

Informe de Analisis de sistema para Salida a Producción del proyecto “Reserva tu Parque” v1

Tabla de contenido

1. DESCRIPCIÓN DEL PROYECTO.....	3
1.1 Propósito y Alcance.....	3
Objetivo Principal.....	3
Alcance del Sistema.....	3
1.2 Contexto del Proyecto.....	4
Problema que Resuelve.....	4
1.3. Stack Tecnológico	4
Backend	4
Frontend.....	4
Desarrollo y Testing	5
Infraestructura	5
1.4 Arquitectura del Sistema.....	5
Patrón de Diseño	5
Estructura de Directorios	5
1.5 Modelos de Datos Principales (Tablas maestras)	6
1.5.1. Entidades Core	6
Entidades de Soporte (Tablas de detalle).....	7
1.5.2. Diagrama entidad relación	8
1.6 Funcionalidades Implementadas.....	9
Módulo Público (Sin Autenticación)	9
Módulo Administrativo (Requiere Autenticación).....	9
Módulo de Notificaciones	10
API y Servicios	10
1.7 Despliegue e Infraestructura	11
Opciones de Despliegue Documentadas.....	11
Requisitos de Servidor.....	11
2. FLUJOS DEL SISTEMA	12
2.1 Flujo de Reservación Ciudadana.....	12
Descripción General	12
Actores	12
Diagrama de Flujo Principal	12
2.2 Flujo de Gestión Administrativa	15
Descripción General	15
Actores	15
Diagrama de Flujo	15
3 ERRORES E INCONSISTENCIAS ENCONTRADAS	19
3.1. Problemas al presentar el QR:.....	19
3.2. Cuando se intenta ver los eventos:.....	19
3.4. Detalles de reservaciones incompleto	20
4 Conclusión.....	21

1. DESCRIPCIÓN DEL PROYECTO

Información General

Nombre del Proyecto: ADN Reserva tu Parque

Versión Laravel: 10.0

Versión PHP: ^8.1

Base de Datos: MySQL (adn_reservaciones_parques)

Repositorio: Git

Estado: En desarrollo / Pre-producción

1.1 Propósito y Alcance

Objetivo Principal

Sistema web de gestión y reservación de espacios en parques públicos institucionales, permitiendo a ciudadanos solicitar el uso de zonas específicas para eventos y actividades, mientras facilita a los administradores la gestión centralizada de todas las solicitudes.

Alcance del Sistema

Usuarios objetivo:

- **Ciudadanos/Solicitantes:** Personas que desean reservar espacios en parques públicos
- **Operadores de Parque:** Personal encargado de gestionar reservas de un parque específico
- **Administradores:** Personal con acceso total al sistema para supervisión y gestión global

Funcionalidades principales:

1. Sistema de reservaciones públicas en línea
2. Gestión administrativa de solicitudes
3. Notificaciones automáticas por correo electrónico
4. Generación de códigos QR para confirmación de reservas
5. Panel de control con estadísticas
6. Gestión de múltiples parques y zonas
7. Sistema de roles y permisos

1.2 Contexto del Proyecto

Problema que Resuelve

Antes del sistema:

- Proceso manual y presencial para reservar espacios en parques
- Falta de transparencia en el proceso de aprobación
- Imposibilidad de verificar disponibilidad en tiempo real
- Documentación en papel difícil de gestionar
- Sin trazabilidad de solicitudes

Con el sistema:

- Proceso 100% digital y accesible 24/7
- Transparencia total: el solicitante puede consultar el estado de su reserva
- Validación automática de disponibilidad
- Base de datos centralizada
- Trazabilidad completa con fechas de creación y modificación
- Notificaciones automáticas
- Códigos QR para validación en sitio

1.3. Stack Tecnológico

Backend

- **Framework:** Laravel 10.0
- **Lenguaje:** PHP 8.1+
- **Base de Datos:** MySQL
- **ORM:** Eloquent
- **Autenticación API:** Laravel Sanctum 3.2
- **HTTP Client:** GuzzleHTTP 7.2 (integraciones externas)
- **Mail:** SMTP vía Office 365

Frontend

- **Templating:** Blade (Laravel)

- **Build Tool:** Vite 4.0
- **HTTP Client:** Axios 1.1.2
- **Generación QR:** qr-code-styling 1.6.0-rc.1

Desarrollo y Testing

- **Testing:** PHPUnit 10.1
- **Mocking:** Mockery 1.4.4
- **Code Style:** Laravel Pint 1.0
- **Error Handling:** Spatie Ignition 2.0
- **Fake Data:** FakerPHP 1.9.1

Infraestructura

- **Containerización:** Docker
- **Servidores soportados:** Apache, Nginx, XAMPP
- **Despliegue:** Compatible con cPanel, Docker Compose

1.4 Arquitectura del Sistema

Patrón de Diseño

- **MVC (Model-View-Controller):** Parcialmente implementado
 - **Models:** Eloquent ORM con 15 modelos principales
 - **Views:** Blade templates
 - **Controllers:** Lógica actualmente en closures de rutas (web.php)

Estructura de Directorios

```

adn_reserva_tu_parque/
├── app/
│   ├── Http/
│   │   └── Controllers/      # Controlador base (lógica en
│   │   └── Middleware/      # TokenAuthentication personalizado
│   │   └── Models/          # 15 modelos Eloquent
│   │   └── Mail/            # Clases de envío de correos
│   └── config/              # Configuraciones (app, database, mail,
etc.)
├── database/
│   ├── migrations/          # 5 migraciones principales
│   ├── seeders/             # Seeders para datos iniciales
│   └── rebuild/             # Scripts SQL para recrear estructura
├── resources/
│   └── views/                # Plantillas Blade (login, reservaciones,
panel admin)
├── lang/                    # Archivos de idioma
└── routes/

```

└─ web.php	# 1300+ líneas con todas las rutas y lógica
└─ public/	# Assets públicos (CSS, JS, imágenes)
└─ storage/	# Logs, cache, archivos generados
└─ tests/	# Tests unitarios y de integración

1.5 Modelos de Datos Principales (Tablas maestras)

1.5.1. Entidades Core

1. *tbl_reservaciones (Reservaciones)*

Tabla central del sistema que almacena todas las solicitudes de reserva.

Campos principales:

- `id_solicitante` - Relación con solicitante
- `id_parque` - Parque seleccionado
- `id_zona` - Zona específica del parque
- `id_evento` - Tipo de evento
- `fecha_evento`, `hora_inicio`, `hora_fin` - Temporalidad
- `motivo_evento`, `descripcion` - Detalles del evento
- `responsables` - Personas responsables
- `cantidad_participantes_adultos`, `cantidad_participantes_niños` - Capacidad
- `codigo_reservacion` - Código único para QR
- `estado` - espera | pendiente | confirmada | rechazada | vencida | realizada

2. *tbl_solicitantes (Solicitantes)*

Usuarios que realizan solicitudes de reserva (no autenticados).

Campos:

- `nombres`, `apellidos`
- `id_tipo_doc`, `documento` - Identificación
- `celular`, `correo` - Contacto

3. *tbl_usuario (Usuarios Administrativos)*

Personal del sistema (operadores y administradores).

Campos:

- `nombre`, `apellido`, `usuario`, `password`
- `tipo_documento`, `documento`
- `correo`, `telefono`
- `id_parque` - Parque asignado (NULL para administradores)

- `id_user_type` - Rol/permiso
- `estado` - activo | inactivo

4. ***tbl_parques (Parques)***

Información de parques disponibles para reserva.

Campos:

- `nombre_parque`, `descripcion`
- `correo` - Email de contacto
- `provincia`, `municipio`, `sector`, `circunscripcion` - Ubicación
- `coordenadas_maps` - Geolocalización
- `direccion`
- `espera` - Modo cola de espera (boolean)
- `estado` - activo | inactivo

5. ***tbl_zona_parques (Zonas)***

Áreas específicas dentro de cada parque.

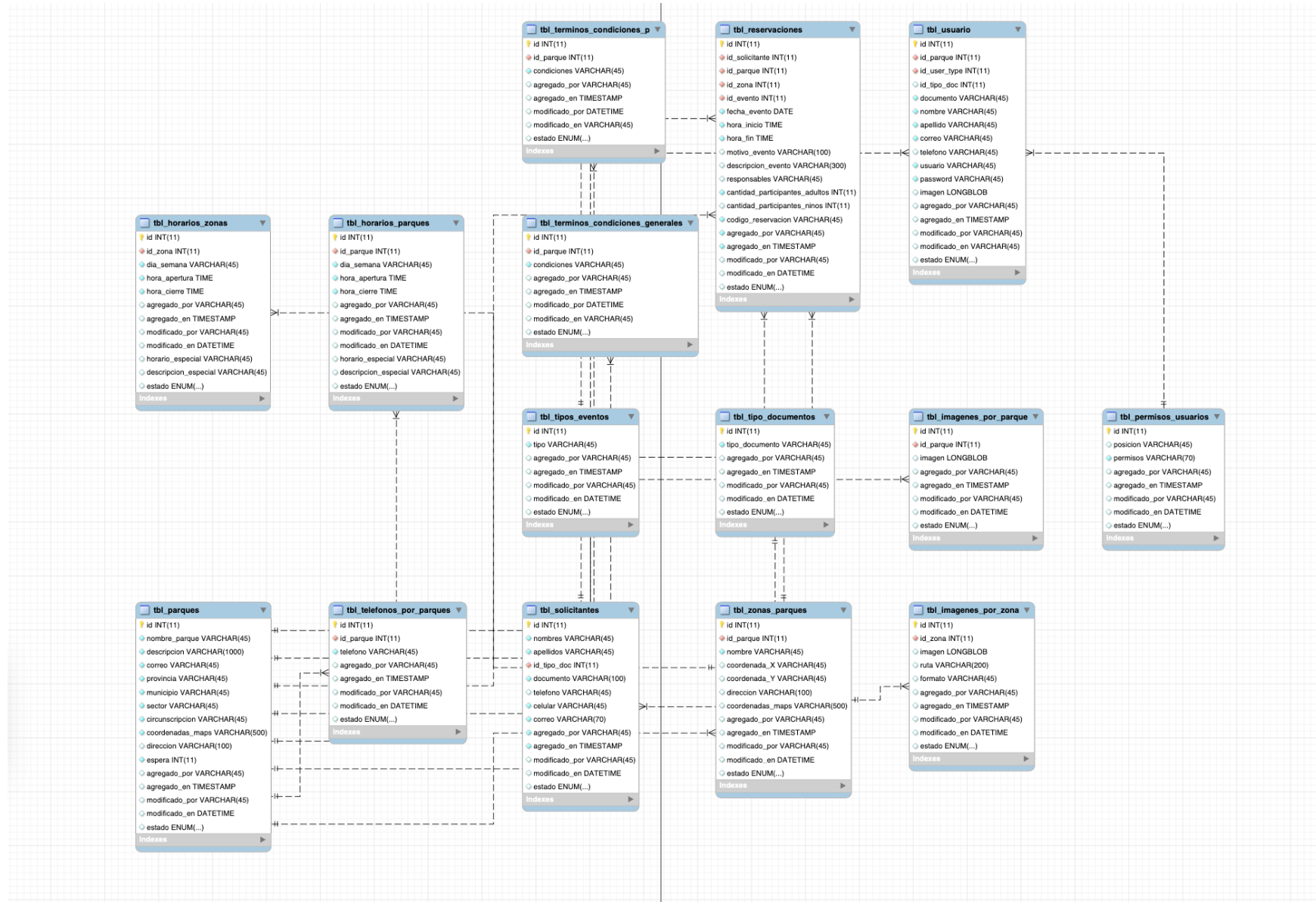
Campos:

- `nombre` - Nombre de la zona
- `direccion` - Ubicación específica
- `coordenadas_maps`
- `id_parque` - Relación con parque

Entidades de Soporte (Tablas de detalle)

- **`tbl_tipos_eventos`** - Catálogo de eventos permitidos
- **`tbl_horarios_parques`** - Horarios de apertura/cierre por parque
- **`tbl_horarios_zonas`** - Horarios específicos por zona y día
- **`tbl_imagenes_por_parque`** - Galería de imágenes de parques
- **`tbl_imagenes_por_zona`** - Galería de imágenes de zonas
- **`tbl_telefonos_por_parque`** - Múltiples teléfonos de contacto
- **`tbl_tipo_documento`** - Catálogo de tipos de identificación
- **`tbl_permisos_usuarios`** - Roles y permisos del sistema
- **`tbl_terminos_condiciones_generales`** - T&C por parque

1.5.2. Diagrama entidad relación



1.6 Funcionalidades Implementadas

Módulo Público (Sin Autenticación)

1. Sistema de Reservas

- **Ruta:** /reservaciones
- **Funcionalidad:**
 - Formulario de solicitud de reserva
 - Selección de parque → zona → tipo de evento
 - Validación de disponibilidad en tiempo real
 - Carga de imágenes de parques y zonas
 - Visualización de horarios disponibles
 - Registro de datos del solicitante
 - Generación de código de reservación único

2. Consulta de Estado de Reserva

- **Rutas:** /solicitud_estado_json, /Reservaciones_email/{email}
- **Funcionalidad:**
 - Búsqueda por código de reservación o email
 - Visualización del estado actual
 - Información completa de la reserva

3. Confirmación de Reserva

- **Ruta:** /actualizacion
- **Funcionalidad:**
 - Enlace de confirmación enviado por email
 - Actualización automática de estado

Módulo Administrativo (Requiere Autenticación)

4. Sistema de Autenticación

- **Rutas:** /login, /Reservaciones_cuenta, /return
- **Funcionalidad:**
 - Login con usuario y contraseña
 - Validación de credenciales
 - Generación de token de sesión
 - Control de permisos por rol
 - Logout con limpieza de sesión

5. Dashboard

- **Rutas:** /inicio, /Dashboard
- **Funcionalidad:**
 - Estadísticas mensuales de reservaciones
 - Vista discriminada por rol:
 - **Administrador:** Todas las reservas del sistema
 - **Operador:** Solo reservas de su parque asignado
 - Gráficos y métricas de uso

6. Gestión de Solicitudes

- **Rutas:** /R_solicitudes, /R_solicitudes_all, /Office
- **Funcionalidad:**
 - Lista de solicitudes pendientes
 - Filtrado por parque (operadores)
 - Vista completa (administradores)
 - Detalles de cada reserva
 - Aprobación/rechazo de solicitudes
 - Actualización de estado (PATCH /Office_patch/{id})

Módulo de Notificaciones

7. Sistema de Correos Electrónicos

- **Rutas:** /enviar-correo, /enviar-correo_confirmacion
- **Funcionalidad:**
 - Correo de confirmación de solicitud
 - Correo de aprobación/rechazo
 - Plantillas HTML profesionales
 - Código QR incrustado en email
 - Enlace de confirmación
 - Envío vía SMTP Office 365 (alerts@adn.gob.do)

8. Generación de Códigos QR

- **Librería:** qr-code-styling
- **Funcionalidad:**
 - QR único por reservación
 - Contiene código de reservación
 - Incrustado en correos de confirmación
 - Utilizable para validación en sitio

API y Servicios

9. Endpoints de Consulta

- `/Reservaciones_solicitantes/{id_tipo_doc}/{documento}` - Buscar solicitante
- `/Reservaciones_dia/{fecha}/{parque}` - Reservas del día
- `/Reservaciones_doc_where/{fecha}/{id_solicitante}` - Reservas por solicitante
- `/Reservaciones_imagenes_zonas/{id_zona}` - Imágenes de zona
- `/Reservaciones_parque_horarios/{id_parque}` - Horarios de parque
- `/reservaciones_eventos/{id}` - Eventos activos por parque
- `/api/user` - Usuario autenticado (Sanctum)

10. Integración con Servicios Externos

- **GuzzleHTTP:** Cliente HTTP para llamadas a APIs externas
- Validación de documentos de identidad
- Posible integración con servicios gubernamentales

1.7 Despliegue e Infraestructura

Opciones de Despliegue Documentadas

El proyecto incluye `DEPLOYMENT.md` con guías para:

1. **Docker** (Recomendado para desarrollo)
 - Dockerfile incluido
 - Compatible con Docker Compose
 - Aislamiento completo del entorno
2. **XAMPP** (Desarrollo local Windows/Mac/Linux)
 - Apache + MySQL + PHP
 - Configuración de VirtualHost
3. **Nginx** (Producción Linux)
 - Configuración de server block
 - Proxy pass para PHP-FPM
4. **cPanel** (Hosting compartido)
 - Configuración de dominios
 - Import de base de datos
 - Variables de entorno

Requisitos de Servidor

Mínimos:

- PHP 8.1+
- MySQL 5.7+ / MariaDB 10.3+
- Composer 2.x

- Node.js 16+ (para build de assets)
- 512MB RAM mínimo
- 100MB espacio en disco

2. FLUJOS DEL SISTEMA

2.1 Flujo de Reservación Ciudadana

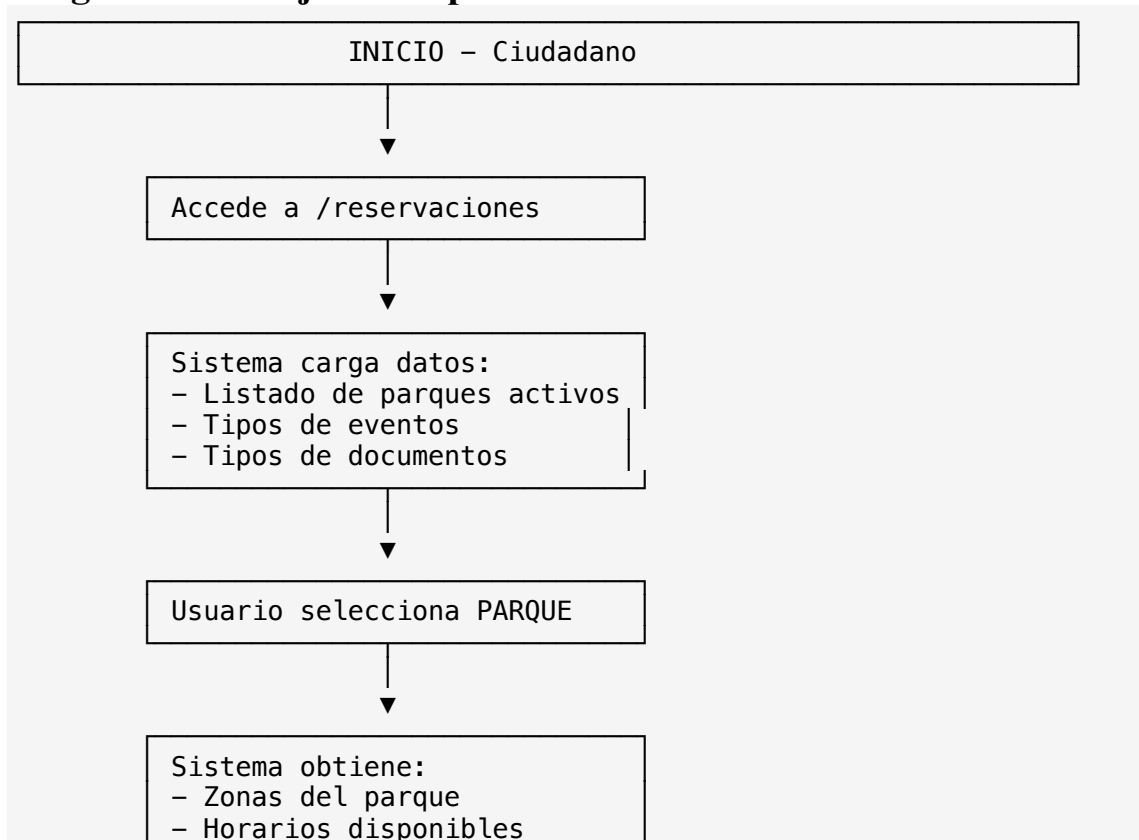
Descripción General

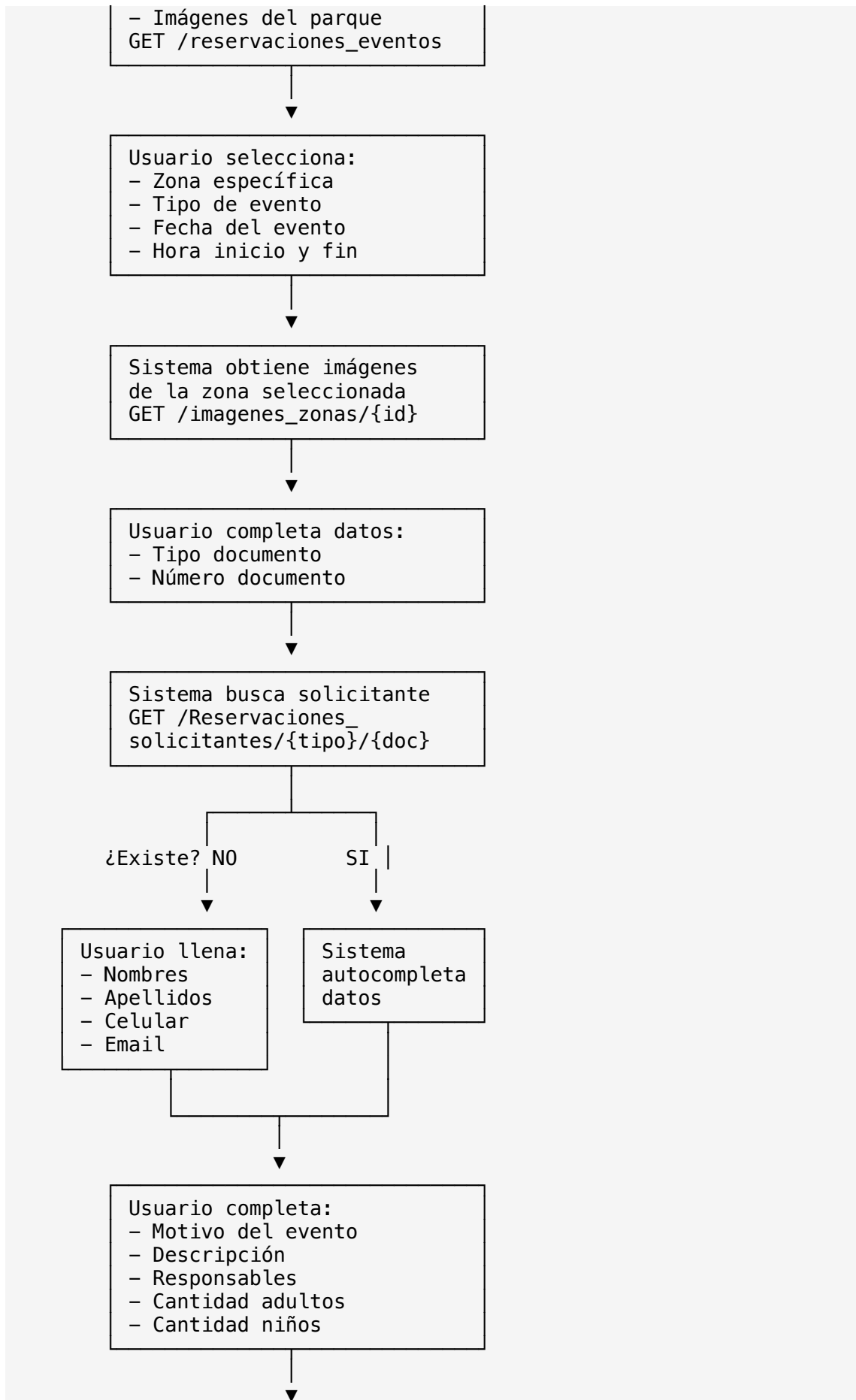
Proceso completo desde que un ciudadano accede al sistema hasta que recibe confirmación de su solicitud de reserva.

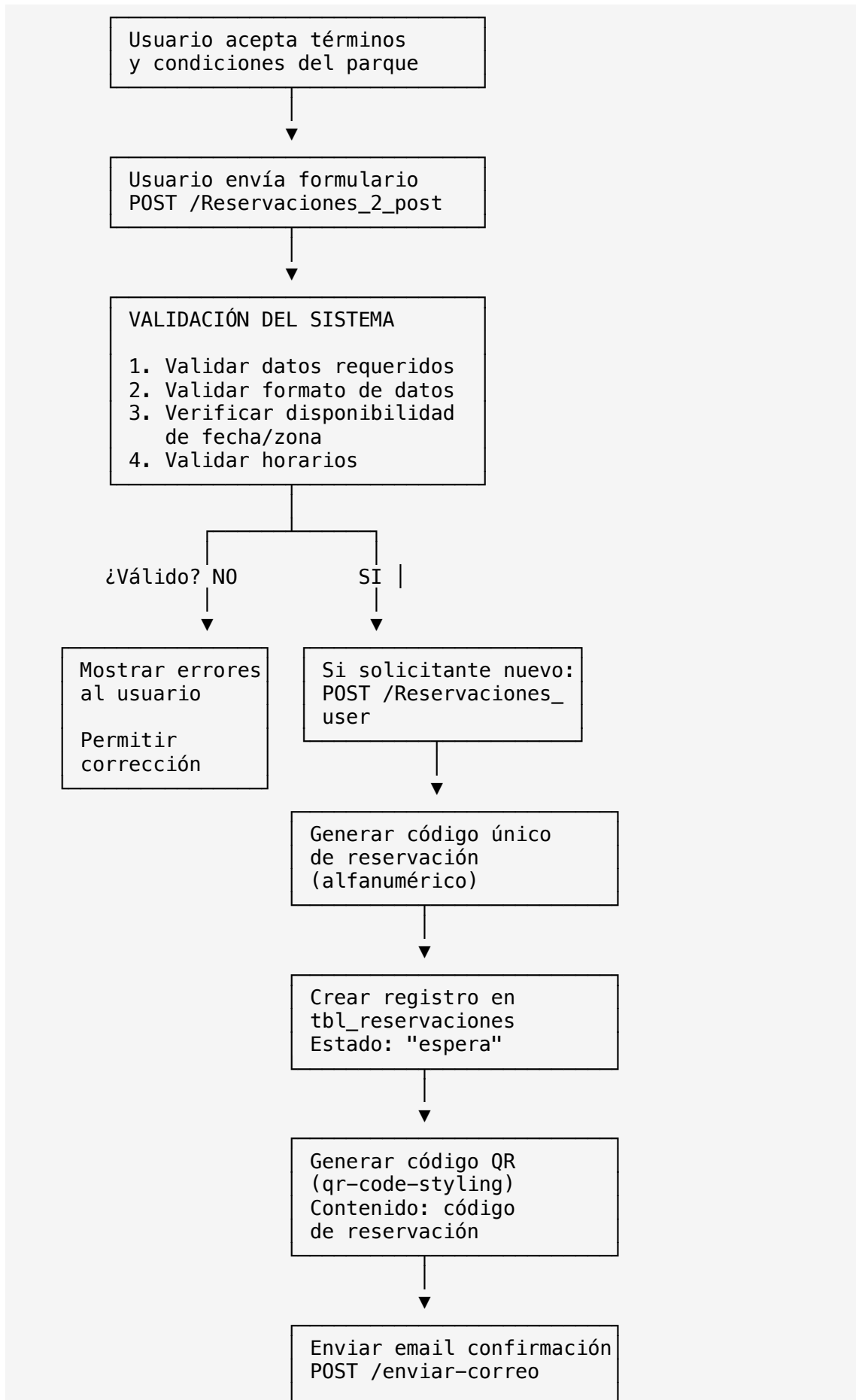
Actores

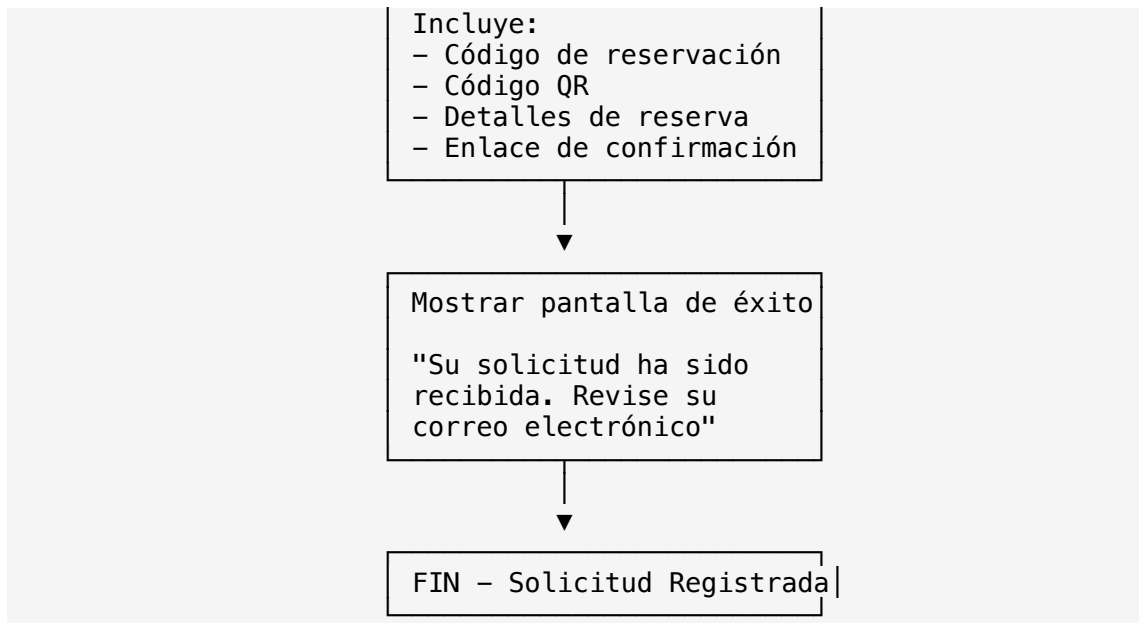
- **Ciudadano/Solicitante:** Usuario público sin autenticación
- **Sistema:** Aplicación Laravel
- **Base de Datos:** MySQL
- **Servicio de Email:** SMTP Office 365
-

Diagrama de Flujo Principal









2.2 Flujo de Gestión Administrativa

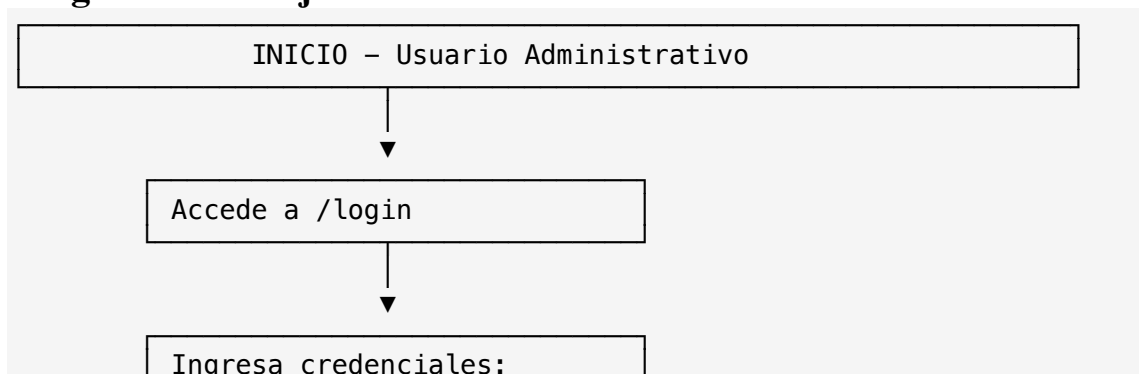
Descripción General

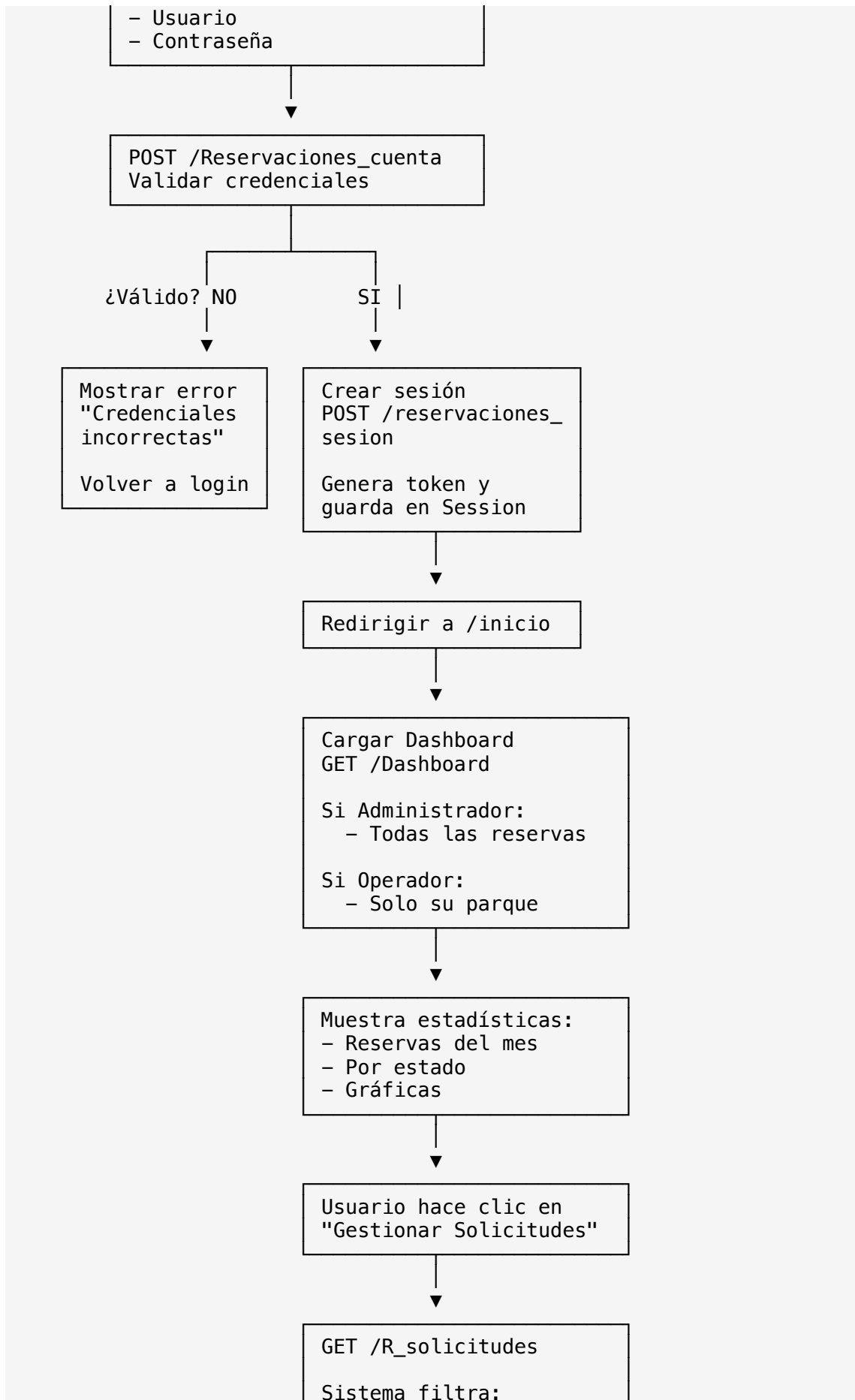
Proceso que sigue un administrador u operador de parque para gestionar las solicitudes de reserva recibidas.

Actores

- **Operador de Parque:** Usuario con permisos limitados a su parque
- **Administrador:** Usuario con permisos globales
- **Sistema:** Aplicación Laravel

Diagrama de Flujo





- Si Admin: todas
- Si Operador: su parque

WHERE estado IN
('espera', 'pendiente')

Muestra tabla con:
- Código reservación
- Solicitante
- Parque
- Zona
- Fecha
- Hora
- Estado
- Acciones

Usuario selecciona
una solicitud

GET /Office_Get_id/{id}

Obtiene detalles
completos de reserva

Muestra modal/vista con:
- Datos del solicitante
- Detalles del evento
- Motivo y descripción
- Responsables
- Cantidad participantes
- Fecha de solicitud

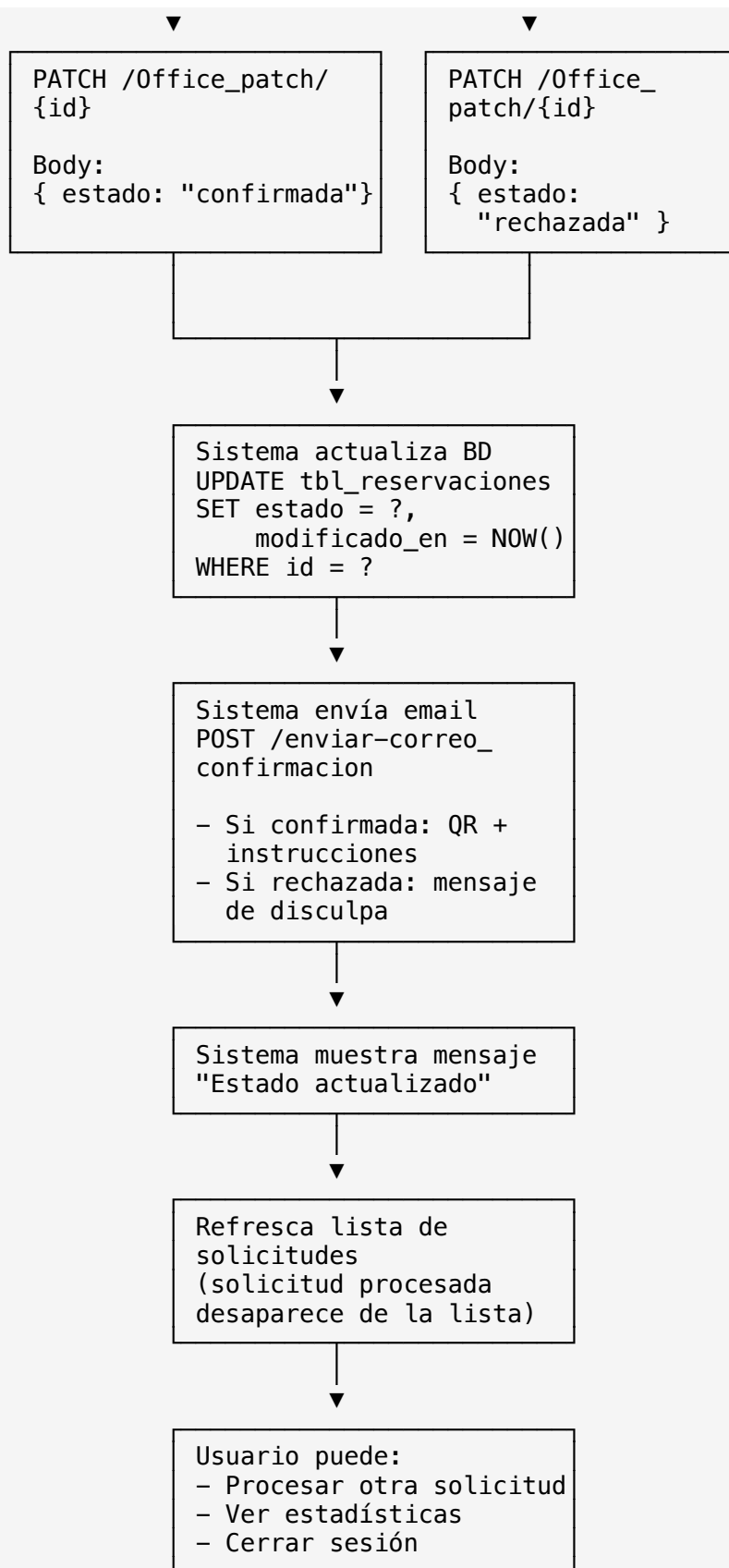
Usuario revisa y decide

APROBAR

RECHAZAR

Selecciona "Confirmar"

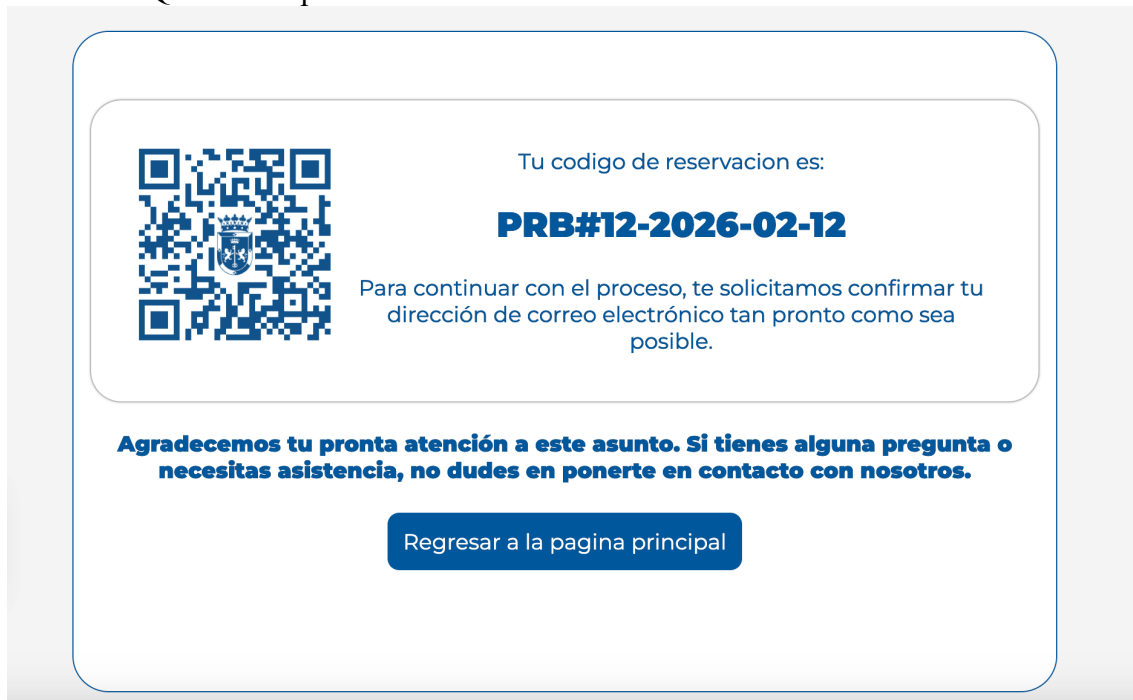
Selecciona
"Rechazar"



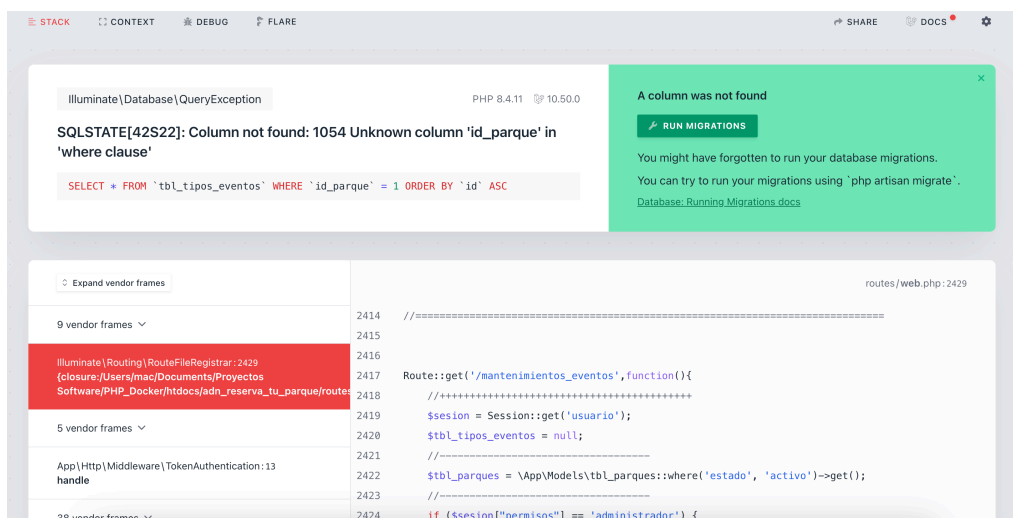
3 ERRORES E INCONSISTENCIAS ENCONTRADAS

3.1. Problemas al presentar el QR:

De manera inconsistente el QR aparece o desaparece, se debe evaluar la razón por la cual dicho QR no siempre se muestra:

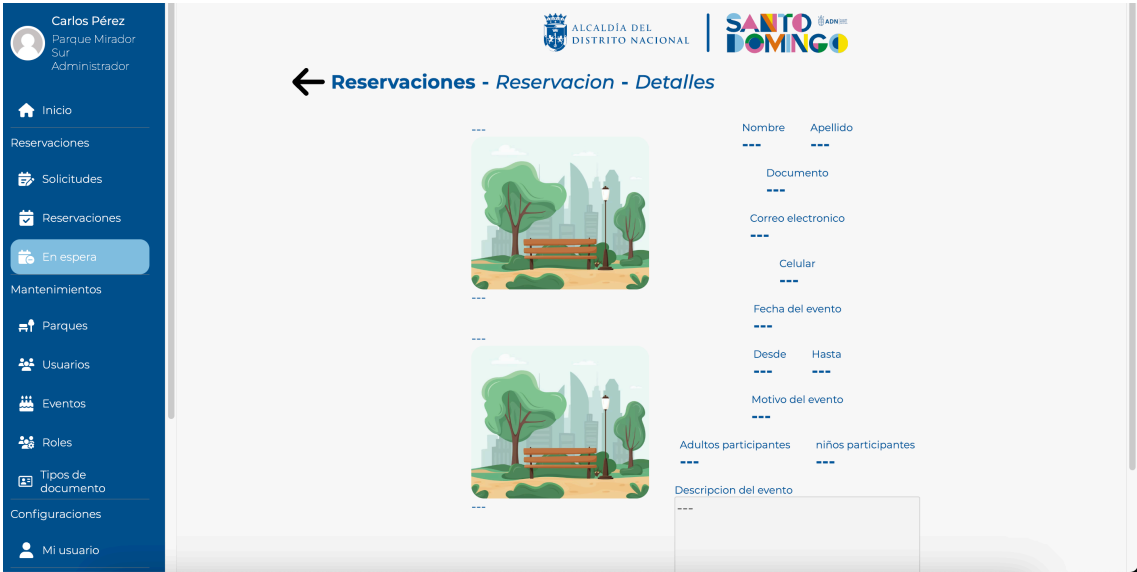


3.2. Cuando se intenta ver los eventos:



3.4. Detalles de reservaciones incompleto

La sección de detalle de reservación no muestra la info desde la BD.



El error posiblemente viene porque este endpoint no está controlado:http://127.0.0.1:8000/Reservasiones_preciso/2

4 Conclusión

ADN Reserva tu Parque es un sistema web funcional desarrollado en Laravel 10, orientado a digitalizar el proceso de reservación de espacios en parques públicos. La plataforma cubre el flujo completo, desde la solicitud realizada por el ciudadano hasta la aprobación administrativa, incorporando notificaciones automáticas y validación mediante códigos QR.

Si bien el sistema cumple con gran parte de los requerimientos necesarios para atender los casos de uso principales, se identificaron errores relevantes que podrían provocar que ciertos procesos no se ejecuten de manera correcta o que la experiencia operativa se vea afectada. Estos puntos deben ser atendidos antes de considerar el sistema completamente preparado para un entorno de producción.

Adicionalmente, aunque la interfaz de usuario resulta intuitiva y funcional, una actualización visual orientada a una estética más moderna, amigable y atractiva aportaría un valor significativo, mejorando la percepción del producto y la experiencia

