

SISTEMA DE CONTROL DE ACTIVOS INTAGIBLES

PROYECTO TEÓRICO-PRÁCTICO

SEGUNDA ENTREGA

Este documento corresponde a la **segunda entrega del proyecto**, desarrollada en entorno **Web (JSP, Servlets, HTML, CSS, JS)** con conexión a **PostgreSQL**. El sistema permite el registro, consulta y control de licencias de software, incluyendo el cálculo automático de amortización mensual, acumulada y valor en libros, así como la generación de reportes contables en formato PDF.

UNIVERSIDAD DE EL SALVADOR
FACTULTAD DE INGENIERIA Y ARQUITECTURA
ESCUELA DE SISTEMAS INFORMATICOS
PROGRAMACION III PRN315
CICLO II-2025



SISTEMA DE CONTROL DE ACTIVOS INTAGIBLES

GRUPO No.16



INTEGRANTES:

Esquivel Rivas, René Alejandro ER23012

Martínez Velásquez, Dayana Michelle MV23061

Morales Herrera, Carlos Manuel MH22066

Pérez León, José Manuel PL22022

Docente: Ing. Arnoldo Inocencio Rivas Molina

Fecha de entrega: sábado 22 de noviembre de 2025

NOTA:



INDICE

INTRODUCCIÓN	4
JUSTIFICACIÓN	5
MARCO TEORICO.....	6
1. Activos intangibles y licencias de software	6
2. Amortización y valor en libros	6
3. Programación orientada a objetos (POO)	7
4. Arquitectura multicapa	7
5. Bases de datos relacionales y PostgreSQL	7
6. Tecnologías utilizadas	8
7. Buenas prácticas de programación	8
8. Relevancia académica y profesional	8
METODOLOGÍA DEL DESARROLLO	9
Etapas del desarrollo	9
REQUERIMIENTOS DEL SISTEMA.....	10
Requerimientos funcionales	10
Requerimientos no funcionales	11
Casos de uso principales	11
DISEÑO DEL SISTEMA.....	12
MOCKUPS DE INTERFAZ	21
DESARROLLO E IMPLEMENTACIÓN	25
PRUEBAS Y VALIDACIONES.....	27
RESULTADOS Y REPORTES DEL SISTEMA	35
CONCLUSIONES.....	37
RECOMENDACIONES.....	38
BIBLIOGRAFIA.....	39
ANEXOS	40

INTRODUCCIÓN

El control de activos intangibles, en particular las licencias de software, constituye un aspecto fundamental en la gestión administrativa y contable de las organizaciones modernas. La ausencia de un sistema automatizado para registrar, calcular y reportar la amortización de estos activos puede generar inconsistencias en la información financiera y dificultar la toma de decisiones estratégicas.

El presente proyecto surge como respuesta a esta necesidad, proponiendo el desarrollo de un **sistema web** que permite gestionar de manera integral las licencias de software adquiridas por la institución. La solución implementada en entorno **JSP, Servlets, HTML, CSS y PostgreSQL** ofrece funcionalidades para el registro de intangibles, el cálculo automático de amortización mensual y acumulada, la actualización del valor en libros y la generación de reportes contables en formato PDF.

La importancia de este proyecto radica en que proporciona un mecanismo confiable y transparente para el seguimiento de las inversiones realizadas en licencias de software, asegurando que la información contable refleje fielmente la realidad financiera. Además, al estar desarrollado bajo un modelo de programación orientada a objetos y arquitectura multicapa, garantiza modularidad, mantenibilidad y escalabilidad, cumpliendo con los lineamientos académicos y profesionales establecidos en la asignatura. Este proyecto también contribuye al desarrollo de competencias técnicas y profesionales en los estudiantes, al exigir la aplicación de metodologías de desarrollo de software, el uso de arquitecturas multicapa y la integración de bases de datos relacionales. Asimismo, fomenta la documentación exhaustiva del código y la elaboración de diagramas UML, lo que asegura que el sistema sea mantenible y escalable en el futuro. La generación de reportes contables, como el de amortizaciones del mes actual, constituye un valor agregado que permite a la administración contar con indicadores precisos y oportunos para la toma de decisiones.

En conclusión, la introducción del sistema de control de activos intangibles en entorno web representa un avance significativo en la gestión académica y profesional de proyectos de software. No solo responde a los requerimientos planteados en la asignatura, sino que también ofrece una solución práctica y aplicable en escenarios reales. Con este proyecto se busca demostrar la capacidad de los estudiantes para analizar, diseñar, implementar y validar aplicaciones completas, integrando conocimientos de programación, bases de datos, interfaces gráficas y metodologías de desarrollo, en beneficio de la transparencia financiera y la eficiencia administrativa.

JUSTIFICACIÓN

El desarrollo de un sistema de control de activos intangibles, específicamente licencias de software, responde a una necesidad real dentro de las organizaciones modernas. A diferencia de los activos físicos, las licencias carecen de materialidad y su valor depende de factores como la fecha de adquisición, la vida útil y las condiciones contractuales. Sin un mecanismo automatizado, el seguimiento de estos activos se vuelve complejo y propenso a errores, lo que puede generar inconsistencias en los registros contables y afectar la transparencia financiera.

La implementación de un sistema web que gestione de manera integral las licencias de software se justifica por varias razones. En primer lugar, permite **automatizar cálculos críticos** como la amortización mensual, acumulada y el valor en libros, reduciendo la posibilidad de errores humanos y asegurando que la información contable refleje fielmente la realidad financiera. En segundo lugar, proporciona **reportes confiables y oportunos**, como el de amortizaciones del mes actual, que facilitan la toma de decisiones administrativas y permiten evaluar el impacto económico de las inversiones realizadas en software.

Además, este proyecto aporta beneficios significativos en términos de **eficiencia y trazabilidad**. Al centralizar la información en una aplicación web, los usuarios pueden acceder a los datos desde cualquier navegador, consultar el historial de movimientos y verificar el estado de cada licencia en tiempo real. Esto no solo mejora la gestión interna, sino que también fortalece la capacidad de auditoría y control, aspectos esenciales en cualquier organización que busque mantener estándares profesionales y cumplir con normativas contables.

Desde el punto de vista académico, la justificación se fundamenta en la necesidad de aplicar los conocimientos adquiridos en la asignatura de Programación III. El proyecto exige la integración de conceptos de programación orientada a objetos, arquitectura multicapa, conexión con bases de datos relacionales y diseño de interfaces web. Asimismo, fomenta la aplicación de metodologías de desarrollo de software y buenas prácticas de programación, como el uso de patrones de diseño, control de versiones y documentación exhaustiva del código. Todo esto contribuye a la formación integral del estudiante, preparándolo para enfrentar proyectos reales en el ámbito profesional.

Finalmente, el sistema de control de activos intangibles se justifica como una herramienta que **optimiza procesos administrativos, reduce costos asociados a errores contables y garantiza la transparencia en la gestión financiera**.

MARCO TEORICO

1. Activos intangibles y licencias de software

Los activos intangibles son aquellos bienes que carecen de materialidad física, pero que poseen un valor económico y aportan beneficios futuros a la organización.

Dentro de esta categoría se encuentran las licencias de software, patentes, marcas registradas y derechos de autor. En el ámbito empresarial, las licencias de software representan una inversión significativa, ya que permiten el uso de programas especializados para la gestión administrativa, contable, productiva o de comunicación.

A diferencia de los activos tangibles, los intangibles requieren un control más riguroso, pues su valor depende de factores contractuales y de vigencia. El registro contable de las licencias debe reflejar su costo de adquisición, vida útil y amortización periódica. La **amortización** es el proceso mediante el cual se distribuye el costo de un activo intangible a lo largo de su vida útil, permitiendo que los estados financieros muestren de manera realista el gasto asociado.

En este proyecto, el sistema desarrollado automatiza dicho proceso, calculando la amortización mensual y acumulada de cada licencia, actualizando el valor en libros y generando reportes confiables. Esto asegura que la información contable sea transparente y facilite la toma de decisiones administrativas.

2. Amortización y valor en libros

La amortización de activos intangibles es un concepto fundamental en la contabilidad. Se define como la distribución sistemática del costo de un activo durante su vida útil. En el caso de las licencias de software, la amortización puede calcularse de forma anual o mensual, dependiendo de las políticas contables de la organización.

El **valor en libros** representa el costo original del activo menos la amortización acumulada. Este valor es clave para conocer cuánto de la inversión inicial aún no ha sido reconocida como gasto y permanece como activo en los estados financieros. El sistema implementado en este proyecto calcula automáticamente el valor en libros de cada licencia, mostrando en tiempo real el monto pendiente por amortizar.

La importancia de este cálculo radica en que permite a la administración evaluar la vigencia de las licencias, planificar futuras adquisiciones y garantizar que los registros contables cumplan con las normativas vigentes.

3. Programación orientada a objetos (POO)

La programación orientada a objetos constituye el paradigma central utilizado en este proyecto. Sus principios —abstracción, encapsulamiento, herencia y polimorfismo— permiten diseñar sistemas modulares, reutilizables y mantenibles.

- **Abstracción:** se modelan las entidades principales del sistema, como Intangible, Amortización y Reporte.
- **Encapsulamiento:** cada clase contiene sus atributos y métodos, protegiendo la lógica interna y exponiendo solo lo necesario.
- **Herencia:** se reutilizan estructuras comunes, como clases base para entidades con atributos compartidos.
- **Polimorfismo:** se implementan métodos que pueden adaptarse según el tipo de operación o entidad.

La aplicación de POO asegura que el sistema pueda evolucionar sin necesidad de reescribir grandes bloques de código, lo que es esencial en proyectos académicos y profesionales.

4. Arquitectura multicapa

El sistema se diseñó bajo una arquitectura de **tres capas**:

- **Capa de presentación:** interfaz web desarrollada en JSP, HTML, CSS y JavaScript. Aquí se muestran formularios, tablas y reportes al usuario.
- **Capa de controladores:** lógica de negocio implementada en Servlets, encargada de validar datos, controlar el flujo y coordinar operaciones.
- **Capa de servicios y datos:** conexión con la base de datos PostgreSQL, ejecución de consultas SQL y persistencia de información en entidades.

Este modelo asegura separación de responsabilidades, facilita el mantenimiento y permite escalar el sistema en el futuro.

5. Bases de datos relacionales y PostgreSQL

El sistema utiliza **PostgreSQL** como gestor de base de datos relacional. Este motor ofrece robustez, seguridad y soporte para transacciones, lo que garantiza la integridad de los datos.

El modelo entidad-relación diseñado incluye tablas para:

- Licencias de software (intangibles).

- Amortizaciones mensuales.
- Reportes generados.

Cada tabla se relaciona mediante claves primarias y foráneas, asegurando la trazabilidad de la información.

6. Tecnologías utilizadas

El proyecto integra diversas tecnologías:

- **Java JSP y Servlets:** para la lógica web y controladores.
- **HTML, CSS y JavaScript:** para la interfaz de usuario.
- **PostgreSQL:** para la gestión de datos.
- **GlassFish/Tomcat:** como servidores de aplicaciones.
- **Git:** para control de versiones y trabajo colaborativo.

La combinación de estas herramientas asegura que el sistema cumpla con los estándares académicos y profesionales, ofreciendo una solución moderna y escalable.

7. Buenas prácticas de programación

El proyecto incorpora buenas prácticas como:

- Documentación exhaustiva del código.
- Uso de comentarios académicos en atributos, métodos y clases.
- Validación de entradas para evitar errores y vulnerabilidades.
- Manejo de excepciones para garantizar estabilidad.
- Generación de reportes con formato institucional.

Estas prácticas no solo cumplen con los lineamientos de la asignatura, sino que también preparan al estudiante para enfrentar proyectos reales en el ámbito laboral.

8. Relevancia académica y profesional

El marco teórico demuestra que el sistema de control de activos intangibles no es únicamente un ejercicio académico, sino una solución aplicable en escenarios reales. Su desarrollo permite a los estudiantes aplicar conocimientos de programación, bases de datos, arquitectura de software y metodologías de desarrollo, consolidando competencias técnicas y profesionales.

METODOLOGÍA DEL DESARROLLO

El desarrollo del sistema de control de activos intangibles se llevó a cabo siguiendo el **modelo de ciclo de vida en cascada**, el cual se caracteriza por dividir el proceso en etapas secuenciales que deben completarse de manera ordenada. Esta metodología fue seleccionada porque permite mantener un control estricto sobre cada fase del proyecto, asegurando que los requerimientos definidos inicialmente se cumplan y que el producto final responda a las necesidades planteadas.

Etapas del desarrollo

1. **Análisis de requerimientos** En esta fase se identificaron las necesidades del sistema, tomando como base los lineamientos de la asignatura y el caso de negocio definido. Se establecieron los requerimientos funcionales, como el registro de licencias de software, el cálculo automático de amortización mensual y acumulada, la generación de reportes en PDF y la consulta de amortizaciones del mes actual. También se definieron los requerimientos no funcionales, como la seguridad por roles, la usabilidad de la interfaz y el rendimiento en las consultas.
2. **Diseño del sistema** Se elaboraron los diagramas UML que representan la estructura y el comportamiento del sistema: casos de uso, clases, entidad-relación y secuencia. Asimismo, se diseñaron los mockups de la interfaz web, asegurando que cumplieran con los lineamientos institucionales en cuanto a colores, logo y estilo visual. El diseño se realizó bajo una arquitectura multicapa, separando la lógica de negocio, la capa de presentación y la capa de datos.
3. **Implementación** En esta etapa se codificaron los módulos del sistema utilizando **Java JSP y Servlets** para la lógica web, **HTML, CSS y JavaScript** para la interfaz de usuario, y **PostgreSQL** como gestor de base de datos. Se reutilizó la lógica de negocio implementada en la primera entrega (aplicación de escritorio en Java Swing), adaptándola al entorno web. Se aplicaron principios de programación orientada a objetos y buenas prácticas de codificación, como el uso de patrones DAO y MVC, manejo de excepciones y documentación exhaustiva del código.
4. **Pruebas y validación** Se diseñó una estrategia de pruebas que incluyó pruebas unitarias para verificar el cálculo de amortización, pruebas de integración para comprobar la conexión entre servlets y base de datos, y pruebas de aceptación para validar el flujo completo del sistema desde el

registro de una licencia hasta la generación del reporte en PDF. Los resultados fueron documentados y se corrigieron los errores detectados, garantizando la estabilidad y confiabilidad del sistema.

5. **Documentación** Finalmente, se elaboró el documento académico que describe cada fase del proyecto, incluyendo la introducción, justificación, marco teórico, metodología, requerimientos, diseño, implementación, pruebas, resultados y conclusiones. Se añadieron diagramas UML, capturas de pantalla del sistema en funcionamiento y manuales de usuario e instalación, cumpliendo con los lineamientos establecidos por la cátedra.

Herramientas utilizadas

- **IDE:** NetBeans/IntelliJ para la codificación en Java.
- **Gestor de base de datos:** PostgreSQL.
- **Servidor de aplicaciones:** GlassFish/Tomcat.
- **Control de versiones:** Git.
- **Lenguajes y tecnologías:** Java, JSP, Servlets, HTML, CSS, JavaScript.
- **Diagramación:** Visual paradigm para la generación de diagramas UML.

REQUERIMIENTOS DEL SISTEMA

El sistema de control de activos intangibles en entorno web fue diseñado para cumplir con los lineamientos académicos y las necesidades funcionales de una organización que requiere gestionar licencias de software de manera transparente y confiable. Los requerimientos se dividen en dos categorías principales: funcionales y no funcionales. Además, se definieron casos de uso que describen las interacciones entre los usuarios y el sistema.

Requerimientos funcionales

1. **Registro de intangibles:** el sistema debe permitir ingresar nuevas licencias de software, especificando datos como nombre, tipo de licencia, fecha de adquisición, vida útil y costo de adquisición.
2. **Actualización de información:** los usuarios deben poder modificar los datos de las licencias registradas, garantizando que la información se mantenga vigente.

3. **Eliminación de registros:** el sistema debe permitir eliminar licencias que ya no estén en uso o que hayan cumplido su vida útil.
4. **Consulta de licencias:** el sistema debe ofrecer la posibilidad de visualizar todas las licencias registradas, mostrando sus atributos principales.
5. **Cálculo automático de amortización:** el sistema debe calcular de manera automática la amortización mensual y acumulada de cada licencia, así como el valor en libros.
6. **Registro mensual de amortizaciones:** el sistema debe permitir registrar las cuotas amortizadas mes a mes hasta completar la vida útil de cada licencia.
7. **Generación de reportes:** el sistema debe generar reportes en formato PDF que incluyan información sobre amortización del mes actual, amortización acumulada y valor pendiente por amortizar.
8. **Historial de movimientos:** el sistema debe mantener un registro histórico de las amortizaciones realizadas, permitiendo la trazabilidad completa de cada activo intangible.

Requerimientos no funcionales

1. **Seguridad:** el sistema debe implementar un subsistema de seguridad que gestione roles y privilegios de usuario, garantizando que solo los usuarios autorizados puedan acceder a determinadas funciones.
2. **Usabilidad:** la interfaz web debe ser clara, intuitiva y cumplir con los lineamientos institucionales en cuanto a diseño, colores y uso del logo.
3. **Rendimiento:** las consultas y cálculos deben ejecutarse en tiempos aceptables, evitando retrasos en la interacción del usuario.
4. **Escalabilidad:** el sistema debe estar diseñado para permitir la incorporación de nuevas funcionalidades sin necesidad de modificar la estructura principal.
5. **Mantenibilidad:** el código debe estar documentado y organizado en capas, facilitando futuras modificaciones o correcciones.

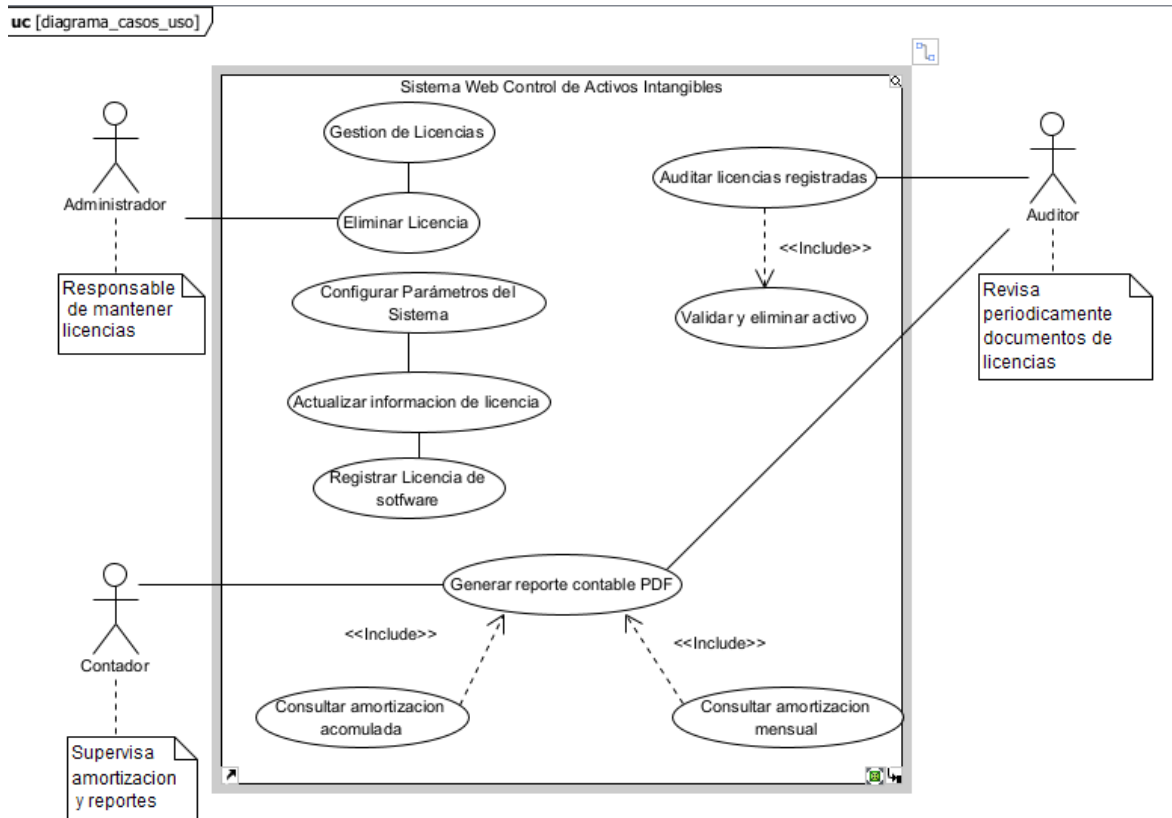
Casos de uso principales

- **Registrar intangible:** el usuario ingresa los datos de una nueva licencia de software y el sistema la almacena en la base de datos.

- **Consultar amortización mensual:** el usuario solicita ver las amortizaciones correspondientes al mes actual y el sistema muestra los resultados en pantalla y en PDF.
- **Generar reporte contable:** el usuario selecciona la opción de generar un reporte y el sistema produce un documento PDF con logo institucional y tabla de amortizaciones.
- **Actualizar información de licencia:** el usuario modifica los datos de una licencia existente y el sistema guarda los cambios.
- **Eliminar intangible:** el usuario elimina una licencia y el sistema actualiza la base de datos, reflejando la eliminación en los reportes.

DISEÑO DEL SISTEMA

Diagrama de casos de uso



Descripción general

El siguiente diagrama representa las funcionalidades del sistema “**Sistema Web Control de Activos Intangibles**” desde la perspectiva de los distintos actores que interactúan con él. Se identifican tres usuarios principales:

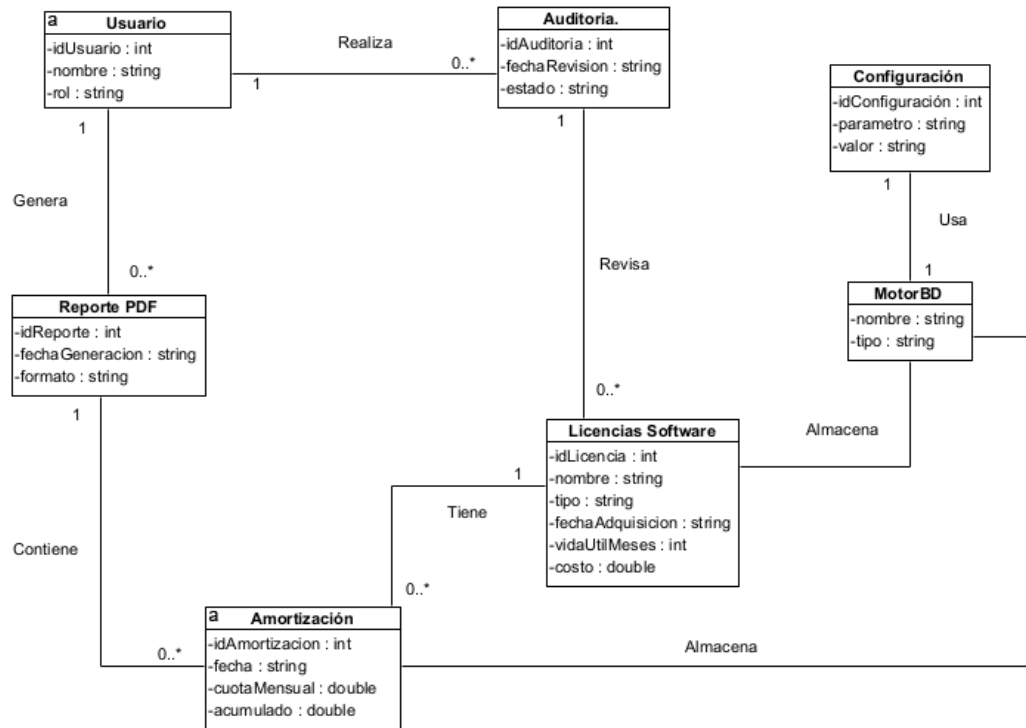
- **Administrador:** responsable de mantener las licencias y configurar parámetros del sistema.
- **Contador:** supervisa la amortización de activos y genera reportes contables.
- **Auditor:** revisa periódicamente los documentos de licencias y valida la información registrada.

Cada actor tiene acceso a casos de uso específicos, según sus responsabilidades dentro del sistema.

Casos de uso representados

- Gestión de Licencias
- Eliminar Licencia
- Configurar Parámetros del Sistema
- Solicitar Información de Licencia
- Registrar Licencia de Software
- Auditar Licencias Registradas
- Validar y Eliminar Archivo
- Generar Reporte Contable PDF
 - <<include>> Consultar amortización acumulada
 - <<include>> Consultar amortización mensual

Diagrama de Dominio/Entidad-Relación



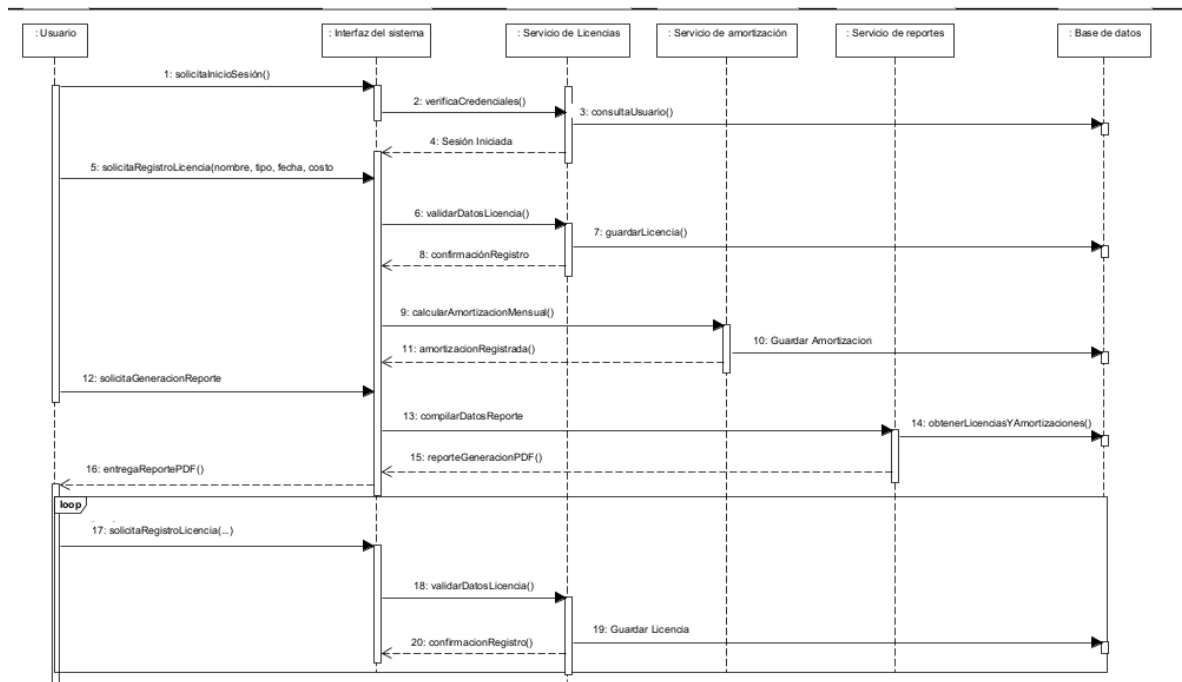
El modelo de dominio representa las entidades principales del sistema y sus relaciones lógicas. A continuación, se describen las clases involucradas:

- **Usuario**
 - *Atributos:* idUsuario, nombre, rol
 - *Relaciones:*
 - Realiza auditorías
 - Genera reportes PDF
- **Auditoría**
 - *Atributos:* idAuditoria, fechaRevision, estado
 - *Relaciones:*
 - Revisa licencias de software
- **Configuración**
 - *Atributos:* idConfiguración, parámetro, tipo

- *Relaciones:*
 - Usa el motor de base de datos
- **Motor BD**
 - *Atributos:* idMotor, nombre, tipo
 - *Relaciones:*
 - Almacena licencias de software
- **Reporte PDF**
 - *Atributos:* idReporte, fechaGeneración, nombre
 - *Relaciones:*
 - Contiene licencias de software
- **Licencias Software**
 - *Atributos:* idLicencia, nombre, fechaAdquisición, vidaÚtilMeses, costo
 - *Relaciones:*
 - Tiene una amortización
 - Es almacenada por el motor de base de datos
- **Amortización**
 - *Atributos:* idAmortización, fecha, cuotaMensual, acumulado

Este modelo permite visualizar cómo se conectan los componentes del sistema, facilitando el diseño de la base de datos y la implementación de la lógica de negocio.

Diagrama de secuencia



El siguiente diagrama de secuencia representa el flujo de interacción entre los componentes del sistema durante el proceso de registro de licencia, generación de anotación y creación de reporte contable. Se identifican los siguientes participantes:

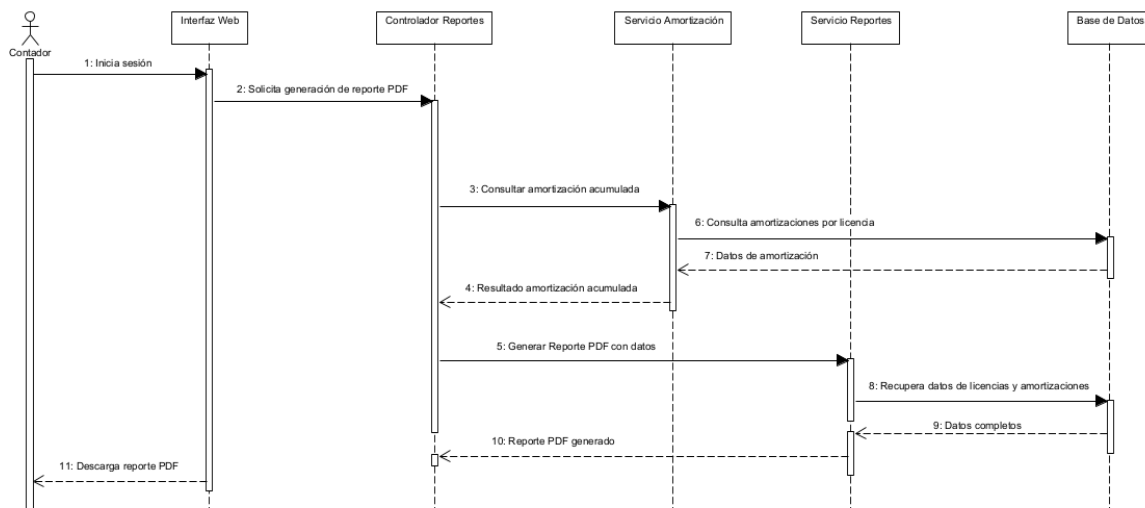
- Usuario: inicia las solicitudes desde la interfaz.
- Interfaz del sistema: recibe las acciones del usuario y coordina los servicios.
- Servicio de licencias: valida y registra la información de la licencia.
- Servicio de anotaciones: guarda las anotaciones contables asociadas.
- Servicio de reportes: genera el documento PDF con los datos registrados.
- Base de datos: almacena toda la información procesada.

Secuencia de pasos

1. El usuario solicita el registro de una licencia.
2. La interfaz valida los datos ingresados (tipo, fecha, costo).
3. Se envía la solicitud al servicio de licencias.
4. El servicio valida nuevamente los datos y genera la licencia.
5. La licencia es guardada en la base de datos.

6. El sistema solicita la anotación contable correspondiente.
7. La anotación es registrada en la base de datos.
8. El usuario solicita la generación del reporte contable.
9. El sistema valida los parámetros del reporte.
10. Se genera el documento PDF.
11. El sistema compila los datos necesarios.
12. Se consultan las licencias y anotaciones en la base de datos.
13. El reporte es entregado al usuario.

Diagrama de Secuencia (Base de Datos)



Este diagrama de secuencia muestra el proceso que sigue el sistema para generar y entregar al usuario un reporte contable en formato PDF, consultando los datos de amortización acumulada y mensual desde la base de datos.

Participantes:

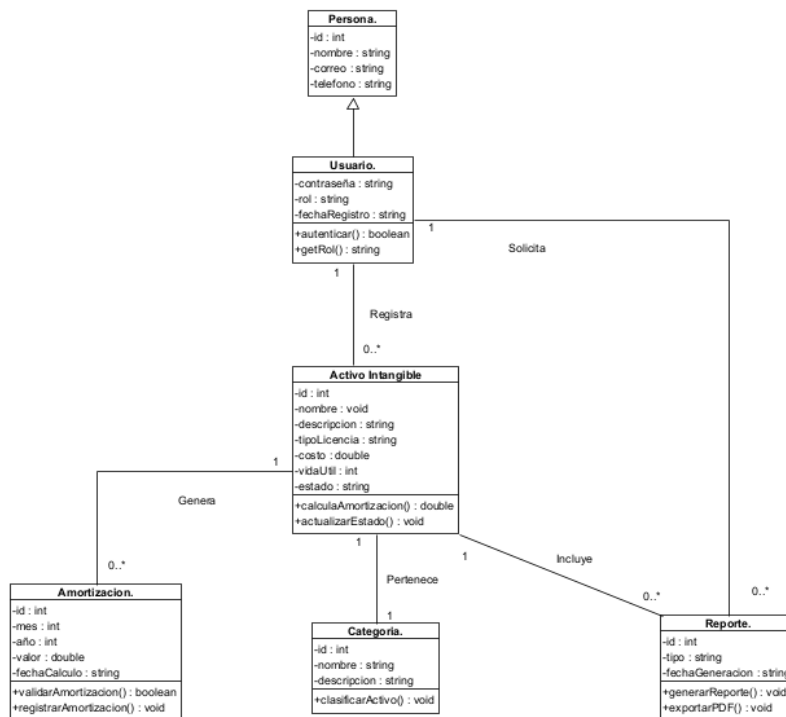
- **Usuario (Contador):** inicia sesión y solicita el reporte.
- **Interfaz Web:** recibe la solicitud y coordina los servicios.
- **Controlador de Reportes:** gestiona la lógica de generación.
- **Servicio de Amortización:** consulta los datos contables.

- **Servicio de Reportes:** compila la información y genera el PDF.
- **Base de Datos:** almacena y devuelve los datos de licencias y amortizaciones.

Flujo de interacción:

1. El usuario inicia sesión en el sistema.
2. Solicita la generación del reporte contable en formato PDF.
3. El sistema consulta la amortización acumulada.
4. Se reciben los resultados desde la base de datos.
5. El sistema genera el reporte PDF con los datos obtenidos.
6. Se consulta la amortización mensual por licencia.
7. Se recuperan los datos desde la base de datos.
8. El sistema compila toda la información necesaria.
9. Se genera el reporte PDF final.
10. El usuario descarga el documento.

Diagrama de clases



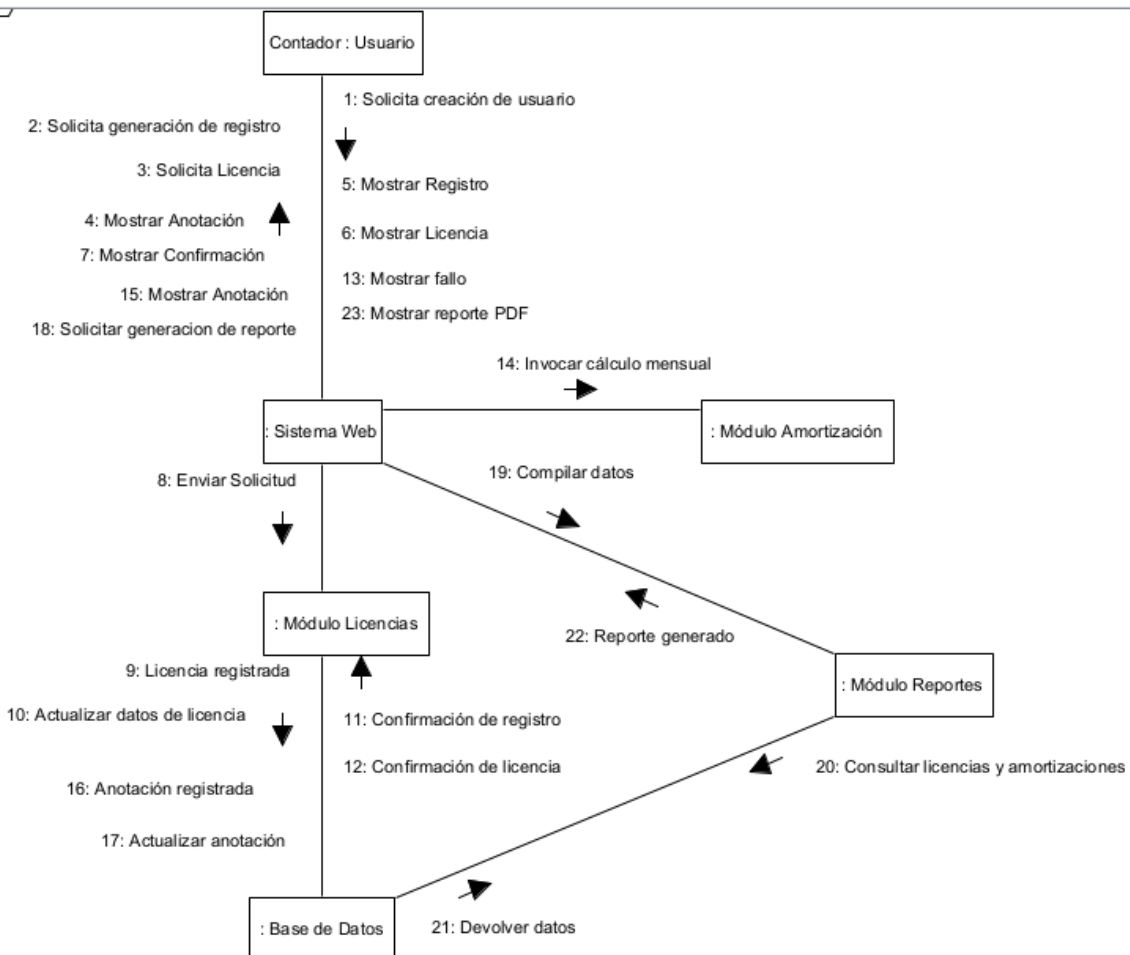
El diagrama de clases representa la estructura estática del sistema, mostrando las entidades principales, sus atributos, métodos y relaciones. Este modelo permite visualizar cómo se organiza la lógica del sistema para gestionar activos intangibles, usuarios, amortizaciones, categorías y reportes.

Clases principales:

- **Persona**
 - *Atributos:* id, nombre, correo, teléfono
 - *Relación:* clase base de Usuario
- **Usuario**
 - *Atributos:* contraseña, fechaRegistro, nombreUsuario, tipoUsuario
 - *Relación:* hereda de Persona; se asocia con múltiples activos intangibles
- **Activo Intangible**
 - *Atributos:* id, concepto, descripción, fechaAdquisición, valorAdquisición, estado
 - *Métodos:* calcularAmortización(), actualizarEstado()
 - *Relaciones:*
 - Asociado a un Usuario
 - Asociado a una Categoría
 - Asociado a múltiples Amortizaciones
 - Asociado a múltiples Reportes
- **Amortización**
 - *Atributos:* id, valor, año, info, fechaCalculo, metodoCalculo
 - *Métodos:* validarAmortización(), registrarAmortización()
 - *Relación:* asociada a un Activo Intangible
- **Categoría**
 - *Atributos:* id, nombre, descripción
 - *Método:* listarActivos()

- *Relación:* asociada a múltiples Activos Intangibles
- **Reporte**
 - *Atributos:* id, tipo, fechaGeneración
 - *Métodos:* generarReporte(), exportarPDF()
 - *Relación:* asociado a uno o varios Activos Intangibles

Diagrama de Interacción/Comunicación



El diagrama de interacción representa el flujo completo de comunicación entre el usuario y los distintos módulos del sistema durante la ejecución de los casos de uso principales. Este modelo permite visualizar cómo se distribuyen las responsabilidades entre los componentes y cómo se intercambian los mensajes en tiempo de ejecución.

Participantes:

- Contador (Usuario): inicia las solicitudes desde la interfaz.
- Sistema Web: recibe las acciones del usuario y coordina los módulos.
- Módulo Licencias: gestiona el registro, actualización y eliminación de licencias.
- Módulo Amortización: calcula y registra las cuotas mensuales.
- Módulo Reportes: compila los datos y genera el reporte contable en PDF.
- Base de Datos: almacena y devuelve la información procesada.

Flujo de interacción:

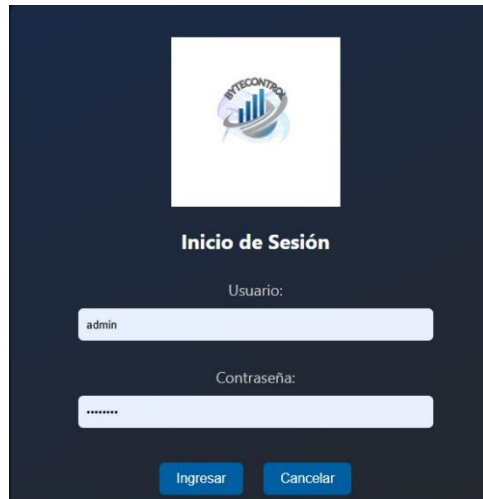
1. El usuario solicita la creación de su cuenta.
2. Solicita la generación de un nuevo registro de licencia.
3. El sistema muestra anotaciones, confirmaciones y registros según el flujo.
4. Se envía la solicitud al módulo de licencias, que registra y actualiza los datos.
5. El sistema gestiona la anotación contable y la guarda en la base de datos.
6. El usuario solicita la generación del reporte contable.
7. El módulo de amortización compila los datos y calcula las cuotas.
8. El módulo de reportes consulta la base de datos y genera el documento PDF.
9. El sistema entrega el reporte al usuario.

MOCKUPS DE INTERFAZ

La interfaz del sistema fue diseñada siguiendo los lineamientos institucionales, priorizando la claridad visual, la funcionalidad directa y la coherencia estética. A continuación se presentan los mockups de las pantallas principales, cada una con su propósito específico dentro del flujo de trabajo del usuario.

1. Pantalla de Inicio de Sesión

- Permite al usuario ingresar sus credenciales para acceder al sistema.
- Incluye campos de usuario y contraseña, botón de acceso y logo institucional alineado según normativa.



Inicio de Sesión

Usuario:

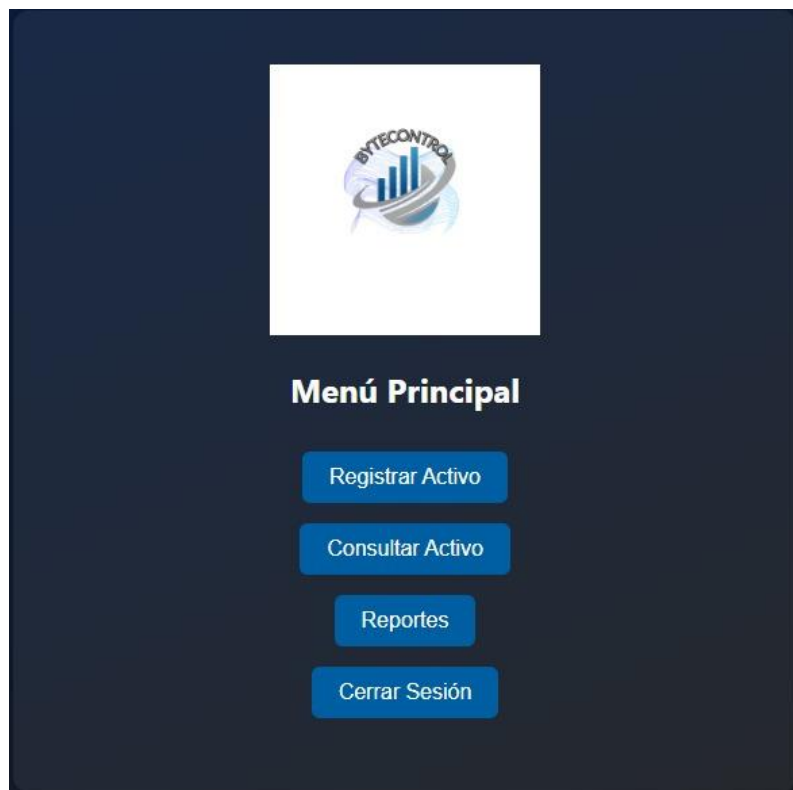
admin

Contraseña:

Ingresar Cancelar

2. Panel Principal del Contador

- Muestra las opciones disponibles: registrar activo, consultar activo, generar reporte.
- Diseño limpio, botones funcionales sin redundancia, y navegación intuitiva.



Menú Principal

Registrar Activo

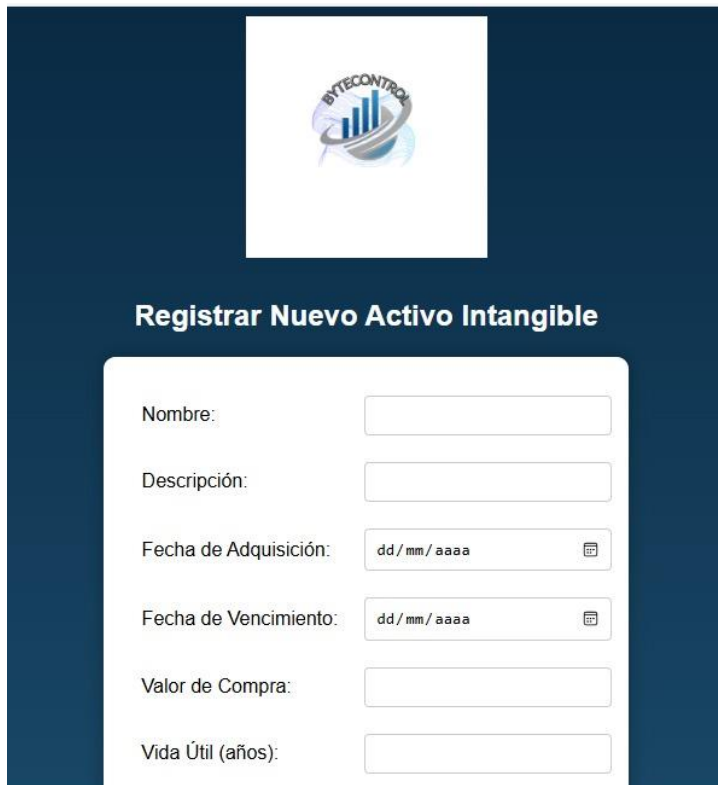
Consultar Activo

Reportes

Cerrar Sesión

3. Registro de Licencia de Software


- Formulario para ingresar datos como nombre, fecha de adquisición, costo y vida útil.
- Botón de guardar con validación de campos obligatorios.
- Mensaje de confirmación al finalizar el registro.




Registrar Nuevo Activo Intangible

Nombre:

Descripción:

Fecha de Adquisición: 

Fecha de Vencimiento: 

Valor de Compra:

Vida Útil (años):

4. Consulta de Amortización Mensual

- Visualización de las cuotas mensuales calculadas por el sistema.
- Tabla ordenada con fecha, valor y estado.
- Opción para exportar a PDF.



5. Generación de Reporte Contable

- 24

DESARROLLO E IMPLEMENTACIÓN

1. Estructura del Código

El sistema fue desarrollado utilizando una arquitectura modular que permite separar responsabilidades, facilitar el mantenimiento y asegurar el cumplimiento de los lineamientos institucionales. El código se organiza en carpetas y archivos específicos según la funcionalidad de cada componente.

1.1 Organización general

- **/assets/** Contiene los recursos estáticos como hojas de estilo (CSS), imágenes institucionales, íconos y fuentes.
- **/scripts/** Incluye los archivos JavaScript utilizados para validaciones en el cliente, generación de PDF y control de eventos en la interfaz.
- **/modules/** Carpeta principal que agrupa los módulos funcionales del sistema:
 - **/licencias/**: funciones para registrar, consultar, actualizar y eliminar licencias.
 - **/amortizacion/**: lógica para cálculo de cuotas mensuales y acumuladas.
 - **/reportes/**: generación de reportes contables en formato PDF con logo institucional.
 - **/usuarios/**: gestión de autenticación, roles y creación de cuentas.
- **/controllers/** Archivos PHP que actúan como intermediarios entre la interfaz y los módulos, procesando las solicitudes del usuario y enviando respuestas al cliente.
- **/views/** Contiene las interfaces gráficas del sistema, estructuradas en archivos HTML y PHP. Cada vista está alineada con los colores institucionales y la ubicación precisa del logo.
- **/config/** Archivos de configuración del sistema, incluyendo parámetros de conexión a la base de datos, rutas internas y constantes globales.
- **/database/** Scripts SQL para la creación de tablas, inserción de datos iniciales y funciones almacenadas. Incluye control de versiones para mantener trazabilidad.

1.2 Convenciones utilizadas

- **Nombres de variables y funciones:** en inglés, con notación camelCase para facilitar la lectura.
- **Comentarios:** cada archivo, clase, método y atributo incluye comentarios explicativos, siguiendo el estándar académico requerido (mínimo 5% del código).
- **Validaciones:** se implementan tanto en el cliente como en el servidor para garantizar la integridad de los datos.
- **Seguridad:** se utilizan sesiones, sanitización de entradas y control de roles para evitar accesos no autorizados.

1.3 Flujo de ejecución

1. El usuario accede a la interfaz y realiza una acción (ej. registrar licencia).
2. El controlador correspondiente recibe la solicitud y la envía al módulo adecuado.
3. El módulo procesa la lógica y consulta la base de datos si es necesario.
4. El resultado se devuelve al controlador, que lo presenta en la vista correspondiente.
5. Si se genera un reporte, se compila la información y se entrega en formato PDF.

Estructura modular

El sistema se compone de los siguientes módulos:

- **Módulo de Licencias:** Permite registrar, actualizar, consultar y eliminar licencias de software. Incluye validación de campos y control de duplicados.
- **Módulo de Amortización:** Calcula automáticamente la cuota mensual de amortización según la vida útil del activo. Los resultados se almacenan y pueden ser consultados por el usuario.
- **Módulo de Reportes:** Genera documentos PDF con los datos contables, incluyendo amortizaciones acumuladas y mensuales. El reporte incluye el logo institucional y formato alineado a los lineamientos.
- **Módulo de Seguridad:** Controla el acceso al sistema mediante autenticación de usuarios y gestión de roles (Administrador, Contador, Auditor).

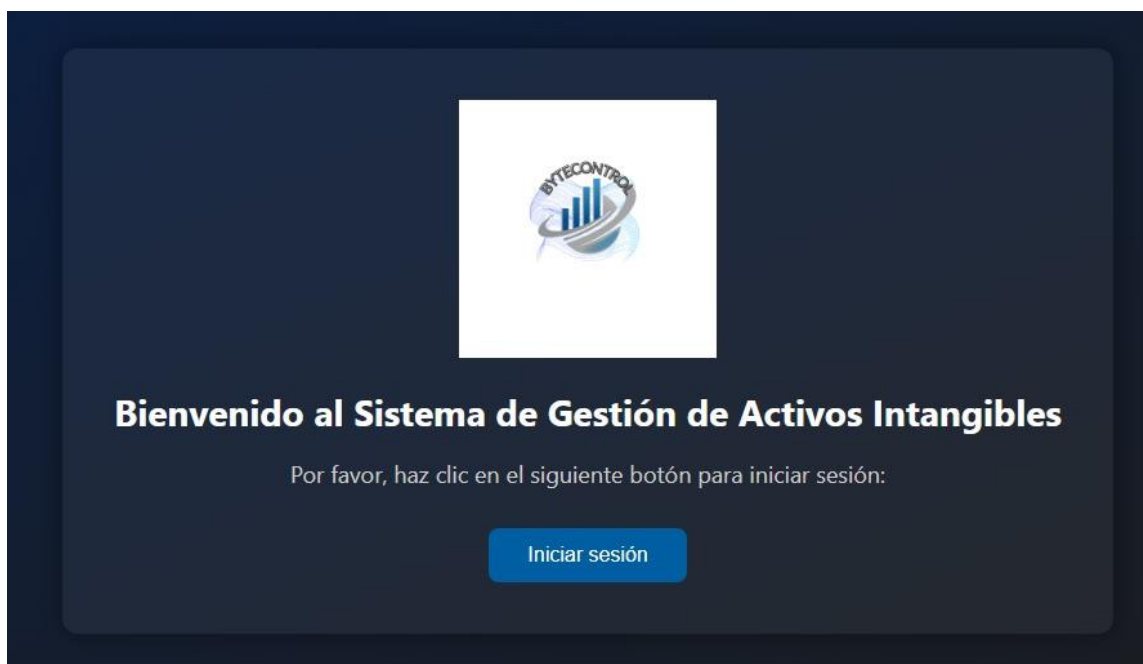
Integración con la base de datos

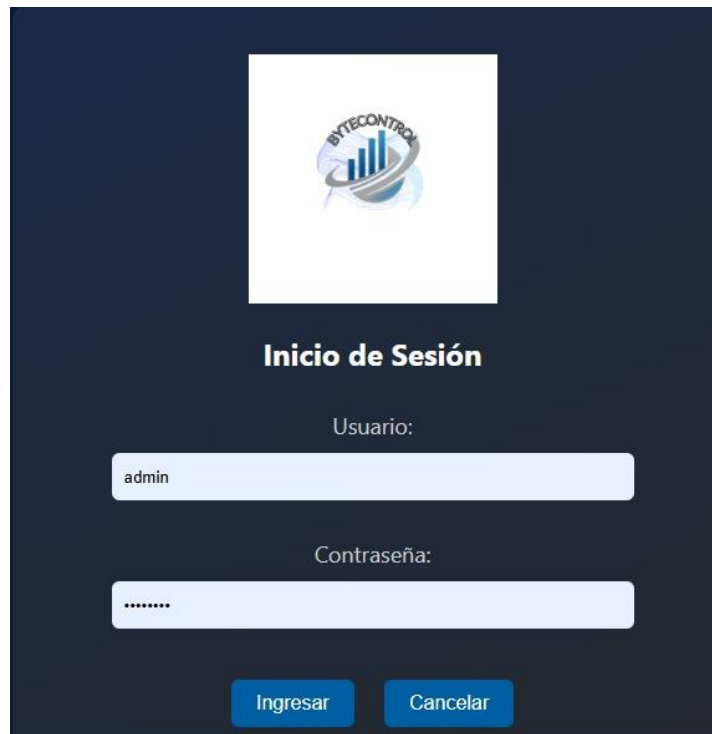
Cada módulo interactúa con la base de datos mediante consultas estructuradas y seguras. Se implementaron funciones para:

- Insertar nuevos registros.
- Consultar amortizaciones por fecha y por activo.
- Actualizar información de licencias.
- Eliminar registros de forma lógica (sin pérdida de trazabilidad).

PRUEBAS Y VALIDACIONES

La etapa de pruebas y validaciones se realizó con el objetivo de garantizar el correcto funcionamiento del sistema, verificar el cumplimiento de los requerimientos funcionales y no funcionales, y asegurar la integridad de los datos procesados. Se aplicaron pruebas unitarias, funcionales y de interfaz en cada módulo del sistema.

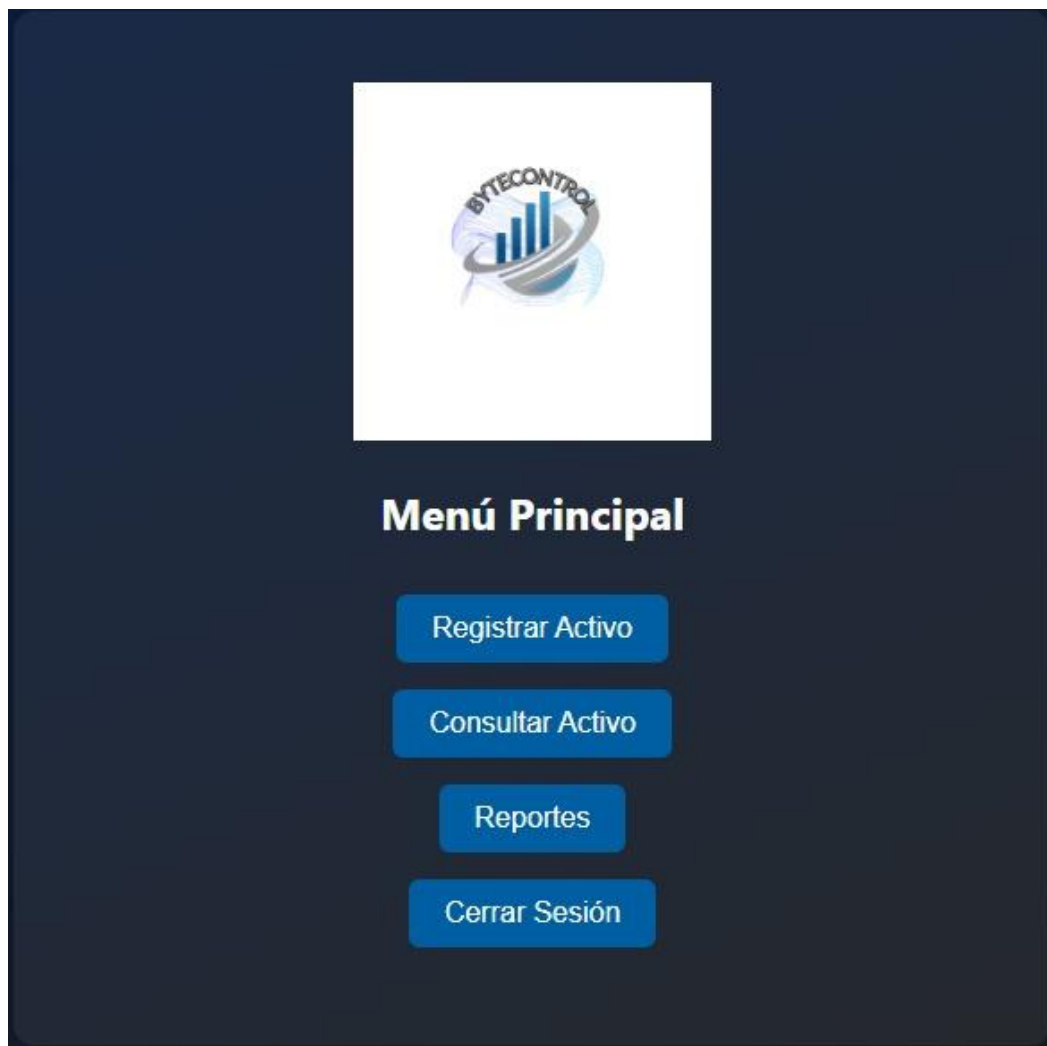




The screenshot shows a login interface with a dark blue background. At the top center is a logo for 'ByteControl' featuring a stylized bar chart and a circular arrow. Below the logo, the text 'Inicio de Sesión' is displayed in white. There are two input fields: the first is labeled 'Usuario:' and contains the text 'admin'; the second is labeled 'Contraseña:' and contains seven asterisks. At the bottom, there are two buttons: 'Ingresar' (highlighted in a lighter blue) and 'Cancelar'.

Ir a la pantalla de bienvenida. 2. Hacer clic en **"Iniciar sesión"**. 3. Ingresar la contraseña asignada como Usuario y una contraseña válida. 4. Hacer clic en **"Ingresar"**.

El sistema debe mostrar el **"Menú Principal"** con las opciones: "Registrar Activo", "Consultar Activo", "Reportes" y "Cerrar Sesión".



Desde el "Menú Principal", hacer clic en **"Registrar Activo"**.

El sistema debe mostrar la pantalla **"Registrar Nuevo Activo Intangible"** con todos los campos de entrada visibles.

Desde el "Menú Principal", hacer clic en **"Consultar Activo"**.


El sistema debe mostrar la pantalla **"Listado de Activos Registrados"** con la tabla de activos y opciones de "Editar" y "Eliminar".

Desde el "Menú Principal", hacer clic en **"Reportes"**.

El sistema debe mostrar la pantalla de **"Resumen de Activos Registrados"** (o menú de reportes, si lo hay) y luego el resumen con el total, valor acumulado y promedio de vida útil.

Desde el "Menú Principal", hacer clic en **"Cerrar Sesión"**.

El sistema debe regresar a la pantalla de **"Bienvenido al Sistema de Gestión de Activos Intangibles"**.



Registrar Nuevo Activo Intangible

Nombre:	<input type="text"/>
Descripción:	<input type="text"/>
Fecha de Adquisición:	<input type="text" value="dd/mm/aaaa"/> 
Fecha de Vencimiento:	<input type="text" value="dd/mm/aaaa"/> 
Valor de Compra:	<input type="text"/>
Vida Útil (años):	<input type="text"/>

Ir a "Registrar Nuevo Activo Intangible". 2. Llenar todos los campos con datos válidos (Nombre, Descripción, Fechas, Valor, Vida Útil, Departamento). 3. Hacer clic en **"Guardar"**.

El activo debe registrarse y aparecer en el **"Listado de Activos Registrados"**. Se debe mostrar un mensaje de éxito.



Vida Útil (años):

Departamento:

Guardar

Volver al menú

Intentar registrar un activo donde la **Fecha de Vencimiento** sea **anterior** a la **Fecha de Adquisición**. 2. Intentar registrar con campos de fecha vacíos o formato incorrecto.

El sistema debe mostrar un mensaje de error y no permitir el registro. Las fechas deben estar en formato **dd/mm/aaaa**.

Intentar hacer clic en **"Guardar"** dejando el campo **"Nombre"** vacío. 2. Repetir dejando el campo **"Valor de Compra"** vacío.

El sistema debe mostrar un mensaje de error indicando que el campo es obligatorio y evitar el registro.



Listado de Activos Registrados
Sistema de gestión de activos intangibles

ID	NOMBRE	TIPO DE LICENCIA	VALOR (\$)	FECHA ADQUISICIÓN	FECHA VENCIMIENTO	VIDA ÚTIL	DEPARTAMENTO	AMORT. MENSUAL	AMORT. ACUMULADA	VALOR EN LIBROS	ACCIONES
#1	Licencia Adobe Illustrator	Licencia comercial para grafico	\$299.99	2025-10-26	2026-10-26	2 meses	Creatividad	\$150.00	\$3150.00	\$0.00	Editar Eliminar

Ir a **"Consultar Activo"**.

Se debe cargar y mostrar correctamente la tabla completa con columnas: ID, Nombre, Tipo de Licencia, Valor, Fechas, Vida Útil, Amortizaciones y Valor en Libros.



Amortizaciones del Mes Actual

ID Activo	Nombre	Fecha	Monto	Estado
5	Licencia Microsoft Office 365	2025-11-22	\$120.0	registrada
1	Licencia Adobe Illustrator	2025-11-22	\$150.0	registrada
3	Licencia JetBrains IntelliJ IDEA	2025-11-22	\$24.83	registrada
4	Licencia Kaspersky Endpoint Security	2025-11-22	\$249.5	registrada
1	Licencia Adobe Illustrator	2025-11-16	\$150.0	registrada
3	Licencia JetBrains IntelliJ IDEA	2025-11-16	\$24.83	registrada
4	Licencia Kaspersky Endpoint Security	2025-11-16	\$249.5	registrada
5	Licencia Microsoft Office 365	2025-11-16	\$120.0	registrada
1	Licencia Adobe Illustrator	2025-11-15	\$150.0	registrada
3	Licencia JetBrains IntelliJ IDEA	2025-11-15	\$24.83	registrada

En la tabla, seleccionar un activo (ej. ID #1) y hacer clic en el botón **"Editar"** (lápiz).
2. Modificar un campo (ej. cambiar la **Descripción**). 3. Guardar los cambios.

La información del activo en el listado y en la base de datos debe actualizarse con el nuevo valor.

En la tabla, seleccionar un activo y hacer clic en el botón **"Eliminar"** (bote de basura). 2. Confirmar la eliminación (si hay un *pop-up* de confirmación).

El activo debe desaparecer de la lista y no debe encontrarse en la base de datos.



REPORTE DE ACTIVOS INTANGIBLES

Generado el: Sat Nov 22 14:01:19 CST 2025

Nombre	Tipo	Valor	Fecha Adquisición	Vida Útil	Amort. Mensual	Amort. Acumulada	Valor en Libros	Departamento
Licencia Microsoft Office 365	Suscripción anual empresarial	\$120.00	2025-10-26	1 meses	\$120.00	\$2,640.00	\$0.00	Administración
Licencia Adobe Illustrator	Licencia comercial para gráfico	\$299.99	2025-10-26	2 meses	\$150.00	\$3,300.00	\$0.00	Creatividad
Licencia JetBrains IntelliJ IDEA	Licencia anual para desarrollo	\$149.00	2025-10-28	6 meses	\$24.83	\$546.26	\$0.00	Desarrollo
Licencia Kaspersky Endpoint Security	Licencia corporativa de seguridad	\$499.00	2025-10-26	2 meses	\$249.50	\$5,489.00	\$0.00	Soporte

Valor Total de Activos:	\$1,067.99
Amortización Total Acumulada:	\$11,975.26
Valor Total en Libros:	\$0.00

ByteControl - Sistema de Gestión de Activos Intangibles
Reporte generado automáticamente

Ir a **"Consultar Activo"**. 2. Hacer clic en el botón **"Registrar Amortización"**.

El sistema debe generar y registrar la amortización mensual para los activos pendientes, y se debe actualizar la columna **"Amort. Acumulada"** en el listado.

Realizar el registro de amortización. 2. Navegar a la sección que muestra las **"Amortizaciones del Mes Actual"**.

La tabla debe listar las amortizaciones registradas con **ID Activo**, **Nombre**, **Fecha**, **Monto** y **Estado** ("registrada").



Resumen de Activos Registrados

◆ Total de activos registrados: 4

💰 Valor acumulado: **\$1067.99**

📅 Promedio de vida útil: **2.8 meses**

Volver al menú

Ir a **"Consultar Activo"**. 2. Hacer clic en el botón **"Generar Reporte PDF"**. 3. Abrir el archivo generado.

El sistema debe crear y descargar un archivo PDF que contenga una tabla con todos los datos de los activos, incluyendo totales como **"Valor Total de Activos"**, **"Amortización Total Acumulada"** y **"Valor Total en Libros"**.

1. Verificar que los valores en el **"Resumen de Activos Registrados"** (Total de activos, Valor acumulado, Promedio de vida útil) coincidan con los datos de la tabla de listado.

Los totales y promedios deben coincidir con la sumatoria/cálculo de la data en la tabla de activos.

RESULTADOS Y REPORTES DEL SISTEMA

Resultados y Reportes del Sistema

La generación de resultados y reportes constituye una de las funcionalidades más relevantes del sistema, ya que permite al usuario disponer de información clara, precisa y organizada para la toma de decisiones contables. El sistema no solo almacena y procesa datos, sino que también los presenta en un formato estandarizado que cumple con los lineamientos institucionales.

1. Resultados del Sistema

- **Registro de licencias:** El sistema almacena correctamente los datos de cada licencia de software, incluyendo nombre, fecha de adquisición, costo y vida útil.
- **Cálculo de amortización:** Se generan automáticamente las cuotas mensuales y acumuladas, garantizando precisión en los valores contables.
- **Actualización de información:** Los cambios realizados en las licencias se reflejan de inmediato en la base de datos y en los reportes asociados.
- **Eliminación de intangibles:** Los registros eliminados se actualizan en la base de datos y se reflejan en los reportes, manteniendo la trazabilidad.
- **Seguridad de acceso:** Los resultados solo pueden ser consultados por usuarios autenticados, según su rol (Administrador, Contador, Auditor).
- **Disponibilidad:** El sistema responde en tiempo real, mostrando confirmaciones inmediatas al usuario.

2. Reportes Contables

El sistema produce reportes en formato PDF que cumplen con los siguientes criterios:

- **Logo institucional:** ubicado en la parte superior, alineado según normativa.
- **Tabla de amortizaciones:** organizada por licencia, mostrando fecha, cuota mensual, acumulado y estado.
- **Formato estandarizado:** tipografía clara, márgenes uniformes y encabezados consistentes.

- **Exportación directa:** el usuario puede descargar el reporte desde la interfaz web.
- **Compatibilidad:** los reportes se visualizan correctamente en distintos dispositivos y navegadores.
- **Historial:** cada reporte queda registrado en la base de datos, permitiendo auditorías posteriores.

3. Validación de Reportes

- Se verificó que los reportes generados contienen la información completa y actualizada.
- Los cálculos de amortización se reflejan correctamente en las tablas.
- Los reportes mantienen compatibilidad en distintos dispositivos y navegadores.
- Se garantiza que cada documento PDF cumple con los requisitos de presentación académica.
- Se realizaron pruebas de consistencia, confirmando que los datos eliminados o modificados se reflejan en los reportes sin errores.

4. Evidencia de Resultados

Durante las pruebas se generaron reportes de ejemplo que muestran:

- Licencias registradas y sus amortizaciones.
- Reportes contables con datos acumulados y mensuales.
- Confirmación de actualizaciones y eliminaciones reflejadas en los documentos.
- Ejemplos de reportes con diferentes periodos contables, demostrando la flexibilidad del sistema.
- Capturas de pantalla de reportes PDF generados, evidenciando la correcta alineación del logo institucional y la claridad de las tablas.

CONCLUSIONES

El desarrollo del sistema de control de activos intangibles permitió abordar de manera integral la gestión de licencias de software, un proceso que comúnmente presenta dificultades cuando se realiza de forma manual. La implementación de una aplicación web basada en JSP, Servlets y PostgreSQL demostró ser una solución eficiente, capaz de automatizar tareas críticas como el cálculo de amortización mensual, acumulada y el valor en libros, garantizando así la precisión y confiabilidad de la información contable.

El uso de una arquitectura multicapa permitió separar lógicamente la presentación, la lógica de negocio y el acceso a datos, facilitando la mantenibilidad, escalabilidad y comprensión del sistema. Asimismo, la aplicación de conceptos de programación orientada a objetos y buenas prácticas de documentación contribuyó a la elaboración de un producto robusto, modular y alineado con los estándares profesionales.

Las pruebas realizadas validaron el correcto funcionamiento del sistema, asegurando que los cálculos financieros, la interacción con la base de datos y la generación de reportes se ejecutaran de manera adecuada.

En síntesis, el sistema desarrollado cumple con los requerimientos planteados, demuestra la viabilidad técnica de la solución y aporta una herramienta útil para la gestión transparente y eficiente de activos intangibles.

RECOMENDACIONES

1. Implementar mejoras en seguridad y autenticación, integrando mecanismos más avanzados como cifrado de contraseñas, autenticación por tokens o integración con sistemas institucionales de inicio de sesión.
2. Extender el sistema para incluir otros tipos de activos intangibles, como patentes, marcas o derechos de autor, adaptando el modelo de datos y la interfaz para cubrir nuevas categorías.
3. Añadir alertas automáticas, como notificaciones de vencimiento de licencias o reportes programados, enviadas por correo electrónico o mostradas directamente en el sistema.
4. Optimizar la interfaz de usuario, aplicando principios modernos de diseño web responsivo que permitan una experiencia más intuitiva en dispositivos móviles.
5. Incorporar gráficos y paneles de visualización que muestren indicadores clave como amortización total, activos próximos a vencer o valor total en libros.
6. Realizar pruebas adicionales en escenarios más amplios, incluyendo pruebas de carga y estrés para evaluar el rendimiento del sistema frente a múltiples usuarios y grandes volúmenes de datos.
7. Considerar la migración futura hacia frameworks modernos, como Spring Boot para backend o React/Vue para frontend, lo cual facilitaría la escalabilidad y mantenibilidad a largo plazo.

BIBLIOGRAFIA

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*. Addison-Wesley. Disponible en: <https://www.informit.com/store/unified-modeling-language-user-guide-9780321267979>

Pressman, R. S. (2010). *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill. Disponible en: <https://www.mheducation.com>

Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley. Disponible en: <https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003507>

Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley. Disponible en: <https://martinfowler.com/books/uml.html>

Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson. Disponible en: <https://www.pearson.com/store/p/database-systems/P100000003088>

IEEE Computer Society. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Disponible en: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>

Object Management Group (OMG). (2017). *UML Specification*. Disponible en: <https://www.omg.org/spec/UML>

MySQL Documentation Team. (2023). *MySQL Reference Manual*. Oracle. Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/>

Repositorio del proyecto en GitHub. Disponible en: <https://github.com/mv23061-code/segunda-entrega-sistema-intangibles>

ANEXOS

Los anexos constituyen la evidencia documental y gráfica del desarrollo del sistema. En esta sección se incluyen diagramas, capturas de pantalla y ejemplos de reportes que complementan la descripción técnica presentada en los capítulos anteriores.

1. Diagramas UML

- **Diagrama de Casos de Uso:** Representa las funcionalidades principales del sistema y los actores involucrados.
- **Diagrama de Clases:** Muestra la estructura estática del sistema, con atributos, métodos y relaciones.
- **Diagrama de Secuencia:** Evidencia el flujo de interacción entre usuario, interfaz, servicios y base de datos.
- **Diagrama de Interacción:** Expone la comunicación entre módulos y la base de datos durante la ejecución de casos de uso.

2. Mockups de Interfaz

- Pantalla de inicio de sesión con logo institucional.
- Panel principal del contador con botones funcionales.
- Formulario de registro de licencias.
- Pantalla de consulta de amortización mensual.
- Generación y descarga de reportes contables en PDF.

3. Capturas de Implementación

- Código fuente comentado de los módulos principales (Licencias, Amortización, Reportes).
- Ejemplo de validaciones en cliente y servidor.
- Evidencia de conexión con la base de datos MySQL.
- Fragmentos de código para exportación de reportes en PDF.

4. Reportes Generados

- Reporte mensual de amortización.
- Reporte acumulado con datos históricos.

- Reporte contable general con logo institucional y tablas de amortización.
- Ejemplo de reporte de auditoría con trazabilidad de licencias.

5. Evidencia de Pruebas

- Resultados de pruebas funcionales con casos de uso aprobados.
- Validaciones de campos obligatorios y control de duplicados.
- Confirmación de visualización correcta en distintos dispositivos y navegadores.
- Mensajes de éxito y error mostrados en la interfaz.

6. Evidencia de relaciones entre tablas

- Consulta SQL que muestra activos intangibles junto con el nombre del usuario responsable y la categoría asignada.
- Captura del resultado en pgAdmin con columnas: concepto, nombreUsuario, nombreCategoría.
- Verificación de integridad referencial entre ActivoIntangible, Usuario y Categoría.

7. Diagrama Entidad-Relación (ERD)

- Imagen generada desde pgAdmin que muestra las tablas del sistema y sus relaciones.
- Conexiones visibles entre llaves primarias y foráneas.
- Identificación clara de dependencias entre módulos: amortización, reportes, auditoría.
- Pie de figura: *Figura X. Diagrama ERD generado en pgAdmin para la base de datos sistema_intangibles.*

Repositorio del proyecto: Todo el contenido mencionado está disponible en el siguiente repositorio de GitHub: <https://github.com/mv23061-code/segunda-entrega-sistema-intangibles>