# Documentación del Sistema de Notificaciones

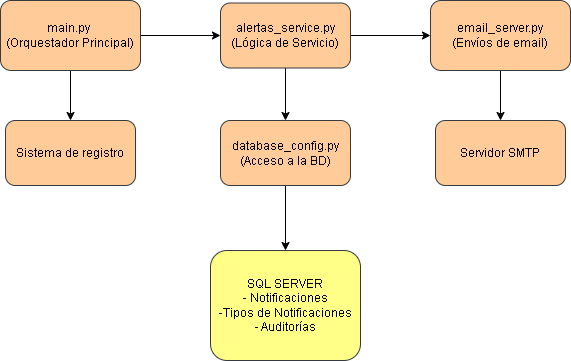
## **Descripción General**

El **Sistema de Notificaciones** es una aplicación desarrollada en Python diseñada para **procesar, enviar y auditar notificaciones por correo electrónico de forma automatizada**. Su arquitectura modular permite una integración sencilla con bases de datos SQL Server, soporte de configuración mediante variables de entorno, y análisis visual del rendimiento a través de un dashboard interactivo creado con Plotly/Dash.

* **Procesamiento automático** de notificaciones pendientes
  + Detecta y gestiona notificaciones en estado "pendiente" desde una base de datos.
  + Aplica validaciones básicas (como la existencia de destinatario y formato de email).
* **Envío de emails** con configuración SMTP
  + Usa smtplib y variables de entorno para conectarse de forma segura a un servidor SMTP.
* **Auditoría completa** de todas las operaciones
  + Registra eventos clave como envíos exitosos, errores y cambios de estado en una tabla de auditoría.
  + Trazabilidad total para cada notificación enviada o fallida.
* **Dashboard visual** para análisis
  + Visualización de métricas como la tendencia de notificaciones enviadas por tipo y la distribución de estados (enviado, pendiente, error).

## **Arquitectura del Sistema**

### **Diagrama de Arquitectura**



### **Componentes Principales**

|  |  |  |
| --- | --- | --- |
| **Componentes** | **Propósitos** | **Dependencias** |
| main.py | Orquestador principal y bucle de ejecución | alertas\_service, logging |
| alertas\_service.py | Lógica procesamiento | database\_config, email\_service |
| email\_service.py | Envío de emails vía SMTP | smtplib |
| database\_config.py | Acceso a datos y conexión DB | pyodbc |
| dashboard\_plotly.py | Visualización y análisis | plotly, dash, pandas |

## **Componentes del Sistema**

### **1. Main.py - Orquestador Principal**

**Propósito**: Ejecuta el bucle principal de procesamiento cada 60 segundos.

**Funcionalidades**:

* Inicialización del sistema de logging
* Bucle infinito de procesamiento
* Control de intervalos de ejecución

Texto

El contenido generado por IA puede ser incorrecto.

### **2. AlertasService.py**

**Propósito**: contiene la lógica principal para extraer notificaciones de la base de datos, validarlas, enviarlas y registrar auditorías.

#### **Clase ProcesadorNotificaciones**

* **Método principal**: procesar\_pendientes()

**Flujo de procesamiento**:

1. Obtener notificaciones pendientes
2. Validar cada notificación
3. Enviar email
4. Actualizar estado
5. Registrar auditoría

#### **Clase NotificacionesService**

**Métodos principales**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Método** | **Propósito** | **Parámetros** | **Retorno** |
| obtener\_notificaciones\_pendientes() | Obtiene notificaciones con estado 'pendiente' | Ninguno | Lista de notificaciones |
| actualizar\_estado\_notificacion() | Cambia el estado de una notificación | id\_notificacion, nuevo\_estado | Boolean |
| registrar\_auditoria() | Registra eventos en tabla de auditoría | id\_notificacion, accion, descripcion | Boolean |

### **3. EmailService.py - Envío de Emails**

**Propósito**: gestiona el envío de emails utilizando configuración SMTP segura.

**Configuración SMTP**:

* Servidor SMTP configurable
* Autenticación con usuario/password
* Formato HTML para emails

**Validaciones**:

* Configuración SMTP completa
* Formato básico de email
* Manejo de errores de conexión

### **4. DatabaseConfig.py - Acceso a Datos**

**Propósito**: clase reutilizable para conectarse y operar con una base de datos SQL Server utilizando pyodbc.

**Características**:

* Pool de conexiones con pyodbc
* Manejo automático de transacciones
* Conversión automática a diccionarios
* Configuración robusta con timeouts

**Métodos**:

* execute\_query(): Para consultas SELECT
* execute\_non\_query(): Para INSERT/UPDATE/DELETE
* test\_connection(): Verificación de conectividad

### **5. DashboardPlotly.py - Visualización**

**Propósito**: módulo para generar visualizaciones gráficas usando plotly y dash.

**Gráficos disponibles**:

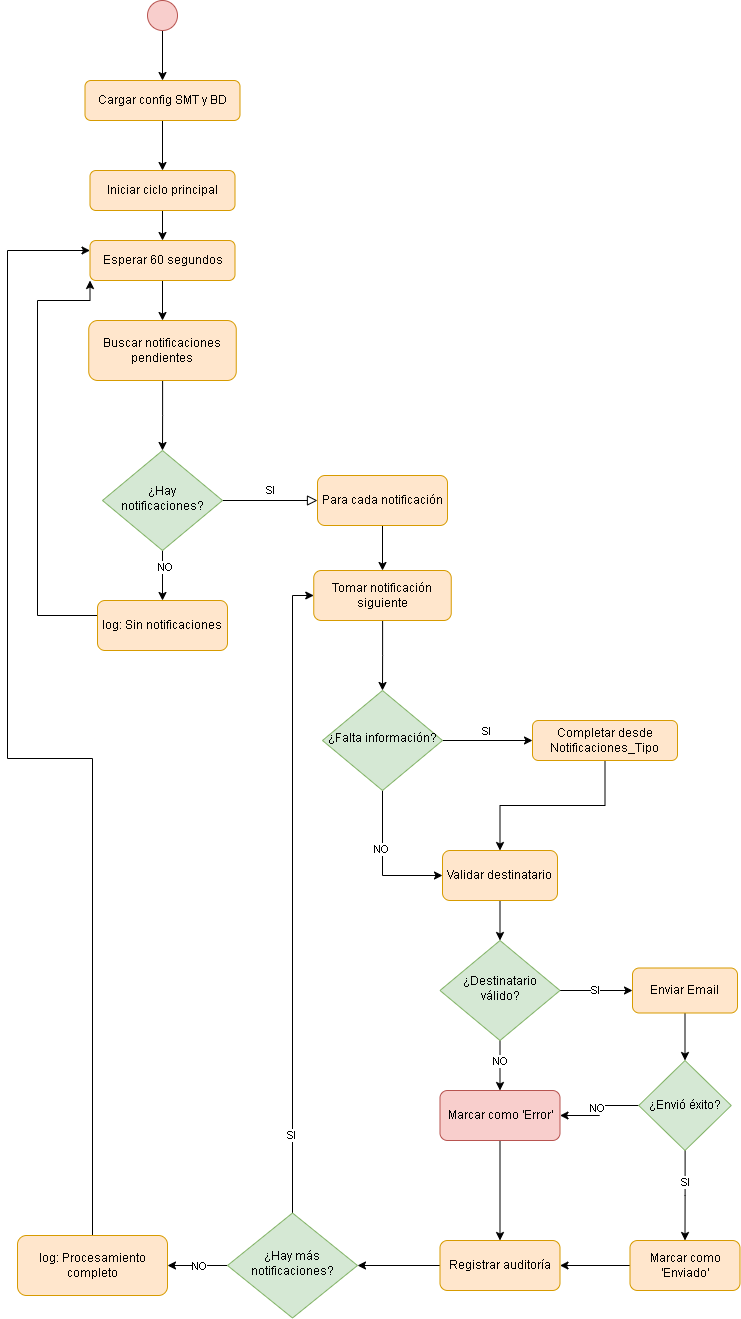
* **Líneas temporales**: Tendencias de notificaciones por tipo
* **Gráfico de dona**: Distribución de estados
* **Dashboard combinado**: Vista integral del sistema

**Períodos soportados**:

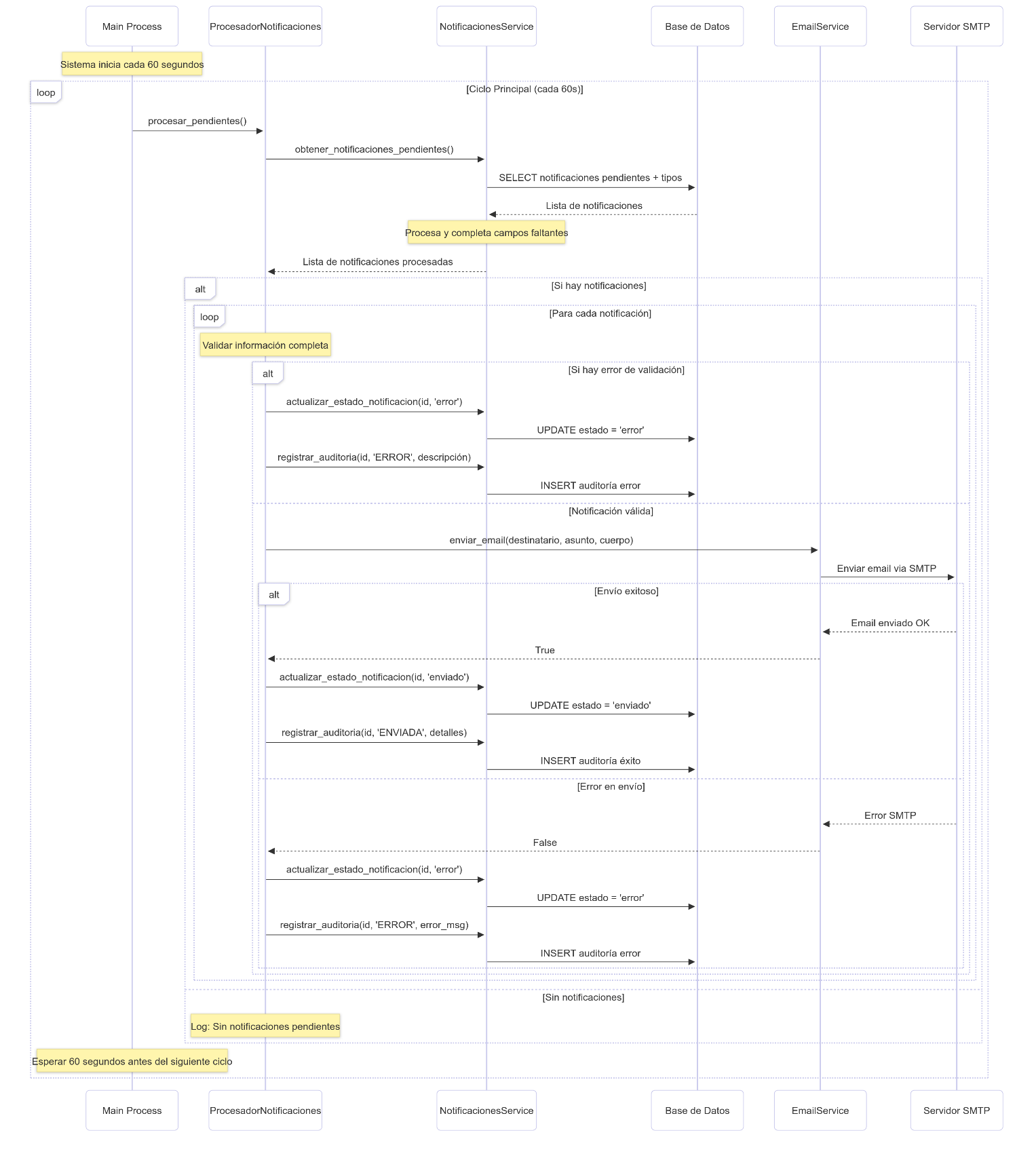
* 1 semana
* 1 mes
* 3 meses

## **Flujos de Proceso**

### **Flujo Principal de Procesamiento**



### **Diagrama de Secuencia - Procesamiento de Notificaciones**



## **Base de Datos**

### **Estructura de Tablas**

#### **Tabla: Notificaciones**

Texto

El contenido generado por IA puede ser incorrecto.

#### **Tabla: Notificaciones\_Tipo**

Texto

El contenido generado por IA puede ser incorrecto.

#### **Tabla: Auditoria**

Texto

El contenido generado por IA puede ser incorrecto.

### **Estados de Notificaciones**

|  |  |  |
| --- | --- | --- |
| **Estado** | **Descripción** | **Siguiente Estado** |
| pendiente | Notificación creada, esperando procesamiento | enviado o error |
| enviado | Notificación enviada exitosamente | Estado final |
| error | Error en el procesamiento o envío | Estado final |

## **Uso y Ejecución**

### **Instalación de Dependencias**

pip install pyodbc python-dotenv smtplib logging

pip install plotly dash pandas numpy # Para dashboard

### **Ejecución del Sistema**

#### **1. Procesador Principal**

python main.py

**Comportamiento**: Ejecuta continuamente, procesando notificaciones cada 60 segundos.

#### **2. Dashboard Interactivo**

Texto

El contenido generado por IA puede ser incorrecto.

**Acceso**: http://localhost

#### **3. Dashboard Estático**

python dashboard\_plotly.py

**Comportamiento**: Genera gráficos y los muestra en el navegador.

## **Monitoreo y Dashboard**

### **Métricas Disponibles**

#### **1. Tendencias Temporales**

* Notificaciones por día/semana/mes
* Distribución por tipo de notificación
* Líneas de tendencia automáticas

#### **2. Estados de Notificaciones**

* Porcentaje de éxito (enviado)
* Porcentaje de errores
* Notificaciones pendientes

#### **3. Análisis por Período**

* Vista semanal (últimos 7 días)
* Vista mensual (últimos 30 días)
* Vista trimestral (últimos 90 días)

### **Dashboard Interactivo**

**Características**:

* Filtros por período
* Actualización automática
* Gráficos responsivos
* Exportación a HTML

**Componentes**:

1. **Selector de período**: Dropdown para elegir rango temporal
2. **Gráfico de líneas**: Tendencias por tipo de notificación
3. **Gráfico de dona**: Distribución de estados actual
4. **Estadísticas resumidas**: Totales y promedios

### **Uso del Dashboard**

Texto

El contenido generado por IA puede ser incorrecto.

## **Mantenimiento**

### **Problemas Comunes**

#### **1. Error de Conexión a Base de Datos**

Error: [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Login failed

**Solución**: Verificar credenciales en archivo .env

#### **2. Error SMTP**

Error: (535, b'5.7.8 Username and Password not accepted')

**Solución**:

* Verificar credenciales SMTP
* Para Gmail: usar contraseña de aplicación
* Verificar que el servidor SMTP permita conexiones

#### **3. Notificaciones Sin Procesar**

**Síntomas**: Notificaciones permanecen en estado 'pendiente'

**Causas posibles**:

* Sin destinatarios configurados
* Email inválido
* Error en configuración SMTP