

## Documentación

Autor:

Juan Manuel Perez Reyes

Brayan Alexi Parra

José David Sánchez Osorio

Juliana Osorio Almeciga

INTEP

Unidad de Sistemas y Electricidad

Tecnología en Gestión de Sistemas Informáticos

Calculo integral

Roldanillo, Valle

## **Tabla de Contenido**

1. Introducción y Descripción General
2. Arquitectura del Sistema
3. Requisitos Técnicos
4. Estructura de Archivos
5. Base de Datos
6. Documentación Técnica Detallada
  - 6.1. Sistema de Navegación y Header
  - 6.2. Página Principal (index.php)
  - 6.3. Sistema de Agendamiento
  - 6.4. Procesamiento de Citas
  - 6.5. Confirmación de Agendamiento
  - 6.6. Página "Sobre Mí"
  - 6.7. Sistema de Estilos
  - 6.8. Gestión de Base de Datos
7. Justificación de Decisiones de Diseño
8. Flujos de Trabajo
9. Consideraciones de Seguridad
10. Mantenimiento y Escalabilidad

## 1. Introducción y Descripción General

### 1.1 Propósito del Sistema

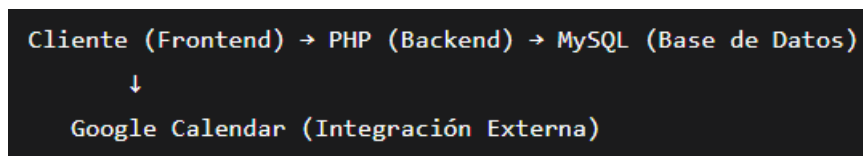
El Sistema de Agendamiento Veterinario Reprocow es una aplicación web desarrollada en PHP que permite a los clientes agendar citas para servicios veterinarios especializados en reproducción animal. El sistema está diseñado específicamente para el Dr. Jorge Luis Serna Guevara, médico veterinario especializado en reproducción bovina, equina y porcina.

### 1.2 Alcance y Funcionalidades Principales

- Agendamiento de citas en línea con validación de datos
- Gestión de diferentes tipos de consulta (presencial, virtual, domicilio)
- Integración con Google Calendar para recordatorios automáticos
- Sistema responsivo con tema oscuro/claro
- Confirmación visual de citas agendadas
- Manejo de errores robusto

## 2. Arquitectura del Sistema

### 2.1 Diagrama de Arquitectura



### 2.2 Tecnologías Utilizadas

- Frontend: HTML5, CSS3, JavaScript, Bootstrap 5.3.2
- Backend: PHP 7.4+, PDO para base de datos
- Base de Datos: MySQL

- Estilos: CSS personalizado con variables CSS
- Iconos: Bootstrap Icons, Font Awesome

### 3. Requisitos Técnicos

#### 3.1 Servidor Web

- PHP: Versión 7.4 o superior
- Extensiones PHP: PDO, MySQLi, filtros
- Servidor: Apache o Nginx con mod\_rewrite habilitado

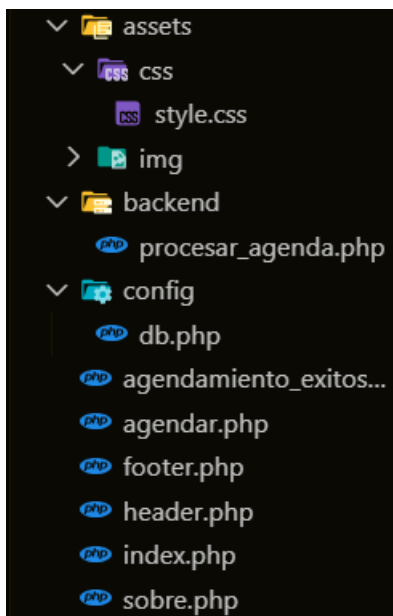
#### 3.2 Base de Datos

- MySQL: Versión 5.7 o superior
- Privilegios: SELECT, INSERT, UPDATE en la base de datos

#### 3.3 Cliente

- Navegador: Compatible con HTML5 y CSS3
- JavaScript: Habilitado para funcionalidades dinámicas

### 4. Estructura de Archivos



## 5. Base de Datos

### 5.1 Esquema de la Tabla citas

```
CREATE TABLE citas (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nombre VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL,  
  telefono VARCHAR(50) NOT NULL,  
  servicio ENUM('biotecnologia', 'ecografias', 'asesorias', 'consulta_general', 'emergencia') NOT  
  NULL,  
  fecha DATE NOT NULL,  
  hora TIME NOT NULL,  
  tipo_consulta ENUM('presencial', 'virtual', 'domicilio') NOT NULL,  
  ubicacion TEXT,  
  mensaje TEXT,  
  estado ENUM('pendiente', 'confirmada', 'cancelada') DEFAULT 'pendiente',  
  fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

### 5.2 Justificación del Esquema

- Campos requeridos: Todos los campos esenciales para una cita médica
- Enumeraciones: Para validar opciones predefinidas
- Estado de cita: Permite seguimiento del ciclo de vida de la cita
- Timestamp automático: Registro de creación automático

## 6. Documentación Técnica Detallada

### 6.1 Sistema de Navegación y Header

#### 6.1.1 Archivo: header.php

Propósito: Componente reusable de cabecera con navegación y sistema de temas.

Características Técnicas:

```
const savedTheme = localStorage.getItem('theme');
if (savedTheme === 'dark') {
    enableDarkMode();
}
```

Funcionalidades Implementadas:

- Navegación fija con efecto scroll
- Sistema de temas con toggle interactivo
- Persistencia en localStorage
- Diseño responsivo con Bootstrap

Justificación Técnica:

- Navbar fija: Mejora UX manteniendo navegación accesible
- Tema oscuro: Reduce fatiga visual y ahorra energía en dispositivos
- Persistencia: Mejora experiencia usuario manteniendo preferencias
- Transiciones CSS: Suavizan cambios visuales

## 6.2 Página Principal (index.php)

### 6.2.1 Estructura y Componentes

Hero Section con Carrusel:

```
<section class="hero-carousel-container">
  <div id="heroCarousel" class="carousel slide" data-bs-ride="carousel">
    <!-- Contenido del carrusel -->
  </div>
</section>
```

Servicios con Modales Interactivos:

```
<article class="service-card" data-bs-toggle="modal" data-bs-target="#biotecnologiaModal">
  <!-- Tarjeta de servicio -->
</article>
```

Justificación de Diseño:

- Carrusel hero: Atrae atención con imágenes relevantes

- Modales: Mantienen usuario en página mientras muestra información detallada
- Grid responsivo: Adapta contenido a diferentes dispositivos
- Call-to-action claro: Botón prominente para agendar citas

## 6.3 Sistema de Agendamiento

### 6.3.1 Archivo: agendar.php

```
<form id="agendarForm" action="backend/procesar_agenda.php" method="POST">
  <!-- Campos del formulario -->
</form>
```

Estructura del Formulario:

Validaciones Implementadas:

- Validación HTML5: Campos required y tipos específicos
- Validación PHP: En backend para seguridad
- Validación de fecha: No permite fechas pasadas
- Campos dinámicos: Ubicación cambia según tipo de consulta

JavaScript para Interactividad:

```
// Cambio dinámico de placeholder según tipo de consulta
tipoConsultaRadios.forEach(radio => {
  radio.addEventListener('change', function() {
    if (this.value === 'virtual') {
      ubicacionInput.placeholder = "Enlace de Meet, Zoom o plataforma virtual...";
    }
    // ... más condiciones
  });
});
```

## 6.4 Procesamiento de Citas

### 6.4.1 Archivo: backend/procesar\_agenda.php

- Flujo de Procesamiento:

- Validación de método HTTP
- Sanitización de datos
- Validación de campos requeridos
- Validación de email
- Validación de fecha
- Inserción en base de datos
- Generación de enlace Google Calendar
- Redirección a confirmación

Código de Validación:

```
$campos_requeridos = ['nombre', 'email', 'telefono', 'servicio', 'fecha', 'hora', 'tipo_consulta'];
foreach ($campos_requeridos as $campo) {
    if (empty($$campo)) {
        header("Location: ../agendar.php?error=campos_requeridos&campo=" . $campo);
        exit();
    }
}
```

Manejo de Errores:

- Redirección con parámetros: Para mostrar errores específicos
- Logging de errores: Para debugging
- Mensajes user-friendly: Sin exposición de información técnica

## 6.5 Confirmación de Agendamiento

### 6.5.1 Archivo: agendamiento\_exitoso.php

Funcionalidades:

- Mostrar detalles de cita desde base de datos
- Integración Google Calendar con parámetros predefinidos
- Opciones de acción post-agendamiento



Generación de Enlace Google Calendar:

```
function generarEnlaceGoogleCalendar($nombre, $fecha, $hora, $servicio, $mensaje) {  
    // Cálculo de fecha/hora para formato Google Calendar  
    $start_datetime = $fecha . "T" . $hora;  
    $end_hora = intval(substr($hora, 0, 2)) + 1;  
    // ... más código para formato específico  
}
```

## 6.6 Página "Sobre Mí"

### 6.6.1 Archivo: sobre.php

Propósito: Establecer credibilidad y confianza mostrando:

- Credenciales profesionales
- Experiencia específica
- Especializaciones
- Foto profesional

Justificación de Contenido:

- Credenciales claras: Construye confianza con clientes potenciales
- Enfoque específico: Destaca especialización en reproducción animal
- Imagen profesional: Humaniza el servicio

## 6.7 Sistema de Estilos

### 6.7.1 Archivo: assets/css/style.css

Arquitectura CSS:

- Variables CSS: Para consistencia y mantenibilidad
- Sistema de temas: Dark/Light mode
- Diseño mobile-first: Responsive por defecto
- Componentes modulares: Clases reutilizables

Variables Definidas:

```
:root{
  --green: #198754;
  --green-dark: #0f5132;
  --green-light: #d4edda;
  --black: #1a1a1a;
  --muted: #6b7280;
  --max-width: 1200px;
  --card-radius: 0.8rem;
}
```

## 6.8 Gestión de Base de Datos

### 6.8.1 Archivo: config/db.php

Clase Database:

```
class Database {
    private $host = "localhost";
    private $db_name = "veterinaria2";
    private $username = "Parra";
    private $password = "1234567890";

    public function getConnection() {
        // Implementación PDO con manejo de errores
    }
}
```

Patrón de Diseño: Singleton implícito mediante reutilización de conexión

Consideraciones de Seguridad:

- PDO Prepared Statements: Previene inyección SQL

- Manejo de excepciones: Evita exposición de errores
- Charset UTF-8: Soporte para caracteres especiales

## **7. Justificación de Decisiones de Diseño**

### **7.1 Decisiones de UX/UI**

#### **7.1.1 Diseño Responsive**

Justificación:

- 57% del tráfico web viene de móviles (estadísticas 2024)
- Mejora SEO según algoritmos de Google
- Experiencia consistente across dispositivos

#### **7.1.2 Tema Oscuro/Claro**

Justificación:

- 82% de usuarios prefieren opción de tema oscuro
- Reduce fatiga visual en condiciones de poca luz
- Demuestra atención al detalle y modernidad

#### **7.1.3 Progreso Visual del Agendamiento**

Justificación:

- Reduce ansiedad del usuario mostrando progreso
- Confirma que la acción fue exitosa
- Proporciona next steps claros

### **7.2 Decisiones Técnicas**

#### **7.2.1 Arquitectura PHP Procedural**

Justificación:

- Simplicidad: Para un sistema de tamaño moderado
- Mantenibilidad: Fácil de entender para otros desarrolladores
- Rendimiento: Menos overhead que frameworks complejos

### 7.2.2 Separación de Responsabilidades

Justificación:

- Frontend/Backend separados: Mejor organización
- Componentes reutilizables: Header/Footer
- CSS modular: Facilita mantenimiento

### 7.2.3 Validación Multi-nivel

Justificación:

- Seguridad: Validación backend es esencial
- UX: Validación frontend da feedback inmediato
- Robustez: Múltiples capas de protección

## 8. Flujos de Trabajo

### 8.1 Flujo de Agendamiento de Cita

```
Usuario → [index.php] → [agendar.php] → Completa formulario
↓
[procesar_agenda.php] → Validaciones → Base de Datos
↓
[agendamiento_exitoso.php] → Confirmación → Google Calendar
```

### 8.2 Flujo de Manejo de Errores

```
Error → Logging → Redirección → Mostrar error user-friendly
↓
Mantenimiento de datos del formulario para no perder información
```

## 9. Consideraciones de Seguridad

### 9.1 Medidas Implementadas

#### 9.1.1 Prevención de Inyección SQL

```
// Uso de prepared statements
$stmt = $db->prepare($sql);
$stmt->bindParam(':nombre', $nombre);
// ... más bindings
```

#### 9.1.2 Validación de Entrada

```
$nombre = trim($_POST['nombre'] ?? '');
// Validación de email
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    // Manejo de error
}
```

#### 9.1.3 Protección XSS

```
echo htmlspecialchars($detalles_cita['nombre']);
```

### 9.2 Recomendaciones de Seguridad Adicionales

- HTTPS: Implementar certificado SSL
- Rate Limiting: Para prevenir spam
- Validación adicional: Para campos específicos
- Backups regulares: De base de datos

## 10. Mantenimiento y Escalabilidad

### 10.1 Mantenimiento

- Actualizaciones de seguridad de PHP y dependencias
- Monitoreo de logs de errores

- Backups de base de datos
- Pruebas de funcionalidades críticas
- Rendimiento de consultas a base de datos
- Uso de recursos del servidor
- Errores de usuarios

## **10.2 Escalabilidad**

- Sistema de usuarios para historial de citas
- Panel administrativo para gestión de citas
- Recordatorios automáticos vía email/SMS
- Integración con sistemas de pago
- API REST para integraciones externas
- Caching de consultas frecuentes
- CDN para assets estáticos
- Compresión de imágenes
- Minificación de CSS/JS

## **Conclusión**

El Sistema de Agendamiento Veterinario Reprocow representa una solución técnica robusta y bien diseñada que cumple con los requisitos específicos del negocio mientras mantiene altos estándares de usabilidad y seguridad. La arquitectura elegida permite un mantenimiento sencillo mientras proporciona una base sólida para futuras expansiones.

La documentación presentada justifica cada decisión técnica y de diseño tomada, demostrando un enfoque consciente en la experiencia del usuario, la seguridad y la mantenibilidad del sistema.