

SYSREST - Paso a Paso del Despliegue

ESTUDIANTES:

Mateo Arias Cáceres

Sara Lizeth Lozano

INSTRUCTOR:

Gustavo Adolfo Rodriguez Quinayas

TECNOLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE

Centro de Comercio y Servicios Popayán/ Cauca

Modalidad (Virtual)

FICHA (2675809)

AÑO 2024



SYSREST - Paso a Paso del Despliegue

Propósito del desarrollo

El propósito principal del sistema SYSREST es gestionar la asignación de turnos de manera automática y eficiente. Está diseñado para cubrir las necesidades de gestión de citas en instituciones que prestan servicios como asesorías y atención en cajas. El sistema permite a los usuarios reservar citas y turnos de forma ágil, minimizando tiempos de espera y optimizando el flujo de atención.

Además, SYSREST incluye funcionalidades como la asignación automática de personal a los turnos, teniendo en cuenta restricciones laborales y el número máximo de turnos permitidos por día. Esto asegura una distribución justa y equitativa de la carga de trabajo.

Arquitectura General

La arquitectura del sistema SYSREST está basada en un modelo cliente-servidor. El frontend está desarrollado con PHP y CSS para permitir una experiencia de usuario fluida en la asignación y consulta de turnos. El backend, también en PHP, maneja la lógica del negocio, incluyendo la interacción con la base de datos MariaDB para almacenar la información de clientes, servicios y turnos.

El sistema está dividido en módulos: uno para la gestión de clientes, otro para la gestión de servicios, y un módulo central para la asignación y visualización de turnos. Cada módulo interactúa con la base de datos a través de consultas SQL optimizadas, garantizando un rendimiento adecuado incluso bajo cargas de trabajo altas.

Diagrama de Arquitectura

A continuación se muestra un diagrama simplificado de la arquitectura del sistema SYSREST:



Configuración de la Base de Datos

La base de datos utilizada por SYSREST es MariaDB, un sistema de gestión de bases de datos relacional basado en MySQL. Para configurar la conexión a la base de datos en PHP, es necesario definir los parámetros de conexión en el archivo 'config.php', como se muestra a continuación.

```
config.php X
includes > config.php
1  <?php
2  // Mostrar errores
3  ini_set('display_errors', 1);
4  ini_set('display_startup_errors', 1);
5  error_reporting(E_ALL);
6
7  // Conexión a la base de datos
8  $servername = "localhost";
9  $username = "root"; // Cambia esto si usas otro usuario
10 $password = "nueva_contraseña"; // Cambia esto si tienes una contraseña
11 $dbname = "sysrest"; // Asegúrate de que este sea el nombre correcto
12
13 // Crear conexión
14 $conn = new mysqli($servername, $username, $password, $dbname);
15
16 // Comprobar conexión
17 if ($conn->connect_error) {
18     die("Conexión fallida: " . $conn->connect_error);
19 }
20 ?>
21
```

Es importante asegurarse de que el servidor de base de datos MariaDB esté en ejecución y accesible desde el servidor de aplicaciones. Además, las tablas en la base de datos deben estar correctamente definidas con las relaciones y restricciones apropiadas para asegurar la integridad de los datos.

Despliegue del Sistema

El despliegue de SYSREST puede realizarse en un servidor Apache con soporte para PHP. El proceso de despliegue incluye la instalación de las dependencias necesarias, la configuración del servidor y la puesta en marcha del sistema en un entorno de producción.

A continuación, se detallan los pasos para desplegar el sistema:

1. Instalar Apache, PHP y MariaDB en el servidor. Esto se puede hacer utilizando gestores de paquetes como apt-get o yum.
2. Configurar el archivo 'config.php' con las credenciales y la URL de la base de datos.
3. Subir todos los archivos del proyecto SYSREST al directorio web del servidor Apache (ejemplo:/var/www/html/sysrest).
4. Configurar los permisos del directorio para asegurar que el servidor web pueda acceder a los archivos.
5. Asegurarse de que Apache esté ejecutándose en el puerto 8080 o el puerto deseado. Esto se puede ajustar en el archivo de configuración de Apache (httpd.conf).
6. Realizar pruebas de integración para asegurarse de que el frontend se comunica correctamente con el backend y la base de datos.
7. Implementar medidas de seguridad, como la protección de rutas sensibles con HTTPS, para asegurar la integridad de las comunicaciones.

Logging y Monitoreo

Es fundamental implementar un sistema de logging adecuado para monitorear el funcionamiento del sistema. En SYSREST, se recomienda registrar eventos importantes como las reservas de turnos, asignaciones de personal, y errores en el servidor. Esto puede hacerse mediante la utilización de herramientas como Monolog para PHP, que permite la generación de logs en archivos o servicios externos.

Además, se pueden implementar herramientas de monitoreo como Prometheus y Grafana para visualizar métricas del rendimiento del sistema y asegurar una alta disponibilidad en entornos de producción.

Recomendaciones para el Entorno de Producción

En entornos de producción, es importante asegurar la escalabilidad y disponibilidad del sistema. Se recomienda utilizar Docker para contenerizar los diferentes servicios del sistema, permitiendo una fácil gestión y despliegue de actualizaciones.

Para una mayor flexibilidad, es posible utilizar Kubernetes para orquestar los contenedores en clústeres, lo que proporciona balanceo de carga automático y autoescalado en función de la demanda del sistema.

