

# INFORME DE EVALUACIÓN DOCUMENTADA

## Portada

**Título del proyecto:** Sistema de Gestión de un Acuario (Mundo Marino)

**Nombre de la materia:** Programación II

**Docente:** Miguel Silva

**Curso y año:** 2-603 - 2DO AÑO

### Integrantes del grupo:

- Matías Mario Hayashi - DNI: 33909045
- Ailin Nicliz - DNI: 34490614
- Daiana Veliz - DNI: 36074209

## Link al repositorio de GitHub

[P2TP1\\_Grupo10\\_MundoMarino\\_Web](#)

**Nota:** El proyecto fue desarrollado en Java 21 con Spring Boot 3.5.7.

# 1. Descripción general del proyecto

## 1.1 Tema elegido

El proyecto es un **Sistema de Gestión de Acuario**, denominado "Mundo Marino". Su propósito es centralizar la gestión de los recursos biológicos (animales y especies), los espacios físicos (hábitats y zonas) y la administración de personal (empleados) dentro del acuario.

## 1.2 Objetivos del sistema

- **Gestión de Fauna:** Proveer herramientas para el registro, actualización y consulta de información detallada de cada animal (especie, hábitat asignado, cuidador).
- **Administración de Personal:** Administrar roles clave de empleados (Administrador, Cuidador, Guía) y centralizar sus datos de contacto y de ingreso.
- **Seguridad:** Implementar un control de acceso basado en roles para proteger las funcionalidades administrativas.
- **Control Logístico:** Mantener un registro ordenado de los hábitats, zonas y la ubicación

de las especies.

## 1.3 Descripción de los actores y sus roles

Se han definido tres roles de empleados principales, basados en la herencia de la entidad Empleado.java:

Actor	Clase JPA	Rol Principal y Permisos
Administrador	Administrador.java	Rol con permisos completos de CRUD sobre todos los módulos: Empleados, Animales, Especies, Hábitats y Zonas. Es el gestor del sistema.
Cuidador	Cuidador.java	Gestiona el CRUD de Animales. Es responsable de registrar nuevos animales, actualizar su estado y realizar asignaciones de hábitat.
Guía	Guia.java	Accede a la información de Fauna (Animales, Especies) en modo consulta para guiar a los visitantes. También gestiona el CRUD de los Contacto (Visitantes/Suscriptores).

## 2. Arquitectura y desarrollo

### 2.1 Descripción de las capas del sistema

El sistema utiliza la arquitectura **MVC (Modelo-Vista-Controlador)** sobre el framework Spring Boot, distribuido en cinco capas lógicas:

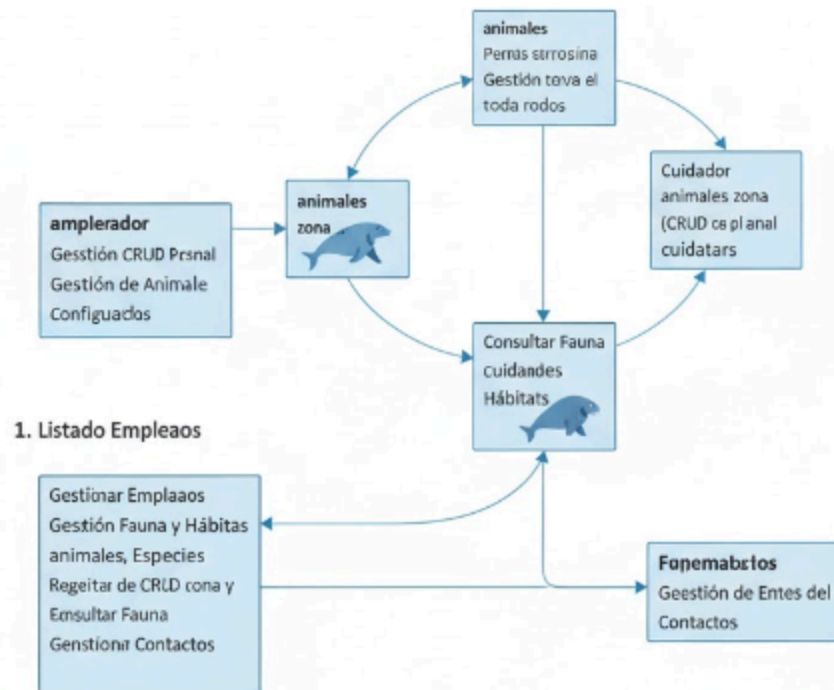
1. **Capa de Entidad (Modelo de Dominio):** Define el modelo de negocio con las clases Java mapeadas a la base de datos mediante JPA.
2. **Capa de Repositorio (Persistencia):** Interfaces para la comunicación directa con la Base de Datos.

3. **Capa de Servicio (Lógica de Negocio):** Centraliza las reglas de negocio, validación y transacciones.
4. **Capa de Controlador (Web/API):** Recibe peticiones HTTP y dirige el flujo de datos.
5. **Capa de Presentación (Vista):** Interfaz de usuario, construida con Thymeleaf y HTML.

## 2.2 Explicación de la estructura del proyecto

El proyecto sigue una organización estándar de Maven/Spring Boot:

- **Estructura de directorios y archivos:** La organización bajo src/main/resources muestra la clara separación de plantillas (/templates) y recursos estáticos (/static).



### 4. Crea Nuevo Empleaos

Tabla	id_empleado	Contingentals
empleaeas	(STRING)	
animales	Cuidador	
animales	id_emrador	
administradores	id_empleado (STRING)	
administradores	id cuidador	
cuidadoes	id_empleado (STRING)	
species	id_cuidador	
dni	id_empleado (STRING)	
guias	id_cuidador	
contactos	id_emrador	
contactos	id_empleador	

- **Configuración de la Base de Datos:** El archivo application.properties define la conexión

a la base de datos MySQL (conector J) y la configuración de Hibernate.

#### 19 Distem de Gestión Acuria del Parque Sistema de Gestión de un Acuario (Mundo Marino)

Tema es: Mundarón - TP1  
Materia: Tema: Comisión 2-603.  
Docente: Tema: Mundo Marino.  
GRUPPO 11: Tema: Mundarón 2-603.  
Alumnos: Matias Niclitz, Piroc DE  
Loreana: Niclitz, Dalana Veliz.

[Nota técnica \(requisitos\)](#)

Descripción general del proyecto

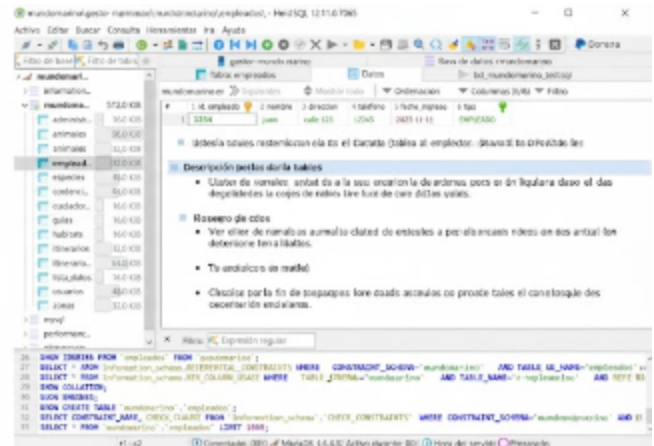
Arquitectura y desarrollo

- El sistema de gestión de Acuario a través de la interfaz de usuario.

Tecnologías e frameworks

Actores	Cases JPA	Guía	Entidad Actores	Hidrógeno del logeo
Administrador - a/c/a	Administrador	Cuidador	255 Usuarios	10.8.8.8
Empleado	Empleado	Empleado	255 Usuarios	255.0.0.0

Existe un sistema de gestión de Acuario a través de la interfaz de usuario.



## 2.3 Descripción de las tecnologías y frameworks utilizados

Categoría	Tecnología/Framework	Justificación (Según pom.xml)
Backend Core	Spring Boot 3.5.7, Java 21	Creación de aplicación web autoejecutable.
Persistencia	Spring Data JPA, Hibernate	Mapecto Objeto-Relacional y transacciones.
Conector DB	MySQL Connector/J	Conexión específica a la base de datos MySQL.
Frontend/Vistas	Thymeleaf	Motor de plantillas para generar vistas HTML dinámicas.
Seguridad	Spring Security 6	Control de autenticación y autorización (gestión de roles).
Validación	Jakarta Validation	Validación de datos con anotaciones (@NotNull, @Email) en las entidades.
Productividad	Lombok	Generación automática de métodos boilerplate.

## 2.4 Explicación de las clases principales y sus responsabilidades

Las entidades reflejan la estructura de la base de datos y la lógica de negocio:

**Empleado.java (Clase Base):** Clase base que implementa la estrategia de herencia InheritanceType.JOINED. Contiene el campo tipo que se asigna automáticamente al nombre de la subclase (Administrador, Cuidador, Guia) en el método @PrePersist.

**Animal.java:** Entidad central para el seguimiento de la fauna, con relaciones ManyToOne a Especie, Habitat y Cuidador.

**Contacto.java:** Almacenamiento de datos personales, con validaciones estrictas (@Min, @Max, @Email, @Past) para garantizar la calidad de los datos de contacto.

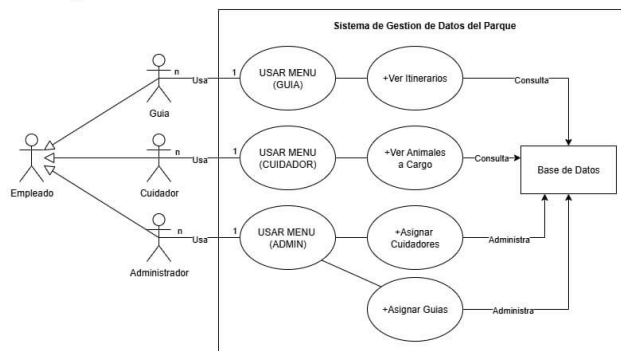
## 3. Casos de uso y diagramas

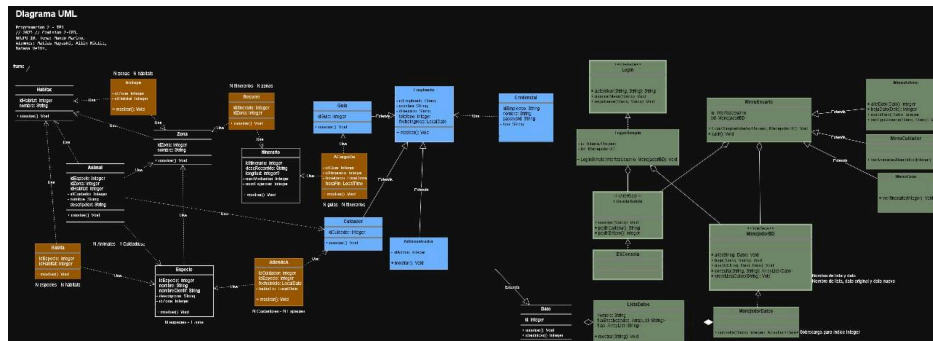
### 3.1 Diagrama de casos de uso con los actores involucrados

El siguiente diagrama visualiza las interacciones entre los actores y los módulos principales

Diagrama Caso-Uso

Programacion 2 - TP1  
// 2025 // Comision 2-603.  
GRUPO 18. Tema: Mundo Marino.  
Alumnos: Matias Hayashi,  
Ailin Niclitz, Dsiane Veliz.





## 3.2 Breve descripción de cada caso de uso

1. **Gestionar Empleados:** Tarea exclusiva del **Administrador** (CRUD sobre todos los roles).
2. **Gestionar Fauna y Hábitats:** **Cuidador** (CRUD de Animales) y **Administrador** (CRUD de Especies y Hábitats).
3. **Consultar Fauna:** Tarea de **Guía** y **Cuidador** (acceso a información de animales y especies).
4. **Gestionar Contactos:** Tarea del **Guía** o **Administrador** (CRUD para la tabla Contacto).

## 4. Base de datos

### 4.1 Diagrama entidad-relación (DER)

Este diagrama UML detalla las entidades del sistema y sus relaciones, incluyendo la herencia de Empleado y las conexiones entre Animal, Especie, Habitat, Zona, Credencial, y ListDatos.

### 4.2 Descripción de las tablas y relaciones

La base de datos se compone de las siguientes tablas principales:

Tabla	Clave Primaria (PK)	Relaciones con FK
<b>empleados</b>	id\_empleado (STRING)	Tabla padre de la herencia JOINED.
<b>animales</b>	id\_animal (INT)	FK a especies, habitats, zona y cuidadores.
<b>contactos</b>	dni (INT)	Sin relaciones externas.
<b>administradores, cuidadores, guías</b>	id\_empleado	FK a empleados (Relación 1:1 por herencia).

especies	id\_especie (INT)	FK a zona.
----------	-------------------	------------

## 4.3 Ejemplo de datos cargados

Se muestra un ejemplo de datos de la tabla empleados que representa el registro de un empleado en la base de datos MariaDB/HeidiSQL.

**Datos de la tabla empleados:**

# 5. Manual de usuario

## 5.1 Manual de uso para el Administrador

Cómo acceder al sistema:

El Administrador ingresa al sistema web con su usuario y contraseña en la pantalla de login (ej. /login).

**Funcionalidades Principales:** Gestión de Personal, Gestión de Fauna, Configuración.

**Ejemplos y Pantallas:**

1. **Listado de Empleados:** Muestra la lista de personal, permitiendo la gestión de acciones como "Nuevo empleado", "Editar" o "Borrar".
2. **Creación de Nuevo Empleado:** Formulario para registrar un nuevo usuario (ej. rol Cuidador), con los campos de datos personales (ID, Nombre, Dirección, Teléfono, Fecha Ingreso, Tipo).

## 5.2 Manual de uso para el Cuidador

Cómo acceder al sistema:

Ingresa al sistema web con su usuario y contraseña (ej. /login).

**Funcionalidades Principales:** Gestión de Animales a su cargo, Registro de Nuevos Animales.

Ejemplo de Pantalla:

El Cuidador accede a la interfaz de gestión de animales.

## 5.3 Manual de uso para el Guía

Cómo acceder al sistema:

Ingresa al sistema web con su usuario y contraseña (ej. /login).

**Funcionalidades Principales:** Consulta de Información de Fauna, Gestión de Contactos (Visitantes).

Ejemplo de Pantalla:

El Guía utiliza una interfaz para consultar los datos de fauna o gestionar los registros de

contacto.