define("DB_USER", "root");
define("DB_PASS", "");
define("DB_NAME", "ssoma_zet");

/* -----
   CONEXIÓN MYSQL
----- */
$conn = new mysqli(DB_HOST,DB_USER,DB_PASS,DB_NAME);

if($conn->connect_error){

    die("Error de conexión: ".$conn->connect_error);
```

```

}

/* -----
CONFIGURACIÓN DE SESIÓN
----- */

if(session_status() == PHP_SESSION_NONE){
    session_start();
}

/* -----
ZONA HORARIA
----- */
date_default_timezone_set("America/Lima");

?>

```

PHP. login.php

```

<?php
    session_start();
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Login - Sistema SSOMA</title>
    <!-- Bootstrap -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
    <!-- Icons -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons/font/bootstrap-icons.css" rel="stylesheet">
    <link rel="stylesheet" href="../css/login.css">
</head>
<body class="bg-light d-flex align-items-center justify-
content-center vh-100">
<div class="card shadow-lg p-4" style="width: 400px; border-
radius: 15px;">

    <div class="text-center mb-4">
        <i class="bi bi-shield-lock-fill text-primary"
style="font-size: 40px;"></i>
        <h4 class="mt-2">Sistema SSOMA</h4>
        <small class="text-muted">Minera ZET</small>
    </div>

    <?php if(isset($_GET['error'])) { ?>
        <div class="alert alert-danger">

```

```

        Usuario o contraseña incorrectos.
    </div>
<?php } ?>
<form method="POST" action="validar_login.php">

    <div class="mb-3">
        <label class="form-label">Usuario</label>
        <div class="input-group">
            <span class="input-group-text">
                <i class="bi bi-person"></i>
            </span>
            <input type="text" name="username"
class="form-control" required>
        </div>
    </div>
    <div class="mb-3">
        <label class="form-label">Contraseña</label>
        <div class="input-group">
            <span class="input-group-text">
                <i class="bi bi-lock"></i>
            </span>
            <input type="password" name="password"
class="form-control" required>
        </div>
    </div>
    <div class="d-grid">
        <button type="submit" class="btn btn-primary">
            <i class="bi bi-box-arrow-in-right"></i>
Ingresar
        </button>
    </div>
</form>
</div>
</body>
</html>

```

PHP. validar_login.php

```

<?php
session_start();
include("../conexion/conexion.php");
$username = trim($_POST['username']);
$password = trim($_POST['password']);
$sql = "SELECT * FROM usuario WHERE username = ? AND activo =
1";
$stmt = $conn->prepare($sql);
if(!$stmt){
    die("Error en prepare(): " . $conn->error);
}
$stmt->bind_param("s", $username);
$stmt->execute();

```

```

$resultado = $stmt->get_result();
if($resultado->num_rows == 1){
    $usuario = $resultado->fetch_assoc();
    if(password_verify($password,
    $usuario['password_hash'])){
        $_SESSION['id_usuario'] = $usuario['id_usuario'];
        $_SESSION['id_rol'] = $usuario['id_rol'];
        $_SESSION['username'] = $usuario['username'];
        $_SESSION['nombrecompleto'] =
    $usuario['nombrecompleto'];
        header("Location: ../dashboard.php");
        exit();
    } else {
        header("Location: login.php?error=1");
    }
} else {
    header("Location: login.php?error=1");
}
?>

```

PHP. logout.php

```

<?php
session_start();
session_destroy();
header("Location: login.php");
exit();
?>

```

PHP. header.php

```

<?php
if(session_status() === PHP_SESSION_NONE){
    session_start();
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Sistema SSOMA - Minera ZET</title>

    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <!-- Bootstrap -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/b
ootstrap.min.css" rel="stylesheet">

    <!-- Bootstrap Icons -->

```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons/font/bootstrap-icons.css" rel="stylesheet">

<style>
  body {
    background-color: #f4f6f9;
  }

  .sidebar {
    min-height: 100vh;
  }

  .sidebar .nav-link {
    transition: 0.3s;
  }

  .sidebar .nav-link:hover {
    background-color: rgba(255,255,255,0.15);
    border-radius: 6px;
  }

  .content-area {
    padding: 30px;
  }
</style>
</head>
<body>
<!-- TOPBAR -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark
shadow-sm px-4">
  <a class="navbar-brand d-flex align-items-center"
href="#">
    
    <strong>Sistema SSOMA</strong>
  </a>

  <div class="ms-auto text-white">
    <i class="bi bi-person-circle me-1"></i>
    <?php echo $_SESSION['username']; ?>
    |
    <a href="<?php echo
GL_RAIZ_HTTP?>seguridad/logout.php" class="text-warning text-
decoration-none">
      <i class="bi bi-box-arrow-right"></i> Salir
    </a>
  </div>
</nav>

<div class="container-fluid">
<div class="row">

```

PHP. sidebar.php

```

<?php
$id_rol = $_SESSION['id_rol'];
$sql = "SELECT
        o.id_menu,
        o.id_opcion,
        m.descripcion AS menu,
        o.descripcion AS opcion,
        m.imagen,
        CONCAT('ssoma/',o.url) as url
      FROM rolopcion r
      INNER JOIN opcion o ON r.id_opcion=o.id_opcion
      INNER JOIN menu m ON o.id_menu=m.id_menu
      WHERE r.id_rol=?
      AND o.activo=1
      AND m.activo=1
      ORDER BY m.orden, o.orden";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i",$id_rol);
$stmt->execute();
$result = $stmt->get_result();
?>
<div class="col-md-2 bg-primary sidebar p-3">
<ul class="nav flex-column" id="sidebarMenu">
<?php
$menu_actual = "";
while($fila = $result->fetch_assoc()){
    if($menu_actual != $fila['id_menu']){
        if($menu_actual != ""){
            echo "</ul></div></li>";
        }
        $menu_actual = $fila['id_menu'];
    }
    ?>
<li class="nav-item mb-2">
<a class="nav-link text-white" data-bs-toggle="collapse"
href="#menu<?php echo $fila['id_menu']; ?>">
<i class="bi <?php echo $fila['imagen']; ?> me-2"></i>
<?php echo $fila['menu']; ?>
</a>
<div class="collapse" id="menu<?php echo $fila['id_menu'];
?>" data-bs-parent="#sidebarMenu">
<ul class="nav flex-column ms-3">
<?php
}
?>
<li>
<a class="nav-link text-white" href="<?php echo GL_RAIZ_HTTP
. $fila['url']; ?>">
<?php echo $fila['opcion']; ?>
</a>
</li>
<?php

```

```

}
if($menu_actual != ""){
    echo "</ul></div></li>";
}
?>
</ul>

</div>

<div class="col-md-10 content-area">

```

PHP. footer.php

```

</div>
</div>
</div>

<footer class="bg-white text-center py-3 shadow-sm mt-4">
    <small class="text-muted">
        Sistema SSOMA - Minera ZET © 2026 | Desarrollo
        Académico
    </small>
</footer>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/boo
tstrap.bundle.min.js"></script>

</body>
</html>

```

PHP. dashboard.php

```

<?php
include("config.php");
include("layout/header.php");
include("layout/sidebar.php");
?>

<div class="card shadow-sm border-0">
    <div class="card-body p-5">

        <h2 class="mb-3 text-primary">
            Bienvenido al Sistema SSOMA
        </h2>

        <h4 class="mb-4 text-dark">
            MINERA ZET
        </h4>

        <p class="lead">

```

Desarrollo de Aplicaciones web para Ingeniería industrial

Este sistema ha sido desarrollado para gestionar de manera integral la Seguridad, Salud Ocupacional y Medio Ambiente (SSOMA), permitiendo el registro, seguimiento y análisis de accidentes laborales, capacitaciones, equipos de protección personal y acciones correctivas.

</p>

<hr>

<p>

A través de esta plataforma podrá:

</p>

Registrar y analizar accidentes

laborales

Gestionar capacitaciones del personal

Controlar la asignación y vencimiento de

EPP

Dar seguimiento a acciones correctivas

Generar reportes e indicadores

estratégicos

<div class="mt-4 text-muted">

Usuario conectado:

<?php echo \$_SESSION['username'];

?>

</div>

</div>

</div>

<?php include("layout/footer.php"); ?>

CSS. login.css

```
body {
    background: linear-gradient(135deg, #1f3c88, #2a5298);
}
.card {
    background-color: white;
}
```


Anexo 4. Script de carga de datos

MySQL

```

INSERT INTO categoria (descripcion, activo) VALUES
('Factores personales',1),
('Factores del trabajo',1),
('Gestión organizacional',1)

INSERT INTO tipoaccidente (descripcion, activo) VALUES
('Caída al mismo nivel',1),
('Caída de altura',1),
('Golpe contra objeto',1),
('Atrapamiento',1),
('Corte o laceración',1),
('Quemadura',1),
('Contacto eléctrico',1),
('Exposición a sustancias químicas',1);

INSERT INTO gravedad (descripcion, activo) VALUES
('Leve',1),
('Moderado',1),
('Grave',1),
('Fatal',1);

INSERT INTO AREA (descripcion, activo, id tipoarea) VALUES
('Producción',1,1),
('Mantenimiento',1,1),
('Almacén',1,1),
('Laboratorio',1,1),
('Administración',1,2),
('Logística',1,1);

INSERT INTO puesto (descripcion, activo) VALUES
('Operario',1),
('Supervisor',1),
('Técnico de mantenimiento',1),
('Ingeniero SSOMA',1),
('Almacenero',1),
('Analista administrativo',1);

INSERT INTO condicionpeligrosa (descripcion, activo) VALUES
('Piso resbaloso',1),
('Iluminación deficiente',1),
('Falta de señalización',1),
('Equipo defectuoso',1),
('Herramienta inadecuada',1),
('Desorden en el área de trabajo',1);

INSERT INTO actoinseguro (descripcion, activo) VALUES
('No usar EPP',1),
('Manipulación incorrecta de equipos',1),

```

```

('Desobedecer procedimientos',1),
('Uso de herramientas defectuosas',1),
('Trabajar sin autorización',1);

INSERT INTO causainmediata (descripcion, activo) VALUES
('Falta de capacitación',1),
('Uso incorrecto de herramientas',1),
('No uso de EPP',1),
('Condiciones inseguras del área',1),
('Falta de supervisión',1);

INSERT INTO causaraiz (descripcion, activo, id_categoria)
VALUES
('Falta de entrenamiento',1,1),
('Fatiga del trabajador',1,1),
('Procedimientos deficientes',1,2),
('Falta de mantenimiento',1,2),
('Falta de cultura de seguridad',1,3);

INSERT INTO turno (descripcion, horainicio, horafin, activo)
VALUES
('Turno mañana','06:00','14:00',1),
('Turno tarde','14:00','22:00',1),
('Turno noche','22:00','06:00',1);

INSERT INTO estado (descripcion, activo) VALUES
('Registrado',1),
('En investigación',1),
('En seguimiento',1),
('Cerrado',1);

INSERT INTO tipoaccion (descripcion, activo) VALUES
('Capacitación',1),
('Mejora de procedimiento',1),
('Mantenimiento de equipo',1),
('Cambio de EPP',1),
('Señalización de seguridad',1);

INSERT INTO estadoaccion (descripcion, activo) VALUES
('Pendiente',1),
('En proceso',1),
('Implementado',1),
('Verificado',1),
('Cerrado',1);

INSERT INTO curso (descripcion, activo) VALUES
('Seguridad industrial',1),
('Uso de EPP',1),
('Prevención de accidentes',1),
('Primeros auxilios',1),
('Trabajo en altura',1);

```

```

INSERT INTO modalidad (descripcion, activo) VALUES
('Presencial',1),
('Virtual',1),
('Semipresencial',1);

INSERT INTO tipocapacitacion (descripcion, activo) VALUES
('Inducción',1),
('Capacitación periódica',1),
('Capacitación especializada',1);

INSERT INTO estadoepp (descripcion, activo) VALUES
('Entregado',1),
('En uso',1),
('Vencido',1),
('Reemplazado',1);

INSERT INTO epp (descripcion, vidautilmeses, obligatorio,
activo) VALUES
('Casco de seguridad',24,'SI',1),
('Guantes de seguridad',6,'SI',1),
('Lentes de protección',12,'SI',1),
('Botas de seguridad',12,'SI',1),
('Protector auditivo',12,'SI',1),
('Mascarilla respiratoria',6,'SI',1);

INSERT INTO trabajador
(id_trabajador,apellidopaterno,apellidomaterno,nombre,fechana
cimiento,fechaingreso,sexo,capacitacionssoma,id_area,id_puest
o,activo)
VALUES
('70000001','Perez','Gomez','Juan','1990-03-10','2021-01-
10','M',1,1,1,1),
('70000002','Garcia','Lopez','Luis','1988-07-12','2020-05-
15','M',1,1,1,1),
('70000003','Rodriguez','Perez','Carlos','1992-04-18','2022-
03-01','M',1,2,2,1),
('70000004','Sanchez','Torres','Miguel','1987-09-20','2019-
06-10','M',1,3,1,1),
('70000005','Ramirez','Diaz','Pedro','1991-12-02','2021-08-
22','M',1,1,1,1),
('70000006','Torres','Rojas','Jose','1993-05-11','2023-02-
01','M',1,2,3,1),
('70000007','Flores','Vargas','Marco','1990-06-30','2021-04-
12','M',1,1,1,1),
('70000008','Castro','Mendoza','Diego','1989-11-01','2020-09-
05','M',1,4,4,1),
('70000009','Vargas','Ruiz','Luis','1994-08-15','2022-01-
10','M',1,3,2,1),
('70000010','Morales','Silva','Jorge','1988-01-25','2019-03-
14','M',1,2,1,1);

```

```

INSERT INTO accidente
(descripcion,fechaaccidente,horaaccidente,id_tipoaccidente,id
_turno,id_area,id_supervisor,
id_causaraiz,id_condicionpeligrosa,id_causainmediata,id_actoi
nseguro,fecha_registro,usuario_registro)
VALUES
('Caída por piso mojado','2025-01-
10','09:30',1,1,1,'70000001',1,1,3,1,NOW(),'juliozet'),
('Golpe con herramienta','2025-01-
12','10:45',3,1,2,'70000002',2,3,2,2,NOW(),'juliozet'),
('Corte con metal','2025-01-
15','14:20',5,2,1,'70000003',1,4,2,3,NOW(),'juliozet'),
('Quemadura leve','2025-01-
20','16:30',6,2,3,'70000004',3,2,3,1,NOW(),'juliozet'),
('Caída de escalera','2025-01-
25','08:10',2,1,1,'70000005',2,1,1,2,NOW(),'juliozet');

INSERT INTO accidente_detalle
(id_accidente,id_trabajador,diaperdido,costoestimado,id_grave
dad)
VALUES
(1,'70000005',2,150.00,1),
(2,'70000004',3,300.00,2),
(3,'70000002',1,120.00,1),
(4,'70000006',4,400.00,2),
(5,'70000007',10,900.00,3);

INSERT INTO trabajadorepp
(id_epp,id_trabajador,fecha_entrega,fecha_caducidad,activo,en
tregadopor,observaciones,id_estadoepp)
VALUES
(1,'70000001','2025-01-01','2027-01-
01',1,'Supervisor','Entrega inicial',1),
(2,'70000002','2025-01-01','2026-01-
01',1,'Supervisor','Entrega inicial',1),
(3,'70000003','2025-01-01','2026-01-
01',1,'Supervisor','Entrega inicial',1),
(4,'70000004','2025-01-01','2026-01-
01',1,'Supervisor','Entrega inicial',1),
(5,'70000005','2025-01-01','2026-01-
01',1,'Supervisor','Entrega inicial',1);

INSERT INTO estadocapacitacion (descripcion, activo) VALUES
('Programada',1),
('Convocada',1),
('En ejecución',1),
('Finalizada',1),
('Aprobada',1),
('Desaprobada',1),
('Cancelada',1),
('Reprogramada',1);

```

```

INSERT INTO capacitacion
(fecha_inicio,fecha_fin,nombre,temaprincipal,id_curso,total_h
oras,id_tipocapitacion,id_modalidad,id_estado,certificacion
,instructor)
VALUES
('2025-02-01','2025-02-01','Seguridad Industrial','Prevención
de accidentes',1,8,1,1,1,'SI','Ing. Carlos Diaz'),
('2025-03-01','2025-03-01','Uso de EPP','Equipos de
protección',2,6,1,1,1,'SI','Ing. Luis Torres'),
('2025-04-01','2025-04-01','Trabajo en altura','Riesgos en
altura',5,8,2,1,1,'SI','Ing. Marco Ruiz');

INSERT INTO trabajadorcapitacion
(id_trabajador,id_capitacion,notaobtenida,aprobado,observac
iones)
VALUES
('70000001',1,18,1,'Aprobado'),
('70000002',1,16,1,'Aprobado'),
('70000003',2,17,1,'Aprobado'),
('70000004',2,15,1,'Aprobado'),
('70000005',3,19,1,'Excelente');

INSERT INTO accioncorrectiva
(descripcion,fecha_registro,fecha_compromiso,fecha_cierre,obs
ervaciones,id_estadoaccion,id_tipoaccion,id_responsable,activ
o)
VALUES
('Señalizar zona resbalosa',NOW(),'2025-02-
01',NULL,'Pendiente implementación',1,5,'70000002',1),
('Capacitación en uso de herramientas',NOW(),'2025-02-
10',NULL,'Programada',1,1,'70000003',1),
('Reemplazo de escalera defectuosa',NOW(),'2025-02-
05',NULL,'Compra en proceso',2,3,'70000004',1);

```

Editorial

Anexo 5. Código fuente formulario gestión de roles

PHP. seg_rol.php

```
<?php
require_once("../config.php");
include("../layout/header.php");
include("../layout/sidebar.php");

$sql = "SELECT * FROM rol ORDER BY id_rol";
$result = $conn->query($sql);
?>

<h3 class="mb-4">Gestión de Roles</h3>
<div class="card shadow-sm">
<div class="card-header bg-primary text-white">
    <a href="seg_rol_form.php" class="btn btn-light btn-sm">
        <i class="bi bi-plus-circle"></i> Nuevo Rol
    </a>
</div>
<div class="card-body">
<table class="table table-bordered table-hover">
    <thead class="table-light">
        <tr>
            <th width="5%">ID</th>
            <th>Nombre</th>
            <th width="45%">Descripción</th>
            <th width="10%">Estado</th>
            <th width="10%">Acciones</th>
        </tr>
    </thead>
<tbody>
<?php while($fila = $result->fetch_assoc()) { ?>
<tr>
    <td><?php echo $fila['id_rol']; ?></td>
    <td><?php echo $fila['nombre']; ?></td>
    <td><?php echo $fila['descripcion']; ?></td>
    <td align="center">
<?php
if($fila['activo']==1){
echo "<span class='badge bg-success'>Activo</span>";
}else{
echo "<span class='badge bg-danger'>Inactivo</span>";
}
?>
</td>
<td align="center">
<a href="seg_rol_form.php?id=<?php echo $fila['id_rol']; ?>"
class="btn btn-warning btn-sm">
<i class="bi bi-pencil"></i>
</a>
<a href="#" onclick="confirmarEliminar(<?php echo
$fila['id_rol']; ?>)"
```

```

class="btn btn-danger btn-sm">
<i class="bi bi-trash"></i>
</a>
</td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
<script>
function confirmarEliminar(id){
if(confirm("¿Está seguro de eliminar este registro?")){
window.location = "seg_rol_eliminar.php?id=" + id;
}
}
</script>
<?php include("../layout/footer.php"); ?>

```

PHP. seg_rol_form.php

```

<?php
error_reporting(E_ALL);
ini_set('display_errors',1);
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
include_once(GL_RAIZ."layout/header.php");
include_once(GL_RAIZ."layout/sidebar.php");
$id="";
$nombre="";
$descripcion="";
$activo=1; // valor por defecto
if(isset($_GET['id'])){
$id=$_GET['id'];
$sql="SELECT * FROM rol WHERE id_rol=?";
$stmt=$conn->prepare($sql);
$stmt->bind_param("i",$id);
$stmt->execute();
$result=$stmt->get_result();
$fila=$result->fetch_assoc();
$nombre=$fila['nombre'];
$descripcion=$fila['descripcion'];
$activo=$fila['activo']; // NUEVO CAMPO
}
?>
<h3 class="mb-4">Formulario Rol</h3>

<div class="card shadow-sm">

<div class="card-body">

```

```

<form method="POST" action="seg_rol_guardar.php">

<input type="hidden" name="id_rol" value="<?php echo $id ?>">

<div class="mb-3">
<label class="form-label">Nombre</label>

<input type="text" name="nombre"
class="form-control"
value="<?php echo $nombre ?>"
required>
</div>

<div class="mb-3">
<label class="form-label">Descripción del Rol</label>

<input type="text" name="descripcion"
class="form-control"
value="<?php echo $descripcion ?>"
required>
</div>
<!-- NUEVO CHECKBOX -->
<div class="form-check mb-3">
<input class="form-check-input"
type="checkbox"
name="activo"
value="1"
<?php if($activo==1) echo "checked"; ?>>
<label class="form-check-label">
Activo
</label>
</div>
<button class="btn btn-primary">
Guardar
</button>
<a href="seg_rol.php" class="btn btn-secondary">
Cancelar
</a>
</form>
</div>
</div>
<?php include("../layout/footer.php"); ?>

```

PHP. seg_rol_guardar.php

```

<?php
error_reporting(E_ALL);
ini_set('display_errors',1);

include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");

```



```

/* Recibir datos del formulario */
$id = isset($_POST['id_rol']) ? $_POST['id_rol'] : "";
$nombre = isset($_POST['nombre']) ? $_POST['nombre'] : "";
$descripcion = isset($_POST['descripcion']) ?
$_POST['descripcion'] : "";
$activo = isset($_POST['activo']) ? 1 : 0; // checkbox
/* SI ES NUEVO REGISTRO */
if($id == ""){
$sql = "INSERT INTO rol (nombre, descripcion, activo)
VALUES (?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ssi", $nombre, $descripcion, $activo);
$stmt->execute();
}
/* SI ES EDICIÓN */
else{
$sql = "UPDATE rol
SET nombre=?, descripcion=?, activo=?
WHERE id_rol=?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ssii", $nombre, $descripcion, $activo, $id);
$stmt->execute();
}
/* Redirigir al listado */
header("Location: seg_rol.php");
exit();
?>

```

PHP. seg_rol_eliminar.php

```

<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
if(isset($_GET['id'])){
$id = $_GET['id'];
$conn->begin_transaction();
try {
/* Eliminar detalle del accidente */
$sql="DELETE FROM accidente_detalle WHERE
id_accidente=?";
$stmt=$conn->prepare($sql);
$stmt->bind_param("i", $id);
$stmt->execute();
/* Eliminar accidente */
$sql="DELETE FROM accidente WHERE id_accidente=?";
$stmt=$conn->prepare($sql);
$stmt->bind_param("i", $id);
$stmt->execute();
$conn->commit();
}

```

```
}  
catch(Exception $e) {  
    $conn->rollback();  
}  
}  
header("Location: acc_registro.php");  
exit();  
?>
```

FEPOL
Fond
Editorial

Anexo 6. Código fuente formulario gestión de accidentes

PHP. acc_registro.php

```
<?php
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
include_once(GL_RAIZ."layout/header.php");
include_once(GL_RAIZ."layout/sidebar.php");
$anio = isset($_GET['anio']) ? $_GET['anio'] : date("Y");
?>
<!-- Bootstrap Icons -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.css">
<h3 class="mb-4">Gestión de Accidentes</h3>
<div class="card shadow-sm">
<div class="card-body">
<form method="GET" class="row mb-3">
<div class="col-md-3">
<label class="form-label">Filtrar por Año</label>
<select name="anio" class="form-select"
onchange="this.form.submit()">
<?php
$sql="SELECT DISTINCT YEAR(fechaaccidente) anio FROM
accidente ORDER BY anio DESC";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
    $selected=($anio==$fila['anio'])?"selected":"";
    echo "<option value='".$fila['anio']."'
    $selected>".$fila['anio']."</option>";
}
?>
</select>
</div>
<div class="col-md-9 text-end align-self-end">
<a href="acc_editar.php" class="btn btn-primary">
<i class="bi bi-plus-circle"></i> Nuevo Accidente
</a>
</div>
</form>
<table class="table table-bordered table-striped">
<thead>
<tr>
<th width="5%">Id</th>
<th width="10%">Fecha</th>
<th>Descripción</th>
<th width="15%">Estado</th>
<th width="15%">Acciones</th>
</tr>
</thead>
```

```

<tbody>
<?php
$sql="SELECT
ac.id_accidente,
date_format(ac.fechaaccidente,'%d-%m-%Y') as fechaaccidente,
ac.descripcion,
e.descripcion as estado
FROM accidente ac
INNER JOIN estado e ON ac.id_estado=e.id_estado
WHERE YEAR(ac.fechaaccidente)=?
ORDER BY ac.fechaaccidente DESC";
$stmt=$conn->prepare($sql);
$stmt->bind_param("i",$anio);
$stmt->execute();
$result=$stmt->get_result();
while($fila=$result->fetch_assoc()){
?>
<tr>
<td align="center"><?php echo $fila['id_accidente'] ?></td>
<td><?php echo $fila['fechaaccidente'] ?></td>
<td><?php echo $fila['descripcion'] ?></td>
<td><?php echo $fila['estado'] ?></td>
<td align="center">
<!-- EDITAR -->
<a href="acc_editar.php?id=<?php echo $fila['id_accidente']
?>"
class="btn btn-warning btn-sm"
title="Editar accidente">
<i class="bi bi-pencil-square"></i>
</a>
<!-- TRABAJADORES INVOLUCRADOS -->
<a href="acc_trabajadores.php?id=<?php echo
$fila['id_accidente'] ?>"
class="btn btn-info btn-sm"
title="Trabajadores involucrados">
<i class="bi bi-person-lines-fill"></i>
</a>
<!-- ELIMINAR -->
<a href="#"
onclick="eliminarAccidente(<?php echo $fila['id_accidente']
?>)"
class="btn btn-danger btn-sm"
title="Eliminar accidente">
<i class="bi bi-trash"></i>
</a>
</td>
</tr>
<?php
}
?>
</tbody>
</table>

```

```

</div>
</div>
<script>
function eliminarAccidente(id){
if(confirm("¿Desea eliminar este accidente?")){
window.location="acc_eliminar.php?id="+id;
}
}
</script>
<?php include("../layout/footer.php"); ?>

```

PHP. acc_editar.php

```

<?php
error_reporting(E_ALL);
ini_set('display_errors',1);
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
include_once(GL_RAIZ."layout/header.php");
include_once(GL_RAIZ."layout/sidebar.php");
$id="";
$descripcion="";
$fecha=date("Y-m-d");
$hora=date("H:i");
$id_area="";
$id_tipoaccidente="";
$id_turno="";
$id_estado="";
$id_causaraiz="";
$id_condicionpeligrosa="";
$id_causainmediata="";
$id_actoinseguro="";
$id_supervisor="";
if(isset($_GET['id'])){
$id=$_GET['id'];
$sql="SELECT * FROM accidente WHERE id_accidente=?";
$stmt=$conn->prepare($sql);
$stmt->bind_param("i",$id);
$stmt->execute();
$result=$stmt->get_result();
$fila=$result->fetch_assoc();
$descripcion=$fila['descripcion'];
$fecha=$fila['fechaaccidente'];
$hora=$fila['horaaccidente'];
$id_area=$fila['id_area'];
$id_tipoaccidente=$fila['id_tipoaccidente'];
$id_turno=$fila['id_turno'];
$id_estado=$fila['id_estado'];
$id_causaraiz=$fila['id_causaraiz'];
$id_condicionpeligrosa=$fila['id_condicionpeligrosa'];
$id_causainmediata=$fila['id_causainmediata'];

```

```

$id_actoinseguro=$fila['id_actoinseguro'];
$id_supervisor=$fila['id_supervisor'];
}
?>
<h3 class="mb-4">Registro de Accidente</h3>
<div class="card shadow-sm">
<div class="card-body">
<form method="POST" action="acc_grabar.php">
<input type="hidden" name="id_accidente" value="<?php echo
$id ?>">
<div class="row">
<div class="col-md-3">
<label>Fecha</label>
<input type="date" name="fecha" class="form-control"
value="<?php echo $fecha ?>" required>
</div>
<div class="col-md-3">
<label>Hora</label>
<input type="time" name="hora" class="form-control"
value="<?php echo $hora ?>" required>
</div>
<div class="col-md-6">
<label>Área</label>
<select name="id_area" class="form-select">
<?php
$sql="SELECT * FROM area WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
    $selected=($id_area==$fila['id_area'])?"selected":"";
    echo "<option value='".$fila['id_area']."'
    $selected>".$fila['descripcion']."</option>";
}
?>
</select>
</div>
</div>
<br>
<div class="row">
<div class="col-md-4">
<label>Tipo de Accidente</label>
<select name="id_tipoaccidente" class="form-select">
<?php
$sql="SELECT * FROM tipoaccidente WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
    $selected=($id_tipoaccidente==$fila['id_tipoaccidente'])?"sel
ected":"";
    echo "<option value='".$fila['id_tipoaccidente']."'
    $selected>".$fila['descripcion']."</option>";
}
?>
</select>

```

```

</div>
<div class="col-md-4">
<label>Turno</label>
<select name="id_turno" class="form-select">
<?php
$sql="SELECT * FROM turno WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
$selected=($id_turno==$fila['id_turno'])?"selected":"";
echo "<option value='". $fila['id_turno']. "'
$selected>". $fila['descripcion']. "</option>";
}
?>
</select>
</div>
<div class="col-md-4">
<label>Estado</label>
<select name="id_estado" class="form-select">
<?php
$sql="SELECT * FROM estado WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
$selected=($id_estado==$fila['id_estado'])?"selected":"";
echo "<option value='". $fila['id_estado']. "'
$selected>". $fila['descripcion']. "</option>";
}
?>
</select>
</div>
</div>
<br>
<div class="mb-3">
<label>Descripción del accidente</label>
<textarea name="descripcion"
class="form-control"
rows="3"><?php echo $descripcion ?></textarea>
</div>
<hr>
<h5>Causas del accidente</h5>
<div class="row">
<div class="col-md-6">
<label>Causa raíz</label>
<select name="id_causaraiz" class="form-select">
<?php
$sql="SELECT * FROM causaraiz WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
$selected=($id_causaraiz==$fila['id_causaraiz'])?"selected":"";
echo "<option value='". $fila['id_causaraiz']. "'
$selected>". $fila['descripcion']. "</option>";
}

```

```

?>
</select>
</div>
<div class="col-md-6">
<label>Condición peligrosa</label>
<select name="id_condicionpeligrosa" class="form-select">
<?php
$sql="SELECT * FROM condicionpeligrosa WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
    $selected=($id_condicionpeligrosa==$fila['id_condicionpeligrosa'])?"selected":"";
    echo "<option value='".$fila['id_condicionpeligrosa']."'
    $selected>".$fila['descripcion']."</option>";
}
?>
</select>
</div>
</div>
<br>
<div class="row">
<div class="col-md-6">
<label>Causa inmediata</label>
<select name="id_causainmediata" class="form-select">
<?php
$sql="SELECT * FROM causainmediata WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
    $selected=($id_causainmediata==$fila['id_causainmediata'])?"selected":"";
    echo "<option value='".$fila['id_causainmediata']."'
    $selected>".$fila['descripcion']."</option>";
}
?>
</select>
</div>
<div class="col-md-6">
<label>Acto inseguro</label>
<select name="id_actoinseguro" class="form-select">
<?php
$sql="SELECT * FROM actoinseguro WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
    $selected=($id_actoinseguro==$fila['id_actoinseguro'])?"selected":"";
    echo "<option value='".$fila['id_actoinseguro']."'
    $selected>".$fila['descripcion']."</option>";
}
?>
</select>
</div>
</div>

```



```

<br>
<div class="mb-3">
<label>Trabajador que informó el accidente</label>
<select name="id_supervisor" class="form-select">
<?php
$sql="SELECT id_trabajador,nombre,apellidopaterno
FROM trabajador WHERE activo=1";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
$selected=($id_supervisor==$fila['id_trabajador'])?"selected"
:"";
echo "<option value='".$fila['id_trabajador']."' $selected>"
.$fila['apellidopaterno']."' ".$fila['nombre']."'</option>";
}
?>
</select>
</div>
<br>
<button class="btn btn-primary">
Guardar
</button>
<a href="acc_registro.php"
class="btn btn-secondary">
Cancelar
</a>
</form>
</div>
</div>
<?php include("../layout/footer.php"); ?>

```

PHP. acc_grabar.php

```

<?php
error_reporting(E_ALL);
ini_set('display_errors',1);
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
$id=$ _POST['id accidente'];
$fecha=$ _POST['fecha'];
$hora=$ _POST['hora'];
$descripcion=$ _POST['descripcion'];
$id_area=$ _POST['id_area'];
$id_tipoaccidente=$ _POST['id tipoaccidente'];
$id turno=$ _POST['id turno'];
$id_estado=$ _POST['id estado'];
$id_causaraiz=$ _POST['id_causaraiz'];
$id_condicionpeligrosa=$ _POST['id_condicionpeligrosa'];
$id_causainmediata=$ _POST['id_causainmediata'];
$id_actoinseguro=$ _POST['id_actoinseguro'];
$id_supervisor=$ _POST['id_supervisor'];

```

```

$usuario=$ _SESSION['username'];
if($id==""){
$sql="INSERT INTO accidente
(descripcion,fechaaccidente,horaaccidente,
id_tipoaccidente,id_turno,id_area,id_supervisor,
id_causaraiz,id_condicionpeligrosa,
id_causainmediata,id_actoinseguro,
usuario_registro,id_estado)
VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?)";
$stmt=$conn->prepare($sql);
$stmt->bind_param(
"sssiissiiissi",
$descripcion,
$fecha,
$hora,
$id_tipoaccidente,
$id_turno,
$id_area,
$id_supervisor,
$id_causaraiz,
$id_condicionpeligrosa,
$id_causainmediata,
$id_actoinseguro,
$usuario,
$id_estado
);
$stmt->execute();
}
else{
$sql="UPDATE accidente SET
descripcion=?,
fechaaccidente=?,
horaaccidente=?,
id_tipoaccidente=?,
id_turno=?,
id_area=?,
id_supervisor=?,
id_causaraiz=?,
id_condicionpeligrosa=?,
id_causainmediata=?,
id_actoinseguro=?,
id_estado=?
WHERE id accidente=?";
$stmt=$conn->prepare($sql);
$stmt->bind_param(
"sssiissiiissi",
$descripcion,
$fecha,
$hora,
$id_tipoaccidente,
$id_turno,
$id_area,

```

```

$id_supervisor,
$id_causaraiz,
$id_condicionpeligrosa,
$id_causainmediata,
$id_actoinseguro,
$id_estado,
$id
);
$stmt->execute();
}
header("Location: acc_registro.php");
exit();
?>

```

PHP. acc_eliminar.php

```

<?php
error_reporting(E_ALL);
ini_set('display_errors',1);
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
if(isset($_GET['id'])) {
$id = $_GET['id'];
$conn->begin_transaction();
try {
    /* Eliminar detalle del accidente */
    $sql="DELETE FROM accidente_detalle WHERE
id_accidente=?";
    $stmt=$conn->prepare($sql);
    $stmt->bind_param("i",$id);
    $stmt->execute();
    /* Eliminar accidente */
    $sql="DELETE FROM accidente WHERE id_accidente=?";
    $stmt=$conn->prepare($sql);
    $stmt->bind_param("i",$id);
    $stmt->execute();
    $conn->commit();
}
catch(Exception $e) {
    $conn->rollback();
}
}
header("Location: acc_registro.php");
exit();
?>

```

PHP. acc_trabajadores.php

```

<?php
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
include_once(GL_RAIZ."layout/header.php");
include_once(GL_RAIZ."layout/sidebar.php");
$id_accidente=$_GET['id'];
/* DATOS DEL ACCIDENTE */
$sql="SELECT
date_format(fechaaccidente,'%d-%m-%Y') fecha,
descripcion
FROM accidente
WHERE id_accidente=?";
$stmt=$conn->prepare($sql);
$stmt->bind_param("i",$id_accidente);
$stmt->execute();
$result=$stmt->get_result();
$accidente=$result->fetch_assoc();
?>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.css">
<link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
rc.0/dist/css/select2.min.css" rel="stylesheet"/>
<h3 class="mb-4">Trabajadores involucrados en accidente</h3>
<div class="card shadow-sm">
<div class="card-body">
<div class="alert alert-secondary">
<b>Accidente:</b> <?php echo $accidente['descripcion'] ?><br>
<b>Fecha:</b> <?php echo $accidente['fecha'] ?>
</div>
<form method="POST" action="acc_trabajadores_guardar.php">
<input type="hidden" name="id_accidente" value="<?php echo
$id_accidente ?>">
<div class="row">
<div class="col-md-5">
<label>Trabajador</label>
<select name="id_trabajador" class="form-select trabajador">
<option value="">Selecione trabajador</option>
<?php
$sql="SELECT id_trabajador,apellidopaterno,nombre
FROM trabajador
WHERE activo=1
ORDER BY apellidopaterno";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
echo "<option value='".$fila['id_trabajador']."'>"
.$fila['apellidopaterno']."' ".$fila['nombre']."'</option>";
}
?>
</select>
</div>

```

```

<div class="col-md-2">
<label>Días perdidos</label>
<input type="number" name="diasperdido"
class="form-control">
</div>
<div class="col-md-2">
<label>Costo</label>
<input type="number" step="0.01"
name="costoestimado"
class="form-control">
</div>
<div class="col-md-2">
<label>Gravedad</label>
<select name="id_gravedad" class="form-select">
<?php
$sql="SELECT * FROM gravedad";
$result=$conn->query($sql);
while($fila=$result->fetch_assoc()){
echo "<option value='".$fila['id_gravedad']."'>"
.fila['descripcion']."'>";
}
?>
</select>
</div>
<div class="col-md-1 d-flex align-items-end">
<button class="btn btn-success">
<i class="bi bi-plus-circle"></i>
</button>
</div>
</div>
</form>
<hr>
<h5>Trabajadores registrados</h5>
<table class="table table-bordered">
<thead>
<tr>
<th>Trabajador</th>
<th>Días perdidos</th>
<th>Costo</th>
<th>Gravedad</th>
<th width="10%">Acción</th>
</tr>
</thead>
<tbody>
<?php
$sql="SELECT
d.id_accidente_detalle,
t.apellidopaterno,
t.nombre,
d.diaperdido,
d.costoestimado,
g.descripcion gravedad

```

```

FROM accidente_detalle d
INNER JOIN trabajador t ON d.id_trabajador=t.id_trabajador
INNER JOIN gravedad g ON d.id_gravedad=g.id_gravedad
WHERE d.id_accidente=?";
#echo $sql;
$stmt=$conn->prepare($sql);
$stmt->bind_param("i",$id_accidente);
$stmt->execute();
$result=$stmt->get_result();
while($fila=$result->fetch_assoc()){
?>
<tr>
<td><?php echo $fila['apellidopaterno']." ".$fila['nombre']
?></td>
<td><?php echo $fila['diaperdido'] ?></td>
<td><?php echo $fila['costoestimado'] ?></td>
<td><?php echo $fila['gravedad'] ?></td>
<td align="center">
<a href="#"
onclick="eliminarDetalle(<?php echo
$fila['id_accidente_detalle'] ?>)"
class="btn btn-danger btn-sm">
<i class="bi bi-trash"></i>
</a>
</td>
</tr>
<?php
}
?>
</tbody>
</table>
<a href="acc_registro.php"
class="btn btn-secondary">
<i class="bi bi-arrow-left"></i> Volver
</a>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.mi
n.js"></script>
<script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
rc.0/dist/js/select2.min.js"></script>
<script>
$('.trabajador').select2();
function eliminarDetalle(id){
if(confirm("¿Eliminar trabajador del accidente?")){
window.location="acc_trabajadores_eliminar.php?id="+id;
}
}
</script>
<?php include("../layout/footer.php"); ?>

```

PHP. acc_trabajadores_guardar.php

```
<?php
include("../config.php");
include_once(GL_RAIZ."conexion/conexion.php");
$id_accidente=$_POST['id_accidente'];
$id_trabajador=$_POST['id_trabajador'];
$dias=$_POST['diasperdido'];
$costo=$_POST['costoestimado'];
$gravedad=$_POST['id_gravedad'];
$sql="INSERT INTO accidente_detalle
(id_accidente,id_trabajador,diasperdido, costoestimado,id_gravedad)
VALUES (?, ?, ?, ?, ?)";
$stmt=$conn->prepare($sql);
$stmt->bind_param(
"isiidi",
$id_accidente,
$id_trabajador,
$dias,
$costo,
$gravedad
);
$stmt->execute();
header("Location: acc_trabajadores.php?id=".$id_accidente);
```

FEPOL
Fond
Editorial

Referencias bibliográficas

- Alvarez, J. (2026). Bases de datos para la Ingeniería Industrial (1ª ed.). Professionals On line S.A.C.
- Asfahl, C. R., & Rieske, D. W. (2018). Seguridad industrial y administración de la salud (6.ª ed.). Pearson Educación.
- Chen, P. P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36. <https://doi.org/10.1145/320434.320440>
- Date, C. J. (2004). Introducción a los sistemas de bases de datos (7.ª ed.). Pearson Educación.
- Heurtel, O. (2019). Aprender a desarrollar un sitio web con PHP y MySQL: Ejercicios prácticos y corregidos (3.ª ed.). Ediciones ENI.
- Instituto Nacional de Seguridad y Salud en el Trabajo. (2023). NTP 1.001: Indicadores para la gestión de la seguridad y salud en el trabajo. INSST.
- Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative Industrie 4.0. Acatech – National Academy of Science and Engineering.
- Laudon, K. C., & Laudon, J. P. (2022). Management information systems: Managing the digital firm (17th ed.). Pearson.
- Meyers, F. E., & Stephens, M. P. (2019). Diseño de instalaciones productivas (5.ª ed.). Pearson Educación.
- Niebel, B. W., & Freivalds, A. (2019). Ingeniería industrial: Métodos, estándares y diseño del trabajo (13.ª ed.). McGraw-Hill.
- Nixon, R. (2018). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (5th ed.). O'Reilly Media.

- O'Brien, J. A., & Marakas, G. M. (2019). Management information systems (11th ed.). McGraw-Hill Education.
- Schwab, K. (2017). The fourth industrial revolution. Crown Business.
- Sharda, R., Delen, D., & Turban, E. (2018). Business intelligence, analytics, and data science: A managerial perspective (4th ed.). Pearson.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Fundamentos de bases de datos (7.^a ed.). McGraw-Hill Education.
- Welling, L., & Thomson, L. (2017). PHP and MySQL web development (5th ed.). Addison-Wesley Professional.

