**Arquitectura de Software**

**Documentación**

****

**TEXEA**

Integrantes:

Juan Jose Anavitate Gomez

Juan Camilo García Henao

Corporación Universitaria del Meta

2023

**Tabla de contenido**

Introducción----------------------------------------------------------------------3Requisitos del Sistema---------------------------------------------------------3  
Arquitectura del Sistema------------------------------------------------------6  
Tecnologías Utilizadas----------------------------------------------------------8  
Diagramas UML------------------------------------------------------------------9  
Configuración del Entorno de Desarrollo---------------------------------10  
Conclusión------------------------------------------------------------------------13  
Webgrafía-------------------------------------------------------------------------13

**I. Introducción**

En la vida cotidiana, una persona o una empresa poseen sus gastos o inversiones, esto hace que surja una necesidad de tener un control sobre estos no solo por tema de conocimiento, si no por expandir sus mentes en cuestiones financieras, para esto nace TEXEA quien se provee como una herramienta que almacena esos gastos, funcionando como una base de datos para posteriormente un futuro análisis de sus registros, ayudando a mantener el orden y el control de sus registros de una forma dinámica pero a su vez sencilla, de tal forma que el alcance de la plataforma se dirija tanto para empresas de cualquier tipo o también personas individuales.

**II. Requisitos del Sistema**

1. Requisitos Funcionales

|  |  |
| --- | --- |
| **Especificación de requisitos funcionales** | |
| ID | RF1 |
| Nombre | Registro e inicio de sesión |
| Tipo | General |
| Prioridad | Alta |
| Descripción: Los usuarios deben poder registrarse e iniciar sesión. | |

|  |  |
| --- | --- |
| **Especificación de requisitos funcionales** | |
| ID | RF2 |
| Nombre | Dashboard |
| Tipo | Usuario |
| Prioridad | Alta |
| Los usuarios registrados pueden acceder a un dashboard. | |
| **Especificación de requisitos funcionales** | |
| ID | RF3 |
| Nombre | Gestionar base de datos de gastos |
| Tipo | Usuario |
| Prioridad | Alta |
| Descripción: En el Dashboard, los usuarios pueden gestionar su base de datos de gastos. | |

|  |  |
| --- | --- |
| **Especificación de requisitos funcionales** | |
| ID | RF4 |
| Nombre | CRUD |
| Tipo | Usuario |
| Prioridad | Alta |
| Descripción: Se debe proporcionar un CRUD para agregar, actualizar y eliminar datos de la base de datos. | |

1. Requisitos No Funcionales

|  |  |
| --- | --- |
| **Especificación de requisitos no funcionales** | |
| ID | RNF1 |
| Nombre | Interfaz intuitiva |
| Tipo | General |
| Prioridad | Alta |
| Descripción: La interfaz de usuario debe ser intuitiva. | |

|  |  |
| --- | --- |
| **Especificación de requisitos no funcionales** | |
| ID | RNF2 |
| Nombre | Seguridad |
| Tipo | General |
| Prioridad | Alta |
| Descripción: El sistema debe ser seguro, con autenticación y autorización adecuadas. | |

|  |  |
| --- | --- |
| **Especificación de requisitos no funcionales** | |
| ID | RNF2.1 |
| Nombre | SQL Injection |
| Tipo | General |
| Prioridad | Alta |
| Descripción: La aplicación implementa medidas de seguridad para prevenir ataques de SQL injection. | |

|  |  |
| --- | --- |
| **Especificación de requisitos no funcionales** | |
| ID | RNF3 |
| Nombre | Base de datos escalable |
| Tipo | General |
| Prioridad | Media |
| Descripción: La base de datos debe ser eficiente y escalable. | |

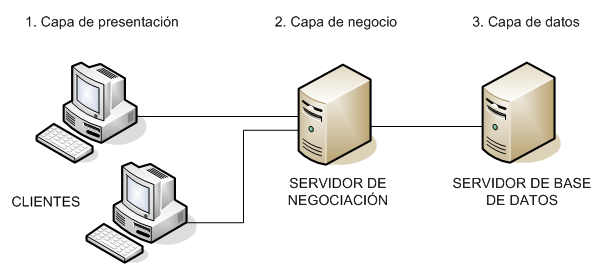
**III. Arquitectura del Sistema**

1. Arquitectura de Capas

**Capa de Presentación:** Interfaz de usuario desarrollado con HTML, CSS, JavaScript (AJAX).

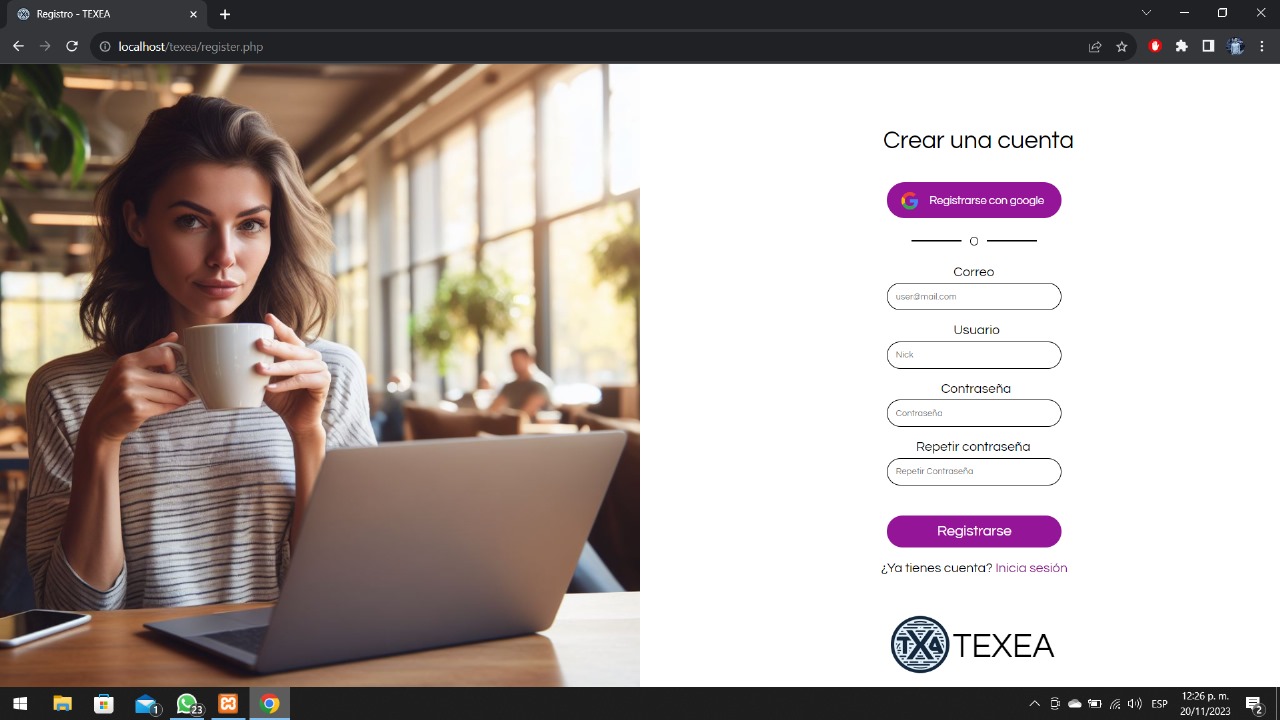
**Capa de Lógica de Negocios:** PHP (usando XAMPP).

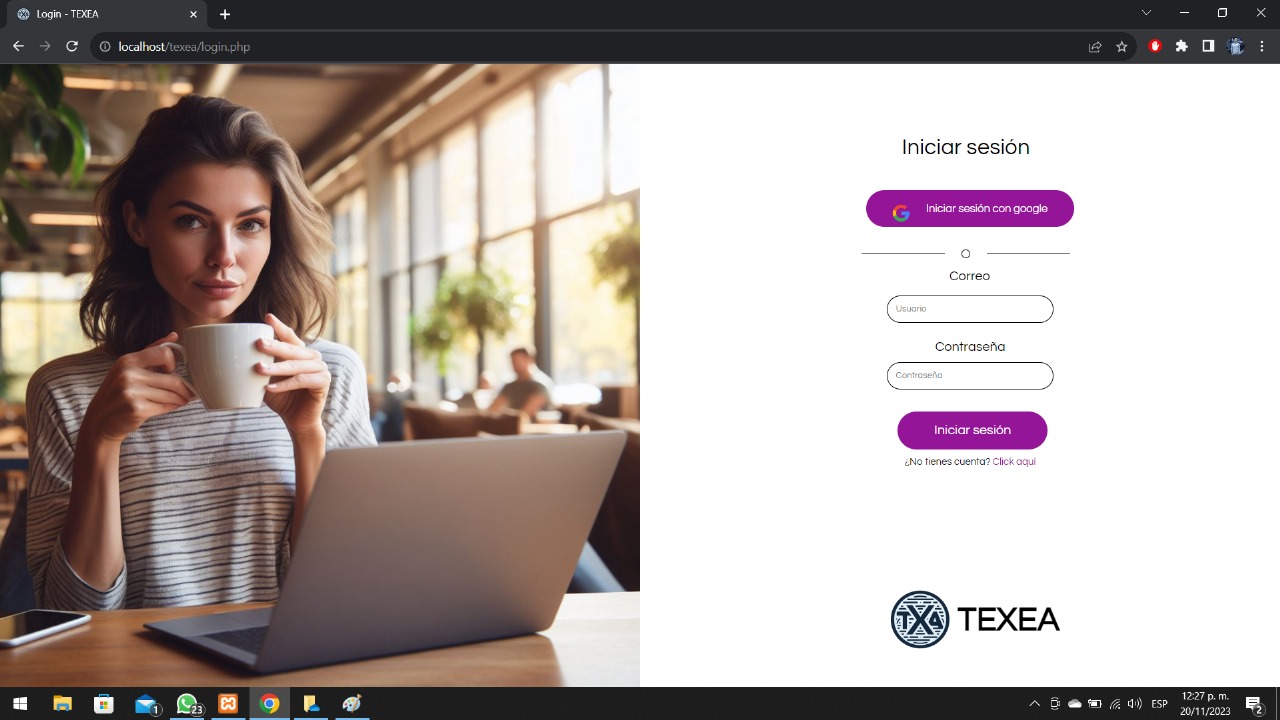
**Capa de datos:** PostgreSQL (base de datos).



2. Flujo de Control

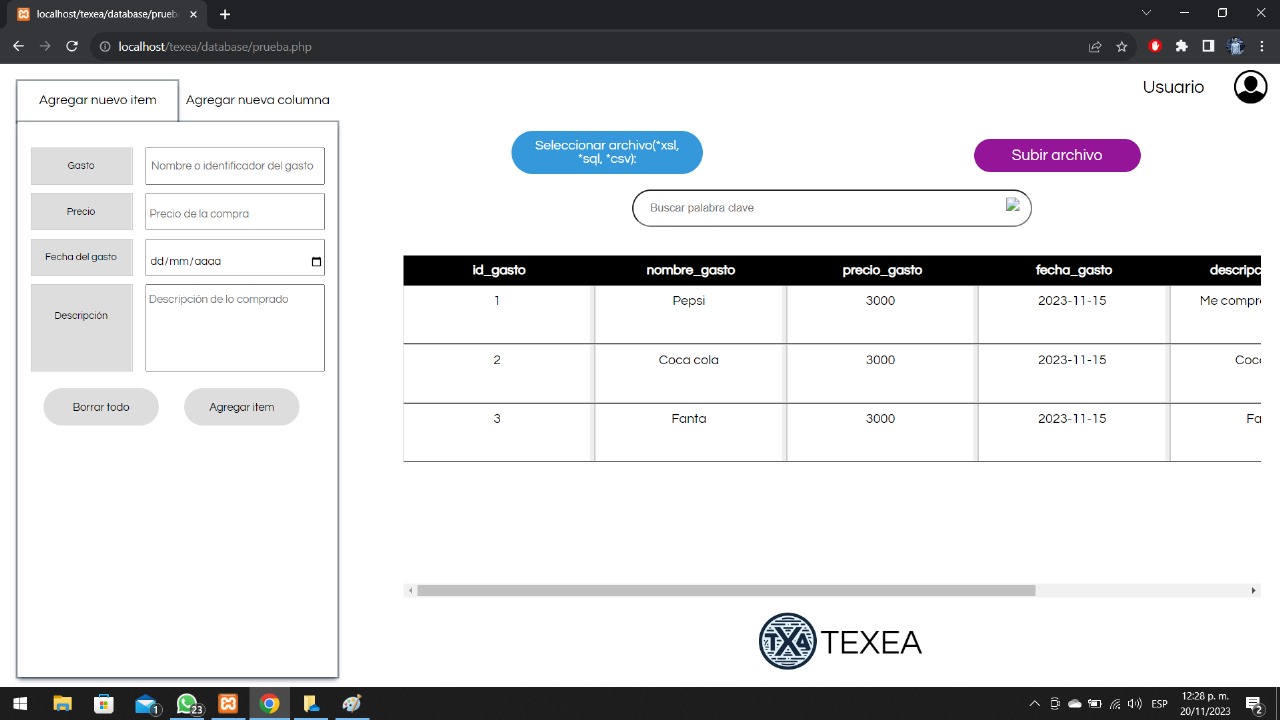
Registro e Inicio de Sesión: HTML, PHP (manejo de formularios), PostgreSQL para almacenar datos de usuario.





Dashboard: JavaScript para la interactividad, AJAX para comunicarse con el servidor.

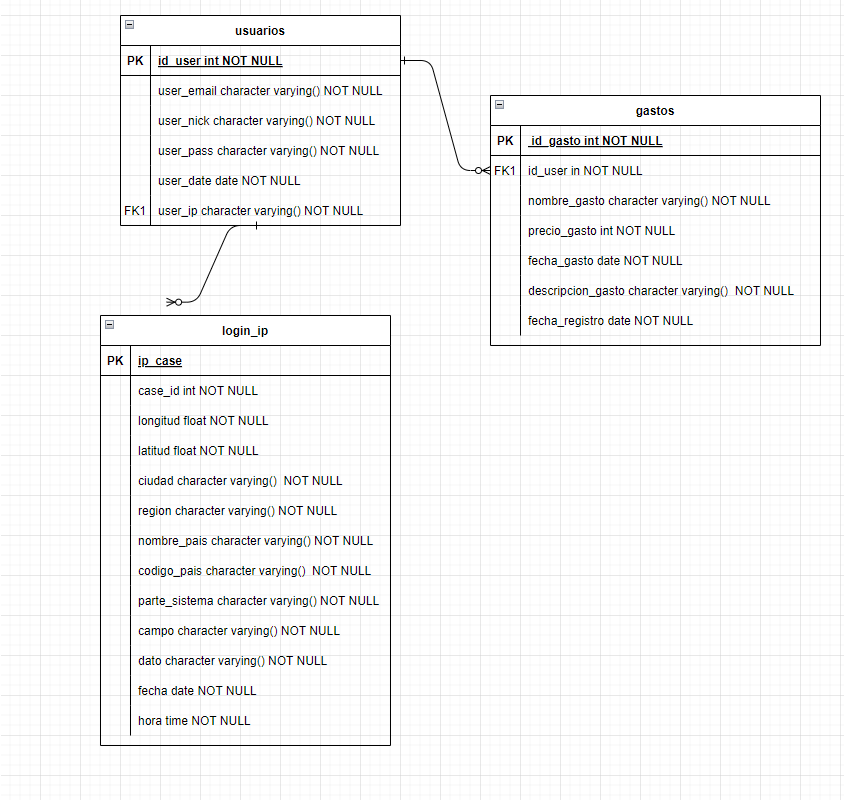
CRUD: Interacción entre el frontend y el backend utilizando PHP para procesar las operaciones CRUD, PostgreSQL para almacenar y recuperar datos.

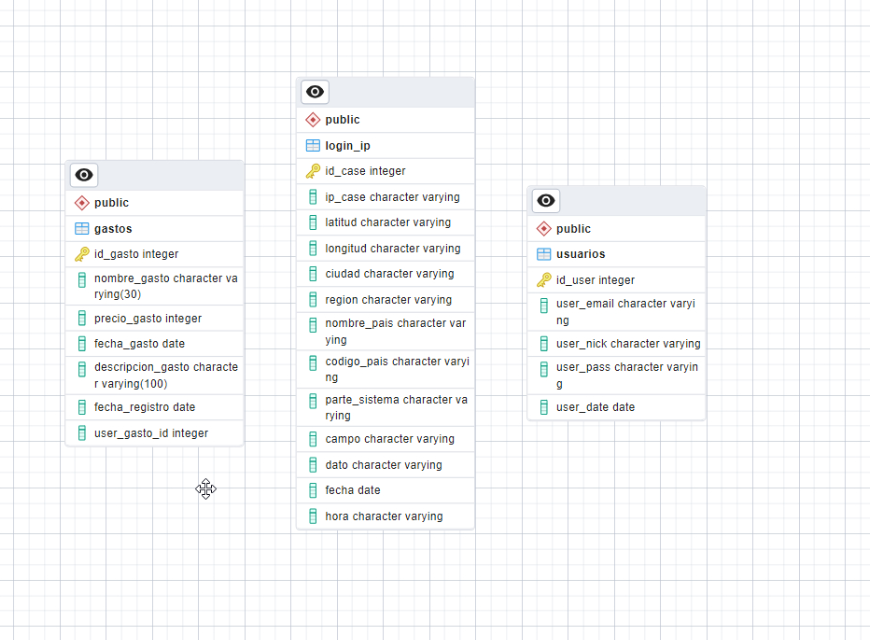
****

**Base de datos:**

Base de Datos: Texea

Esquema: public



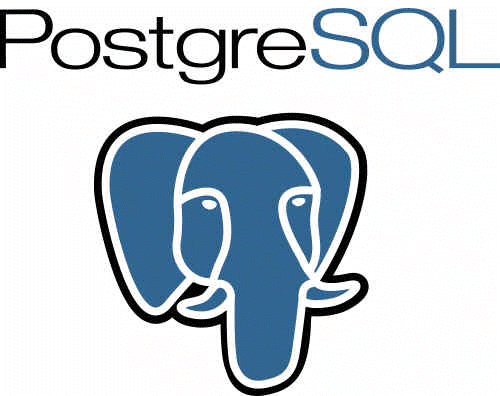
****

**IV. Tecnologías Utilizadas**

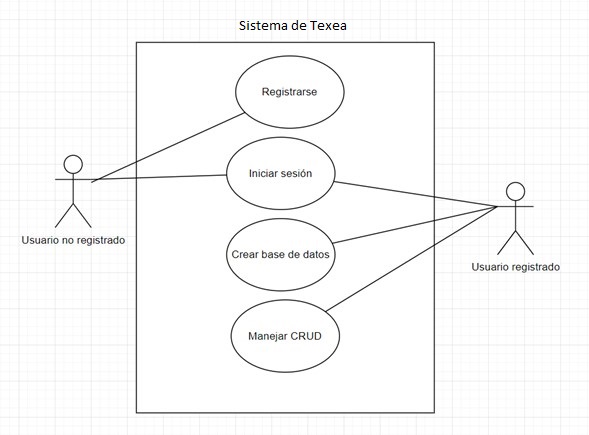
Frontend: HTML, CSS, JavaScript.



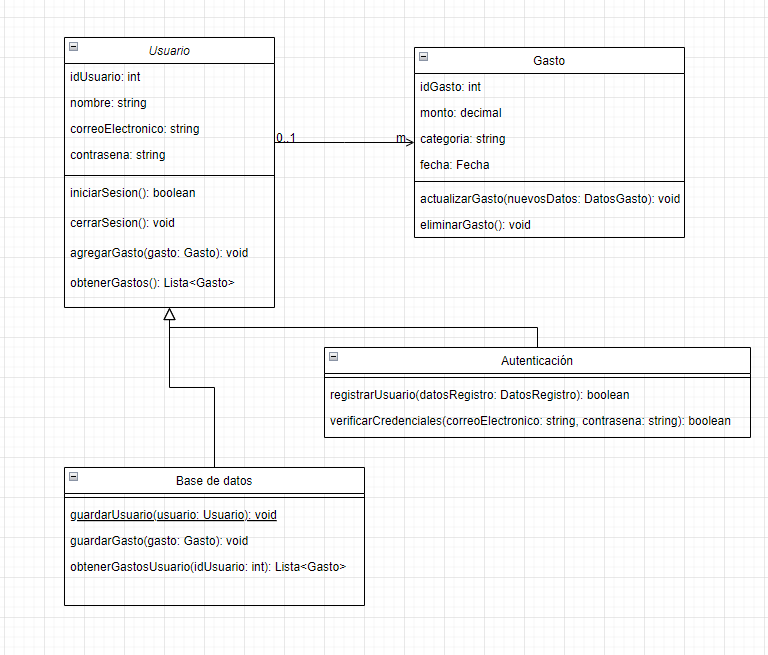
Backend: XAMPP (Apache, PHP), PostgreSQL.



**V. Diagramas UML**

1. Caso de usos de Texea

2. Diagrama de Clases

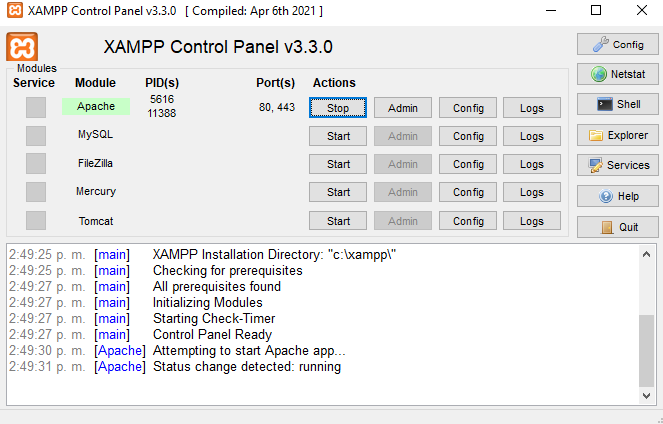


**VI. Configuración del Entorno de Desarrollo**

1. Configuración de XAMPP

Descargar e instala XAMPP desde https://www.apachefriends.org.

Inicia XAMPP y asegurarse de que Apache esté ejecutándose en el puerto 80

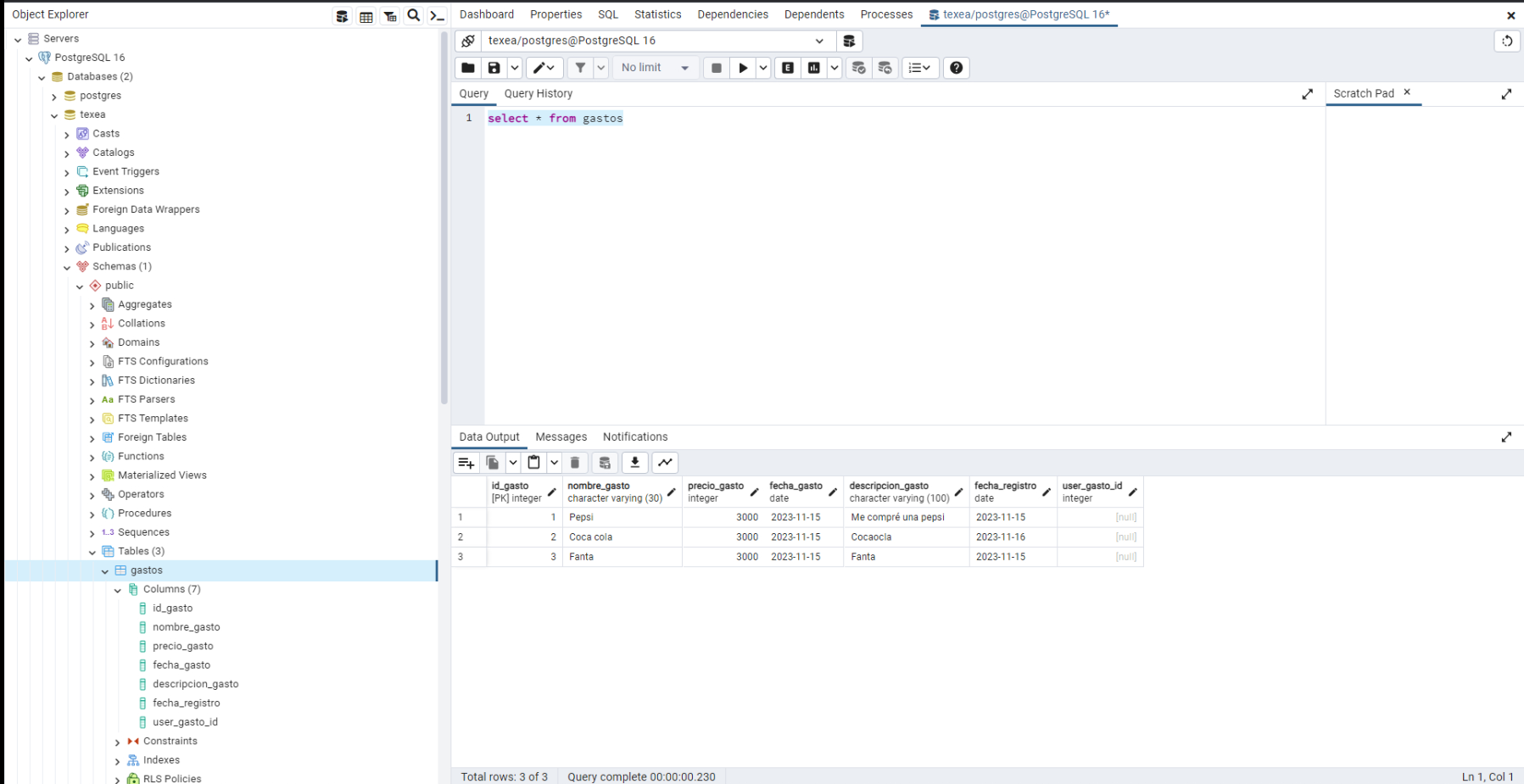


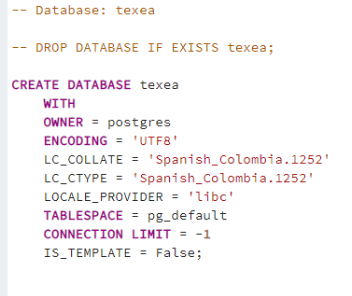
2. Configuración de la Base de Datos PostgreSQL

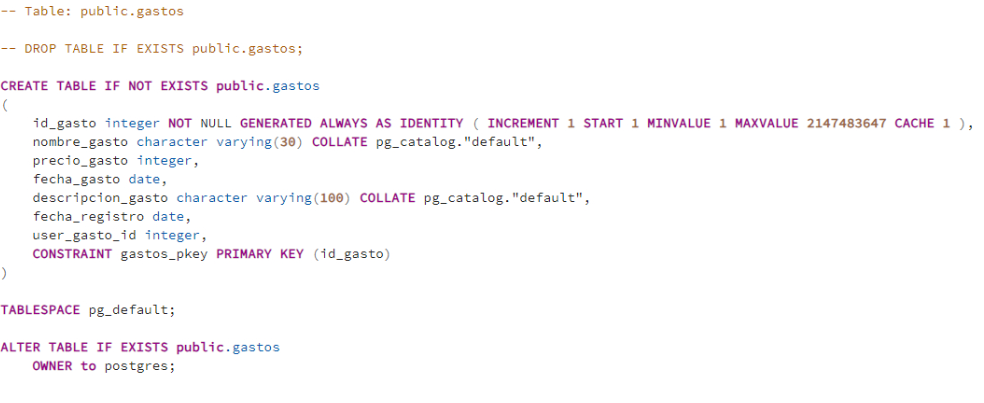
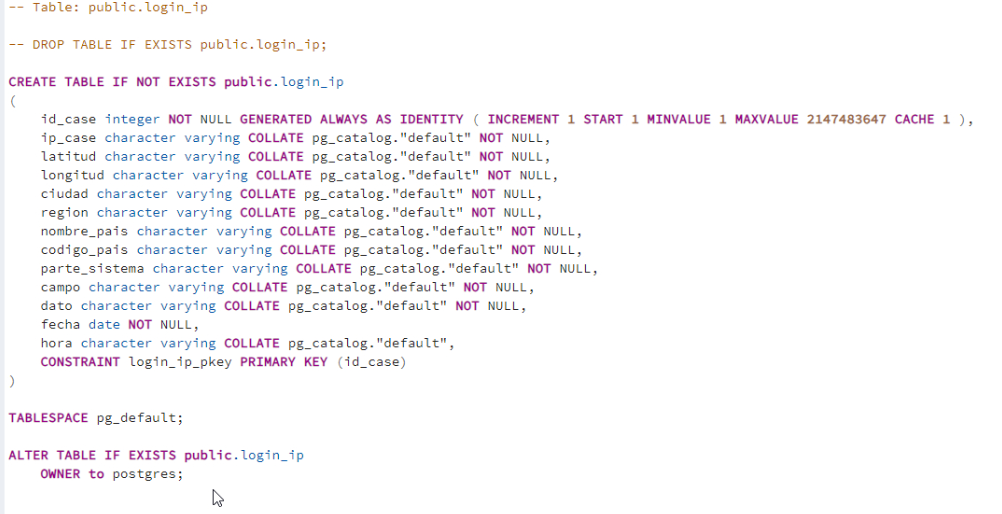
Descarga e instala PostgreSQL desde https://www.postgresql.org/download/.

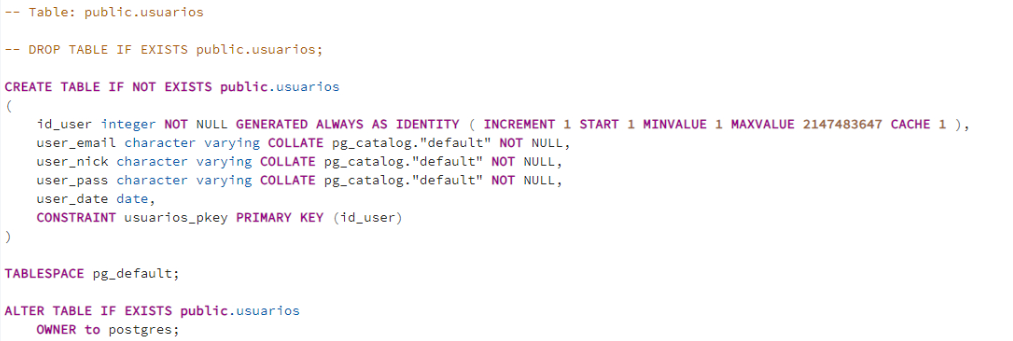
Durante la instalación, asegúrate de recordar la contraseña del usuario 'postgres', ya que la necesitarás para realizar ciertas operaciones.

Inicia el servicio de PostgreSQL y asegúrate de que esté utilizando el puerto 5432.









3. Configuración de Texea

Descargar el código fuente de Texea desde https://github.com/anavitateg/texea.git

Abre el archivo de configuración de conexión a la base de datos en el código fuente (conexionsql.php).

VII. Conclusión

Texea, un sistema de control de gastos diseñado para facilitar la gestión financiera personal, se ha desarrollado con éxito utilizando tecnologías web modernas. La arquitectura del sistema se ha estructurado en capas, con una interfaz de presentación dinámica basada en HTML, CSS y JavaScript, respaldada por un servidor local XAMPP que utiliza PHP para la lógica de negocios y se conecta a una base de datos PostgreSQL.

VIII. Webgrafía

<https://github.com/anavitateg/texea.git>

<https://www.postgresql.org/download/>

<https://www.apachefriends.org>