

# Sistema de Gestión de Equipos

## Documentación Técnica y Manual de Usuario

Ing. Jorge Hernández López

# Índice

<b>INTRODUCCIÓN.....</b>	<b>3</b>
1.1    Propósito del Sistema.....	3
1.2 Alcance.....	3
<b>2. Arquitectura del Sistema .....</b>	<b>3</b>
2.1 Tecnologías Utilizadas.....	3
2.2 Patrones de Diseño .....	4
<b>Arquitectura MVC del Sistema.....</b>	<b>4</b>
<b>3. Configuración e Instalación .....</b>	<b>4</b>
3.1 Requisitos del Sistema.....	4
3.2 Proceso de Instalación Detallado.....	4
<b>4. Estructura del Proyecto .....</b>	<b>5</b>
4.1 Directorios Principales .....	5
4.2 Descripción de Componentes Principales .....	6
4.2.1 Controladores.....	6
4.2.2 Modelos.....	6
<b>5. Base de Datos .....</b>	<b>6</b>
5.1 Diagrama Entidad-Relación .....	6
5.2 Migraciones.....	7
<b>6. API y Rutas .....</b>	<b>8</b>
6.1 Rutas del Sistema.....	8
6.2 Documentación de Endpoints .....	8
6.2.1 Equipos API.....	8
6.2.2 Departamentos API .....	8
<b>7. Módulos del Sistema.....</b>	<b>9</b>
7.1 Gestión de Equipos .....	9
7.1.1 Funcionalidades.....	9
7.1.2 Flujo de Trabajo .....	9
7.2 Gestión de Departamentos .....	9
7.2.1 Funcionalidades.....	9

## INTRODUCCIÓN

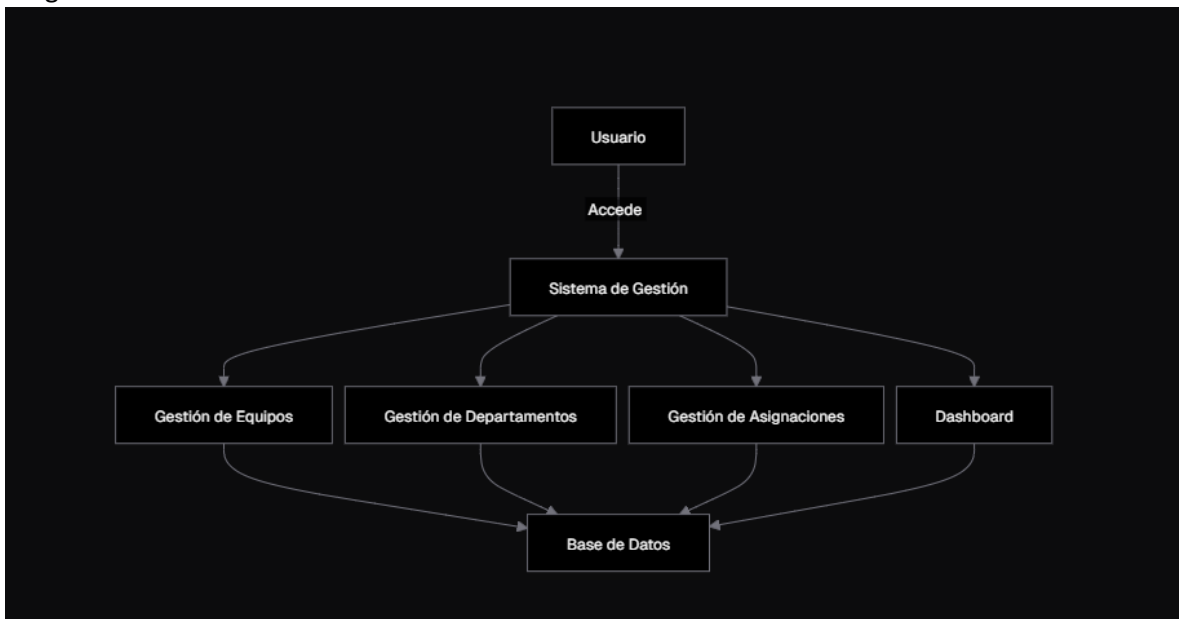
### 1.1 Propósito del Sistema

El Sistema de Gestión de Equipos es una aplicación web desarrollada con Laravel que permite administrar el inventario de equipos tecnológicos, su asignación a departamentos y el seguimiento de su estado.

### 1.2 Alcance

- Gestión completa de equipos tecnológicos
- Administración de departamentos
- Control de asignaciones y devoluciones
- Dashboard interactivo

Diagrama General del Sistema



## 2. Arquitectura del Sistema

### 2.1 Tecnologías Utilizadas

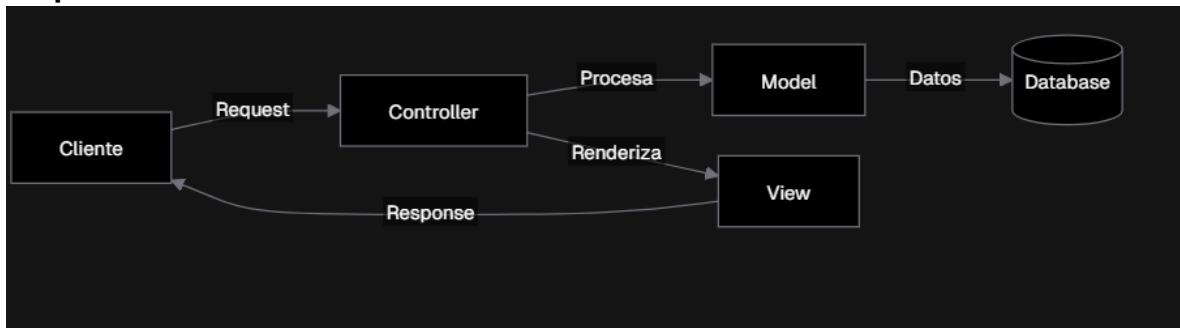
- Backend: Laravel 11.x
- Frontend: Blade Templates, Bootstrap 5
- Base de Datos: MySQL 8.0

- Servidor: Apache

## 2.2 Patrones de Diseño

- MVC (Modelo-Vista-Controlador)
- Repository Pattern
- Service Layer
- Factory Pattern

### Arquitectura MVC del Sistema



## 3. Configuración e Instalación

### 3.1 Requisitos del Sistema

- PHP >= 8.2
- Composer
- MySQL >= 8.0
- Node.js >= 18.x
- NPM >= 9.x

### 3.2 Proceso de Instalación Detallado

#### 1. clonar el Repositorio

```
git clone [url-del-repositorio]
cd sistema-gestion-equipos
```

#### 2. Instalar Dependencias

```
composer install
npm install
```

#### 3. Configuración del Entorno

```
cp .env.example .env
php artisan key:generate
```

4. Configuración de la **Base de Datos** Editar el archivo .env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=sistema_gestion_equipos
DB_USERNAME=tu_usuario
DB_PASSWORD=tu_contraseña
```

5. Migraciones y Seeders

```
php artisan migrate --seed
```

## 4. Estructura del Proyecto

### 4.1 Directorios Principales

```
sistema-gestion-equipos/
├── app/
│   ├── Http/
│   │   ├── Controllers/
│   │   └── Middleware/
│   ├── Models/
│   └── Services/
├── database/
│   ├── migrations/
│   └── seeders/
├── resources/
│   └── views/
└── routes/
    └── web.php
```

## 4.2 Descripción de Componentes Principales

### 4.2.1 Controladores

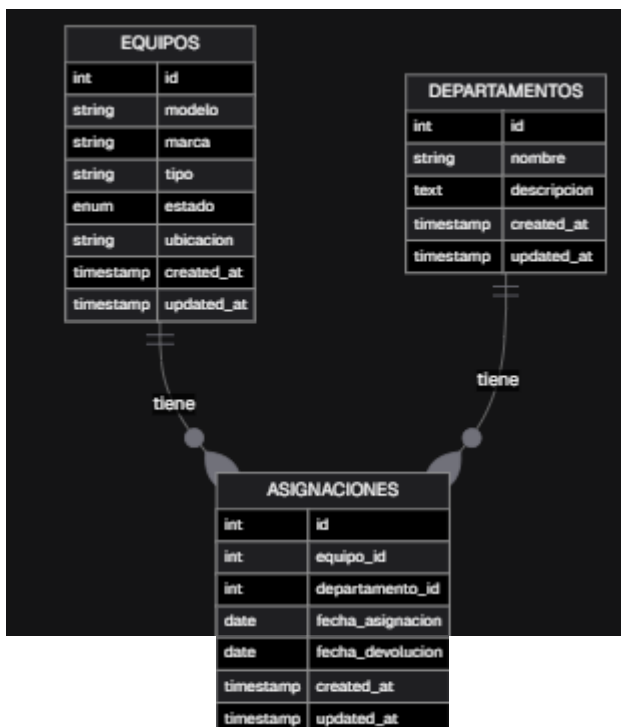
```
app/Http/Controllers/  
├─ EquipoController.php  
├─ DepartamentoController.php  
├─ AsignacionController.php  
└─ DashboardController.php
```

### 4.2.2 Modelos

```
app/Models/  
├─ Equipo.php  
├─ Departamento.php  
└─ Asignacion.php
```

## 5. Base de Datos

### 5.1 Diagrama Entidad-Relación



## 5.2 Migraciones

Detalle de las migraciones principales:

```
// create_equipos_table.php
Schema::create('equipos', function (Blueprint $table) {
    $table->id();
    $table->string('modelo');
    $table->string('marca');
    $table->string('tipo');
    $table->enum('estado', ['disponible', 'asignado']);
    $table->string('ubicacion');
    $table->timestamps();
});

// create_departamentos_table.php
Schema::create('departamentos', function (Blueprint $table) {
    $table->id();
    $table->string('nombre');
    $table->text('descripcion')->nullable();
    $table->timestamps();
});

// create_asignaciones_table.php
Schema::create('asignaciones', function (Blueprint $table) {
    $table->id();
    $table->foreignId('equipo_id')->constrained();
    $table->foreignId('departamento_id')->constrained();
    $table->date('fecha_asignacion');
    $table->date('fecha_devolucion')->nullable();
    $table->timestamps();
});
```

## 6. API y Rutas

### 6.1 Rutas del Sistema

```
// routes/web.php
Route::get('/', [DashboardController::class, 'index'])->name('dashboard');

// Rutas de Equipos
Route::resource('equipos', EquipoController::class);

// Rutas de Departamentos
Route::resource('departamentos', DepartamentoController::class);

// Rutas de Asignaciones
Route::resource('asignaciones', AsignacionController::class);
```

### 6.2 Documentación de Endpoints

#### 6.2.1 Equipos API

Método	URI	Descripción	Parámetros
GET	/equipos	Lista todos los equipos	?page, ?search
POST	/equipos	Crea un nuevo equipo	modelo, marca, tipo, estado
GET	/equipos/id	Muestra un equipo	-
PUT	/equipos/id	Actualiza un equipo	modelo, marca, tipo, estado
DELETE	/equipos/id	Elimina un equipo	-

#### 6.2.2 Departamentos API

Método	URI	Descripción	Parámetros
GET	/departamentos	Lista departamentos	?page
POST	/departamentos	Crea departamento	Nombre, descripción
GET	/departamentos/id	Muestra departamento	-
PUT	/departamentos/id	Actualiza departamentos	Nombre, descripción
DELETE	/departamentos/id	Elimina departamento	-



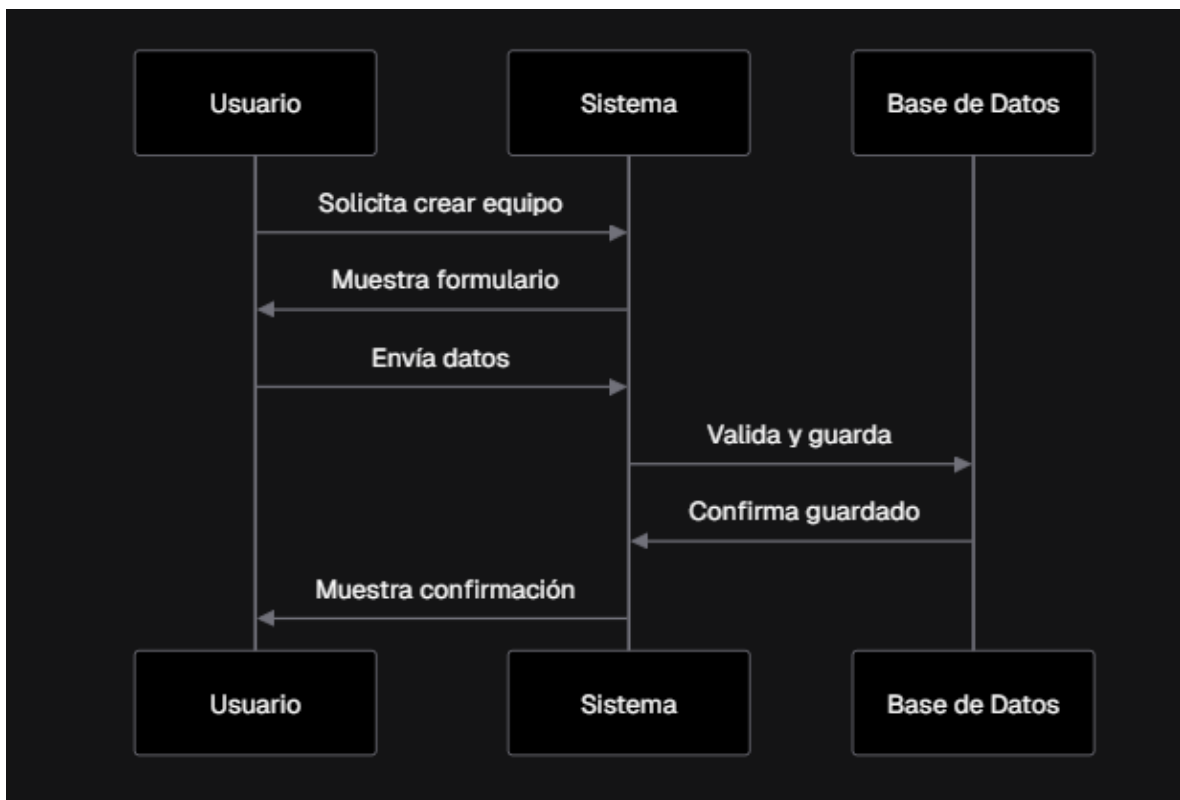
## 7. Módulos del Sistema

### 7.1 Gestión de Equipos

#### 7.1.1 Funcionalidades

- Registro de equipos
- Actualización de información
- Eliminación de equipos
- Búsqueda y filtrado

#### 7.1.2 Flujo de Trabajo



### 7.2 Gestión de Departamentos

#### 7.2.1 Funcionalidades

- Creación de departamentos
- Asignación de responsables
- Gestión de ubicaciones
- Historial de asignaciones

## 8. Flujos de Trabajo

## 8.1 Proceso de Asignación de Equipos

