



# ESCUELA POLITÉCNICA NACIONAL

---

## FACULTAD DE INGENIERÍA DE SISTEMAS

### Manual Técnico Aplicación “Paccalogico”

#### ESTUDIANTES:

Angulo Javier

Anrango Stalin

Barahona Ángel

Caicedo Erick

Calles Kevin

Calvache Mateo

#### DOCENTE:

Patricio Michael Paccha Angamarca

QUITO - 2024



# Contenido

<b>INTRODUCCIÓN .....</b>	<b>3</b>
1.1    Objetivo .....	3
1.2    Alcance.....	3
<b>ARQUITECTURA DEL SISTEMA.....</b>	<b>3</b>
2.1    Visión General .....	3
2.2    Componentes del Sistema.....	3
2.3    Tecnologías utilizadas .....	4
<b>DISEÑO DEL SISTEMA .....</b>	<b>5</b>
3.1    Arquitectura de Software .....	5
3.2    Base de Datos .....	6
<b>REQUISITOS DEL SISTEMAS .....</b>	<b>7</b>
4.1    Requisitos de Software .....	7
4.2    Requisitos de hardware .....	7
<b>IMPLEMENTACIÓN .....</b>	<b>7</b>
5.1    Desarrollo de Componentes .....	7
5.2    Integración de Componentes.....	8
<b>OPERACIÓN Y MANTENIMIENTO .....</b>	<b>8</b>
6.1    Procedimientos de Operación.....	8
6.2    Mantenimiento .....	9
<b>CONSIDERACIONES DE SEGURIDAD.....</b>	<b>10</b>
7.1    Seguridad de la Aplicación .....	10
<b>REFERENCIAS .....</b>	<b>11</b>

# INTRODUCCIÓN

## 1.1 Objetivo

Facilita la implementación, el mantenimiento y la optimización del Zoológico Virtual al proporcionar una guía exhaustiva para desarrolladores, administradores de sistemas y usuarios finales. Capacita a los usuarios para maximizar la funcionalidad, eficiencia y seguridad del sistema, abordando desafíos técnicos, estrategias de optimización y resolución de problemas a través de ejemplos detallados y consejos prácticos.

## 1.2 Alcance

- Estrategias de testing y depuraciones específicas para cada componente del sistema.
- Ejemplos de uso y escenarios de casos de prueba para validar la funcionalidad del sistema.
- Guías para la integración con otros sistemas o servicios, como sistemas de autenticación externos o APIs de terceros para funcionalidades adicionales.

# ARQUITECTURA DEL SISTEMA

## 2.1 Visión General

La arquitectura cliente-servidor del Zoológico Virtual es fundamental para su escalabilidad y seguridad. Esta estructura permite una distribución eficiente de la carga y la implementación de actualizaciones centralizadas, lo que asegura un rendimiento óptimo del sistema. La utilización de TLS para el cifrado de datos en tránsito y la aplicación de sistemas robustos de autenticación protegen de manera efectiva contra accesos no autorizados, garantizando la integridad y confidencialidad de la información.

La comunicación entre el cliente y el servidor, gracias a JDBC, facilita la integración con SQLite y proporciona flexibilidad para futuras tecnologías, destacando su importancia crítica para la escalabilidad y seguridad del Zoológico Virtual.

## 2.2 Componentes del Sistema

La arquitectura cliente-servidor del Zoológico Virtual optimiza tanto la escalabilidad como la seguridad del sistema, permitiendo una gestión eficiente de recursos y facilitando actualizaciones centralizadas. La separación entre cliente y servidor habilita una distribución de carga eficaz, con mejoras aplicadas directamente en el servidor para garantizar que todos los usuarios accedan a la versión más actual sin necesidad de actualizaciones individuales. En

cuanto a la seguridad, se emplean protocolos seguros como TLS para el cifrado de datos en tránsito y sistemas de autenticación y autorización robustos en el servidor, asegurando la protección de los datos y la integridad del sistema.

Los componentes principales del sistema del Zoológico Virtual comprenden:

- **Frontend:** Una interfaz gráfica de usuario (GUI) desarrollada en Java, utilizando JavaFX para proporcionar una experiencia de usuario rica y dinámica. Este componente actúa como el punto de interacción con el usuario, permitiendo la navegación, visualización y gestión de la información de los animales y sus hábitats.
- **Backend:** La lógica de negocio se implementa en Java y se encarga de procesar las solicitudes del usuario, ejecutar la lógica de aplicación y comunicarse con la base de datos. Utiliza patrones de diseño como el Data Access Object (DAO) para abstraer y manejar las operaciones de la base de datos.
- **Base de Datos:** SQLite sirve como el sistema de gestión de base de datos, almacenando toda la información relacionada con los animales, sus hábitats, clasificaciones, y más. La elección de SQLite ofrece una solución ligera y eficiente para gestionar los datos sin necesidad de un servidor de base de datos dedicado, simplificando la arquitectura y la implementación del sistema.

## 2.3 Tecnologías utilizadas

Para el desarrollo e implementación del sistema Zoológico Virtual, se han seleccionado tecnologías específicas que ofrecen una combinación óptima de rendimiento, facilidad de uso y soporte comunitario. Estas tecnologías abarcan tanto el desarrollo del frontend como del backend, así como la gestión de la base de datos, asegurando un sistema cohesivo y eficiente. A continuación, se detallan las tecnologías utilizadas:

- **Java JDK 21.0.2:** Se utiliza como el lenguaje de programación principal tanto para el desarrollo del frontend como del backend. Java es conocido por su portabilidad, seguridad y vasta biblioteca de clases que facilitan la creación de interfaces gráficas de usuario (GUI) robustas y sistemas backend complejos.
- **-SQLite 3.36.0:** Para la gestión de la base de datos, se emplea SQLite, una solución ligera que no requiere de un servidor de base de datos dedicado. SQLite es ideal para este proyecto debido a su facilidad de integración con Java, su eficiencia en el manejo de datos y su capacidad para funcionar en diversos sistemas operativos sin necesidad de configuraciones adicionales.

- **JavaFX:** Para el desarrollo de la interfaz gráfica de usuario (GUI), se utiliza JavaFX, que permite crear interfaces ricas e interactivas para la aplicación de escritorio. JavaFX ofrece una amplia gama de herramientas y componentes que facilitan la implementación de elementos visuales complejos y la interacción del usuario, mejorando la experiencia de usuario final.
- **Visual Studio Code:** Como entorno de desarrollo integrado (IDE), se opta por Visual Studio Code, un editor de código fuente ligero pero potente, que soporta múltiples lenguajes de programación y frameworks, incluido Java. Visual Studio Code ofrece funcionalidades como depuración, control de versiones Git integrado y una amplia selección de extensiones que agilizan el desarrollo y la prueba del software.

Estas tecnologías no solo facilitan el desarrollo y mantenimiento del Zoológico Virtual sino que también aseguran una base sólida para futuras expansiones o mejoras del sistema. La elección de Java y SQLite, complementada con JavaFX para el frontend y Visual Studio Code como IDE, refleja un enfoque en la eficiencia, la portabilidad y la facilidad de uso, aspectos cruciales para el éxito del proyecto.

## DISEÑO DEL SISTEMA

### 3.1 Arquitectura de Software

El diseño del sistema del Zoológico Virtual se estructura siguiendo un modelo de arquitectura de software en tres capas, optimizado para separar claramente las responsabilidades de cada componente, mejorar la mantenibilidad y facilitar la escalabilidad. A continuación, se detalla cada capa:

- **Capa de Presentación:** Esta capa se encarga de interactuar directamente con el usuario, presentando la interfaz gráfica de usuario (GUI) y recogiendo las entradas del usuario para ser procesadas. Desarrollada con JavaFX, proporciona una experiencia de usuario rica e interactiva, permitiendo a los usuarios navegar por la información de los animales, sus hábitats, y realizar diversas operaciones como búsquedas o actualizaciones de datos.
- **Capa de Lógica de Negocio:** En el núcleo del sistema, esta capa implementa la lógica de aplicación necesaria para realizar las operaciones definidas por los requerimientos del sistema. En ella, se procesan las solicitudes provenientes de la capa de presentación, se ejecutan las reglas de negocio, y se gestionan las transacciones con la base de datos. Esta capa está desarrollada en Java, aprovechando patrones de diseño como el Data

Access Object (DAO) para abstraer y manejar las operaciones de la base de datos de manera eficiente.

- **Capa de Acceso a Datos:** Esta capa sirve de puente entre la lógica de negocio y la base de datos. Se encarga de la comunicación directa con la base de datos SQLite, realizando consultas, actualizaciones y operaciones de gestión de datos. La utilización del patrón DAO facilita la interacción con la base de datos, permitiendo una fácil adaptación a cambios en la estructura de la base de datos o incluso la migración a otro sistema de gestión de base de datos sin impactar las capas superiores.

## 3.2 Base de Datos

El diseño de la base de datos del Zoológico Virtual está centrado en garantizar la eficiencia y la integridad de los datos. Utilizando SQLite, una solución ligera y sin servidor, la base de datos está diseñada con el siguiente esquema:

- **Tablas Relacionales:** Se crean tablas para almacenar información específica de los animales (por ejemplo, `Animales`, `Hábitats`, `Clasificaciones`), con campos detallados para describir cada aspecto relevante, como nombre, especie, hábitat, estado de conservación, y fotografías.

	IdAnimal INTEGER PRIMARY KEY AUTOINCREMENT	IdSexo INTEGER REFERENCES Sexo(IdSexo)	IdHabitat INTEGER REFERENCES Habitat(IdHabitat)	IdClasificacion INTEGER REFERENCES Clasificacion(IdClasificacion)	Nombre VARCHAR(50) NOT NULL UNIQUE	Qr VARCHAR(256) NOT NULL UNIQUE	Observacion TEXT	Estado CHAR NOT NULL DEFAULT 'A'
1	1	1 Nombre: 'MACHO' ...	1 Nombre: 'SELVA' O...	1 Nombre: 'MAMIFE...	Leon	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
2	2	1 Nombre: 'MACHO' ...	2 Nombre: 'ARBOLES...	4 Nombre: 'ANFIBIO'...	Sapo	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
3	3	2 Nombre: 'HEMBRA'...	1 Nombre: 'SELVA' O...	1 Nombre: 'MAMIFE...	Tigre	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
4	4	1 Nombre: 'MACHO' ...	2 Nombre: 'ARBOLES...	1 Nombre: 'MAMIFE...	Gorila	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
5	5	1 Nombre: 'MACHO' ...	3 Nombre: 'SEMLAC...	1 Nombre: 'MAMIFE...	Hipopotamo	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
6	6	1 Nombre: 'MACHO' ...	1 Nombre: 'SELVA' O...	2 Nombre: 'REPTIL' ...	Serpiente	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
7	7	1 Nombre: 'MACHO' ...	2 Nombre: 'ARBOLES...	3 Nombre: 'AVE' Obs...	Condor	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
8	8	1 Nombre: 'MACHO' ...	3 Nombre: 'SEMLAC...	3 Nombre: 'AVE' Obs...	PatoAmericano	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
9	9	1 Nombre: 'MACHO' ...	1 Nombre: 'SELVA' O...	1 Nombre: 'MAMIFE...	Capibara	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
10	10	2 Nombre: 'HEMBRA'...	3 Nombre: 'SEMLAC...	4 Nombre: 'ANFIBIO'...	Tortuga	<a href="https://drive.google.com...">https://drive.google.com...</a>	NULL	A
+								

Figura 1: Ejemplos tabla para almacenar información de los animales.

- **Relaciones:** Las relaciones entre tablas se establecen para reflejar la conexión lógica entre diferentes entidades, como animales y sus hábitats, permitiendo consultas eficientes y mantenimiento de la integridad referencial. Por ejemplo, una relación de clave

foránea entre las tablas `Animales` y `Hábitats` asegura que cada animal esté vinculado a un hábitat existente.

- **Índices:** Para mejorar el rendimiento de las consultas, se definen índices en las columnas más consultadas. Esto acelera la búsqueda y recuperación de datos, especialmente importante en operaciones de búsqueda y filtrado que son comunes en el sistema.

Este diseño de base de datos asegura no solo una gestión eficiente de los datos relacionados con el zoológico virtual sino también una fácil expansión para incorporar más información o funcionalidades en el futuro. La elección de SQLite como sistema de gestión de base de datos refleja un enfoque en la simplicidad, portabilidad y baja sobrecarga de mantenimiento, ideal para aplicaciones de escritorio como el Zoológico Virtual.

## REQUISITOS DEL SISTEMAS

### 4.1 Requisitos de Software

- **Sistema Operativo:** Windows 10 o superior, compatible también con macOS y Linux para asegurar una amplia accesibilidad.
- **Java Runtime Environment (JRE):** Versión 8 o superior, necesaria para ejecutar la aplicación Java.
- **SQLite:** Para gestionar la base de datos del Zoológico Virtual, se requiere SQLite 3.36.0 o superior.

### 4.2 Requisitos de hardware

- **Procesador:** Intel Core i3 o superior, AMD equivalente, para un rendimiento óptimo.
- **Memoria RAM:** Mínimo de 4GB, recomendado 8GB para una mejor experiencia de usuario.
- **Espacio en Disco:** 100MB libres para la instalación del software y almacenamiento de la base de datos.

## IMPLEMENTACIÓN

### 5.1 Desarrollo de Componentes

El desarrollo de los componentes del Zoológico Virtual se centra en la creación de una interfaz de usuario intuitiva, la lógica de negocio robusta y una base de datos eficiente, asegurando una experiencia de usuario cohesiva y funcional.

- **Frontend (Interfaz de Usuario):** Utilizando JavaFX, se desarrolla una interfaz gráfica que permite a los usuarios interactuar fácilmente con el sistema, visualizando información sobre los animales, sus hábitats y más. Se prioriza la usabilidad y el diseño atractivo para facilitar la navegación.
- **Backend (Lógica de Negocio): Implementado** en Java, este componente maneja la lógica detrás de las operaciones del sistema, como el procesamiento de solicitudes de los usuarios, la gestión de datos y la ejecución de operaciones CRUD (crear, leer, actualizar, eliminar) en la base de datos.
- **Base de Datos:** Se emplea SQLite para almacenar y gestionar la información del zoológico. La estructura de la base de datos se diseña para optimizar la eficiencia en el almacenamiento y la recuperación de datos, con tablas que representan entidades como animales, hábitats, y clasificaciones.

## 5.2 Integración de Componentes

La integración de los componentes del sistema se realiza a través de una serie de interfaces y APIs que permiten la comunicación fluida entre la interfaz de usuario, la lógica de negocio y la base de datos.

- **Comunicación Frontend-Backend:** Se establece mediante llamadas a métodos y servicios que permiten al frontend solicitar operaciones al backend, que a su vez procesa estas solicitudes, interactúa con la base de datos y devuelve los resultados al usuario.
- **Acceso a Datos:** El backend utiliza el patrón DAO (Data Access Object) para abstraer y encapsular todas las operaciones de acceso a la base de datos. Esto facilita un acceso coherente a los datos y permite cambios en la estructura de la base de datos sin afectar el resto del sistema.
- **Manejo de Sesiones y Seguridad:** Se implementan mecanismos para gestionar sesiones de usuario y asegurar la comunicación entre el cliente y el servidor, protegiendo la información sensible y asegurando que solo los usuarios autorizados puedan acceder a ciertas funcionalidades.

# OPERACIÓN Y MANTENIMIENTO

## 6.1 Procedimientos de Operación

El correcto funcionamiento del Zoológico Virtual depende de seguir procedimientos de operación establecidos para garantizar una experiencia de usuario fluida y eficiente.



1. **Inicio de la Aplicación:** Los usuarios deben iniciar la aplicación ejecutando el archivo ``App.java``. Este proceso activa el entorno del Zoológico Virtual, cargando la interfaz de usuario y estableciendo la conexión con la base de datos.
2. **Acceso a la Aplicación:** Se requiere que los usuarios accedan con credenciales proporcionadas, asegurando que solo usuarios autorizados puedan usar el sistema. Este paso es crucial para la seguridad y la integridad de los datos.
3. **Navegación y Uso:** Una vez dentro del sistema, los usuarios pueden explorar las diferentes funcionalidades ofrecidas a través del menú principal. Esto incluye la visualización de información sobre animales, la gestión de datos y otras operaciones relevantes.
4. **Soporte y Ayuda:** Se debe proporcionar documentación y soporte adecuado para asistir a los usuarios en la operación del sistema, incluyendo guías de usuario y preguntas frecuentes (FAQs).

## 6.2 Mantenimiento

El mantenimiento continuo es esencial para asegurar la operatividad, seguridad y relevancia del Zoológico Virtual a lo largo del tiempo.

- **Actualizaciones de Software:** Regularmente se deben realizar actualizaciones del software, incluyendo la aplicación y sus dependencias, para mejorar la funcionalidad, corregir errores y proteger contra vulnerabilidades de seguridad.
- **Copias de Seguridad de la Base de Datos:** Es fundamental realizar copias de seguridad periódicas de la base de datos para prevenir la pérdida de datos en caso de fallas del sistema o problemas de seguridad. Estas copias deben almacenarse en un lugar seguro y ser fácilmente accesibles para su restauración.
- **Monitoreo del Sistema:** El monitoreo continuo del sistema permite identificar y resolver problemas operativos de manera proactiva, asegurando el rendimiento óptimo y la disponibilidad del sistema.
- **Evaluación y Mejora Continua:** Se debe evaluar regularmente el sistema para identificar áreas de mejora, actualizando el software para incorporar nuevas funcionalidades, mejorar la experiencia del usuario y adaptarse a los cambios tecnológicos.

Estos procedimientos de operación y mantenimiento son cruciales para el éxito a largo plazo del Zoológico Virtual, asegurando que el sistema siga siendo funcional, seguro y relevante para sus usuarios.

# CONSIDERACIONES DE SEGURIDAD

## 7.1 Seguridad de la Aplicación

Para garantizar la seguridad en el Zoológico Virtual, se adoptan medidas clave que incluyen:

- **Autenticación y encriptación:** Se aseguran las credenciales y datos sensibles mediante encriptación y sistemas robustos de autenticación.
- **Control de acceso y gestión de sesiones:** Implementación de control de acceso basado en roles y gestión segura de sesiones para prevenir accesos no autorizados.
- **Validación de entrada:** Rigurosa validación de las entradas de usuario para proteger contra ataques comunes como inyección SQL y XSS.
- **Mantenimiento de seguridad:** Se realizan actualizaciones regulares y se aplica un monitoreo constante para identificar y mitigar vulnerabilidades de seguridad.

## REFERENCIAS

Flanagan, D. (2015). Java Security: The Definitive Guide (2nd ed.). O'Reilly Media.

Flanagan, D. (2019). Java Security: A Beginner's Guide (2nd ed.). O'Reilly Media.

Howard, M., & LeBlanc, D. (2003). Java Security: A Practical Guide for Programmers. Addison-Wesley Professional.

Mularen, P. (2019). Java Security: A Comprehensive Guide to Java's Security Architecture (2nd ed.). Apress.

Oracle. (2021). Java Cryptography Architecture (JCA) Reference Guide. Oracle Corporation.

OWASP. (2021). OWASP Java Security Cheat Sheet. OWASP Foundation.

Schildt, H. (2018). Java: The Complete Reference (11th ed.). McGraw-Hill Education.

Seacord, R. C. (2013). Java Security: Secure Coding Practices. Addison-Wesley Professional.