FACULTAD DE INGENIERÍA

INGENIERIA DE SISTEMAS

Manual de Pruebas

ARQUITECTURA DE DATOS

**AUTORES**

German Darío Sánchez Roa

Milton Fabian Ovalle García

**DOCENTE**

Juan Carlos Martínez Diaz

**NRC:**

10 – 3825

**Introducción**

El propósito de las pruebas es garantizar que el sistema de administración de censos municipales funcione correctamente, cumpla con los requerimientos funcionales y maneje de manera adecuada la interacción entre cliente, servidor y base de datos. Las pruebas incluyen:

* Pruebas Automáticas: Utilizando herramientas como Pytest para simular peticiones reales al servidor y validar respuestas.
* Pruebas Manuales: Verificando las operaciones directamente en la base de datos y la interfaz del cliente.

**Plan de Pruebas para el Proyecto de Censos Municipales**

**Configuración del Entorno de Pruebas**

1. **Entorno Virtual:**
   * Crear un entorno virtual con python -m venv venv.
   * Activar el entorno con venv/Scripts/activate (Windows) o source venv/bin/activate (Linux/Mac).
   * Instalar dependencias necesarias:
   * pip install Flask pytest pymysql
2. **Base de Datos:**
   * Configurar la base de datos MySQL con las tablas necesarias (Municipios, Departamentos) y los disparadores para actualizaciones automáticas.
3. **Servidor y Cliente:**
   * Asegurarse de que el servidor Flask esté correctamente configurado para responder a las rutas API definidas.
   * Preparar el cliente para realizar peticiones al servidor, como registrar, editar y eliminar municipios.

**Pruebas Automáticas**

**Herramientas Utilizadas**

* **Pytest: Para realizar pruebas unitarias y de integración.**

**Casos de Prueba**

**Los casos de prueba reflejan los requerimientos funcionales del proyecto:**

**Registrar un Municipio:**

* + Verificar que se puede agregar un municipio válido.
  + Validar errores al intentar agregar municipios con datos incompletos o duplicados.

**Editar un Municipio:**

* + Comprobar que se actualizan correctamente los datos del municipio.
  + Validar que los valores totales del departamento y el país también se actualizan.

**Eliminar un Municipio:**

* + Verificar que al eliminar un municipio se actualizan las estadísticas del departamento y del país.
  + Probar que no se permite eliminar municipios inexistentes.

**Buscar el Departamento de un Municipio:**

* + Validar que se devuelve el departamento correcto para un municipio dado.

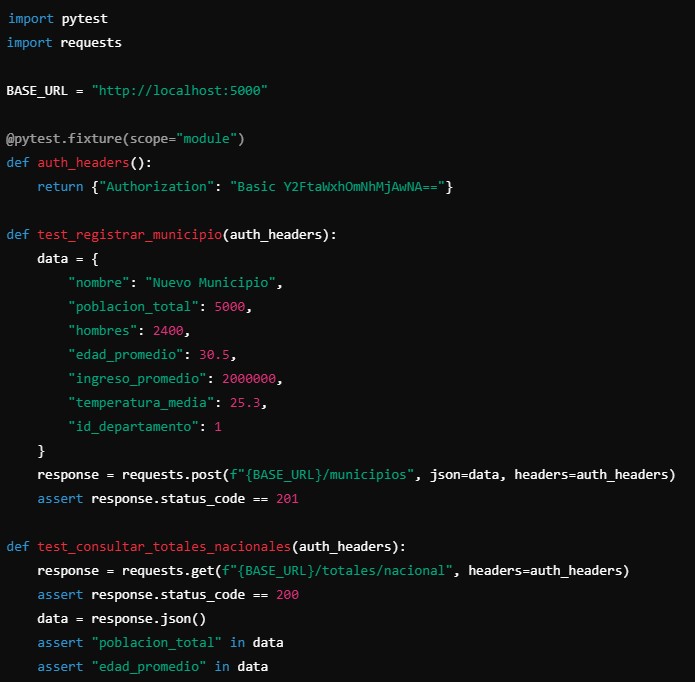
**Consultas por Ingreso y Población:**

* + Probar que se listan correctamente los departamentos con ingresos o poblaciones que cumplen criterios específicos**.**

**Visualizar Totales:**

* + Verificar que los totales por departamento y a nivel nacional son consistentes con los datos almacenados.

**Ejemplo de Código de Prueba**

****

**Pruebas Manuales**

**Interacción con la Base de Datos**

1. **Requerimientos Funcionales:**
   * **Registrar, editar y eliminar municipios directamente desde SQL.**
   * **Verificar los disparadores que actualizan los totales en la tabla Departamentos.**
2. **Consultas de Prueba:**
3. **Registrar un municipio**
4. **Resultados Esperados**

**Automáticas:**

* Respuestas HTTP con códigos 200, 201 (éxito) o 400, 404 (errores).
* Datos JSON correctamente formateados.
* Cambios reflejados en los totales de la base de datos.

**Manuales:**

* Registros y actualizaciones visibles en las tablas.
* Disparadores funcionando sin errores.

1. **Reporte Final**

**El reporte incluirá:**

* Pruebas exitosas y fallidas, con descripciones de errores.
* Ajustes realizados para alcanzar el 100% de funcionalidad.