



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Materia

ADMINISTRACIÓN DE BASE DE DATOS

Autores (Andres Mogro, Eduardo Coronel,
David Vaca)

Estudiantes del curso

ITIZ/3201-3162

PROYECTO INTEGRADOR

Universidad de Las Américas, Quito-
Ecuador

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

1. Objetivos

- Creación de una base de datos en PostgreSQL que permita almacenar, gestionar y analizar la diversidad de información inherente al funcionamiento diario del edificio.
- Garantizar una interfaz de usuario intuitiva y fácil de navegar para los residentes y administradores del edificio para que puedan utilizarlo de manera más eficiente.

2. Definición de Cronograma

Fase 1: Planificación semana 1	Definición de objetivos, responsabilidades, roles y requerimientos necesarios
Fase 2: Diseño semana 2	Consideración de la escalabilidad, seguridad y facilidad de acceso a los usuarios del edificio en el diseño
Fase 3: Implementación semana 3	Evaluación intermedia del progreso y rendimiento de la base de datos en PostgreSQL, contenerización, creación de tablas, seguridad, etc.
Fase 4: Cierre y Documentación	Creación del manual de instalación y ejecución del programa, y cierre de la documentación

3. Definición de Roles

Administrador de Base de datos (DBA):

- Responsable del diseño y estructura de la base de datos
- Gestión de permisos y acceso a la base de datos
- Monitoreo del rendimiento y optimización de consultas

Desarrollador de Base de datos:

- Encargado del diseño y creación de tablas con sus respectivas consultas necesarias
- Desarrollo y mantenimiento de procedimientos almacenados y funciones

Analista de datos:

- Definición de las necesidades de la base de datos y análisis de los requisitos de los datos
- Colaboración con el DBA para asegurar la integridad de los datos

Responsable de Seguridad de datos:

- Garantiza la confidencialidad y seguridad de los datos
- Implementación de políticas de acceso y control de privilegios

Especialista en recuperación:

- Planes de respaldo y recuperación, asegurando recuperabilidad de los datos en caso de falla.

Usuario Final:

- Aprovechamiento de la base de datos creada con sus respectivas necesidad y roles asignados.

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

4. Asignación de Roles

Administrador de Base de datos (DBA):

- David Vaca

Desarrollador de Base de datos:

- Andres Mogro
- David Vaca

Analista de datos:

- Andres Mogro

Responsable de Seguridad de datos:

- Eduardo Coronel

Especialista en recuperación:

- Eduardo Coronel

Usuario Final:

- Propietarios del edificio

5. Asignación de Responsabilidades

Administrador de Base de datos (DBA):

- Creación de queries para realizar las tablas para una empresa que se dedica a administración de edificios

Desarrollador de Base de datos:

- Crear los queries y garantizar que la solución se adapte a la necesidad de la empresa

Analista de datos:

- Poblar las bases de datos

Responsable de Seguridad de datos:

- Evaluación y gestión de riesgos de seguridad de los datos asociados de los residentes con el sistema, como el control de acceso para garantizar quien puede acceder a información sensible.

Especialista en recuperación:

- Realizar una copia de seguridad en caliente con el fin de no detener las operaciones normales del sistema y poder garantizar un buen respaldo de la información, mejorando la productividad, facilitar recuperaciones rápidas de datos, optimizar recursos de almacenamiento, etc.

Usuario Final:

- Propietarios del edificio

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

6. Cronograma de Reuniones de trabajo

CRONOGRAMA	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4
JUEVES	Reunión inicial para definir los objetivos del proyecto y definir el cronograma	Ajustes necesarios en la planificación, recursos, tecnologías y SO	Inicio del plan estratégico, respuesta a incidentes	Reunión para realizar la documentación y ejecución del proyecto
DOMINGO	Reunión para lluvia de ideas para asignar roles y responsabilidades	Reunión para lluvia de ideas para la implementación de la solución	Implementación Del repositorio en GitHub y contenerización	Ajustes de mejora y feedback personal

7. Ideas aportadas para la implementación de la solución por cada participante

1. Base de datos centralizada: establecer esta base centralizada para poder almacenar toda la información relevante de los residentes y áreas que tengan que ver con el mantenimiento del edificio.
2. Portal en línea para los Residentes: Implementar un portal que este en línea donde los residentes puedan acceder fácilmente y pueda revisar su información personal, los días de pago y solicitar otros servicios de mantenimiento, eventos, etc.
3. Integración con Sistemas de Control de Acceso: Integrar un sistema con control de acceso para poder controlar la seguridad en las áreas comunes del edificio.
4. Panel de administración para el personal: poder brindar herramientas a los administradores para monitorear el estado de los pagos del edificio.
5. Retroalimentación y mejora: implementar un sistema de retroalimentación para que los residentes, puedan expresar sus comentarios ya sean negativos o positivos, con la finalidad de poder mejorar el sistema de gestión integral
6. Capacitación y Soporte: brindar una excelente capacitación a los residentes como también al personal administrativo para poder resolver dudas o problemas sobre nuestro sistema.

8. Aporte individual para la consecución de los objetivos del trabajo

1. Interfaces Intuitivas y Amigables: Diseño de una interfaz fácil de navegar para los residentes, realizando pruebas para descartar posibles puntos de confusión.
2. Promoción de la Sostenibilidad: Implementar prácticas que duren con el tiempo, como la reducción de papel por medio de documentación en línea.
3. Establecimiento de Metas medibles: Definir con el equipo de trabajo metas medibles para una mejora continua del sistema.

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

9. Aporte individual para la resolución de posibles conflictos

1. Canal de Comunicación Abierto: Establecer este canal para que los residentes puedan ingresar inquietudes y conflictos que se puedan presentar mediante correos electrónicos, líneas directas, etc.
2. Uso de tecnología para Reportes: Implementar una plataforma accesible para los residentes en línea facilitando el seguimiento de estos, con el fin de que los residentes puedan expresar sus reportes de manera anónima.
3. Revisión de Políticas: Revisar periódicamente las políticas y reglamentos, de seguridad para afirmar que sigan siendo importantes para el sistema, con la posibilidad de actualizar estas políticas según sean necesarios

10. Descripción de la Solución Implementada

En un edificio residencial, se requiere un sistema de gestión integral que permita administrar eficientemente la información relacionada con los residentes, las unidades del edificio, los pagos, los servicios disponibles, los problemas de mantenimiento, los eventos y anuncios, así como los accesos al edificio. La implementación de este sistema de gestión proporcionará a los administradores del edificio una herramienta completa para gestionar y monitorear la información clave relacionada con los residentes. Esto contribuirá a una administración más eficiente y a una mejor experiencia para los residentes.

Requerimientos Funcionales:

Gestión de Residentes:

- Registro de información personal de los residentes, incluyendo nombre, apellido, fecha de nacimiento y dirección.
- Asociación de cada residente con una unidad específica del edificio.

Gestión de Unidades:

- Registro de detalles de las unidades, como número de unidad, tipo, número de habitaciones y piso.
- Asociación de servicios específicos a cada unidad.

Gestión de Pagos:

- Registro de pagos mensuales realizados por los residentes.
- Asociación de cada pago con el residente correspondiente.

Gestión de Servicios:

- Registro de servicios disponibles en el edificio, incluyendo su nombre y costo mensual.
- Asignación de Servicios a Unidades:
- Asociación de servicios específicos a unidades particulares.



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Mantenimiento:

- Registro de problemas y solicitudes de mantenimiento reportados por los residentes.
- Asociación de cada problema con la unidad correspondiente.

Anuncios y Eventos:

- Registro de eventos y anuncios para los residentes.
- Inclusión de título, contenido y fecha de publicación.

Control de Acceso:

- Registro de accesos al edificio por parte de los residentes.
- Asociación de cada acceso con el residente correspondiente.

Requerimientos No Funcionales:

Seguridad:

- Garantizar la seguridad de la información sensible, como datos personales y registros de acceso.

Usabilidad:

- Interfaz de usuario intuitiva para facilitar la navegación y el uso del sistema.

Eficiencia:

- Procesamiento eficiente de consultas y transacciones para mantener un rendimiento óptimo.

Escalabilidad:

- Capacidad para manejar un crecimiento en la cantidad de residentes, unidades y registros sin perder rendimiento.

Respaldo de Datos:

- Implementación de un sistema de respaldo regular para garantizar la integridad de los datos.

11. Instrucciones para acceder al repositorio en GitHub

En este repositorio, descubrirás cómo gestionar una solución para el almacenamiento de datos destinada a una empresa centrada en la administración de edificios. Esta solución está operativa en una máquina con un sistema operativo Oracle 7, que alberga una base de datos contenerizada de PostgreSQL. La base de datos incluye 9 tablas, dos roles y dos usuarios.

<https://github.com/andres3120/Contenerizar-una-base-de-datos-postgres-con-docker-en-linux-.git>

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

12. Documentación de la instalación y ejecución del trabajo

Manual de instalación

- 1) Descargar imagen de Linux, en este caso utilizaremos la maquina Red Hat Enterprise Linux Server release 7.6 (Maipo) proporcionada por Oracle <https://yum.oracle.com/oracle-linux-isos.html>

```
[root@grupo2 ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.6 (Maipo)
[root@grupo2 ~]#
```

- 2) Para iniciar la máquina virtual usaremos el usuario root con su respectivo password= Oracle
- 3) Para actualizar el sistema operativo abrimos una terminal y ponemos el siguiente comando:
yum update
- 4) Para descargar la última versión de Docker
yum install -y Docker
- 5) Para inicializar Docker usamos el siguiente comando
systemctl start Docker
- 6) Para habilitar usamos el comando:
systemctl enable Docker
- 7) Para verificar el estatus
systemctl status Docker
- 8) Verificamos la versión
Docker --version

```
[root@grupo2 ~]# docker --version
Docker version 19.03.11-ol, build 9bb540d
```

- 9) Nos descargamos la imagen de Postgres para Docker.
docker pull postgres
- 10) Para verificar que la imagen se instaló correctamente utilizamos

```
# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	latest	75b7bff7c3ad	9 days ago	425MB

- 11) Para crear un nombre, usuario, password, base de datos, mapeo de puertos utilizamos el siguiente comando:
docker run --name grupo2-edifadmin -e POSTGRES_USER=amogro -e POSTGRES_PASSWORD=admin1234 -e POSTGRES_DB=efifadmindb -p 5433:5432 -d postgres

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

12) Ejecutamos el nombre creado en una consola bash:

```
# docker exec -it grupo2-edifadmin bash
```

13) Ingresamos a la base de datos con el usuario creado:

```
# psql -U amogro --db efifadmindb -password
```

14) En este punto ya tendremos una línea de comando para ingresar las tablas

```
efifadmindb=# create table Residentes (  
ResidenteID INT PRIMARY KEY,  
Nombre VARCHAR(50),  
Apellido VARCHAR(50),  
FechaNacimiento DATE  
);  
CREATE TABLE
```

15) Creación de usuario solo Select

```
# CREATE USER dvaca;
```

16) Asignamos una clave al usuario

```
# ALTER USER dvaca with PASSWORD '1234admin';
```

17) Asignamos permisos para usar el esquema público al usuario:

```
# GRANT USAGE ON SCHEMA public TO dvaca;
```

18) Asignamos permisos para que se conecte a la base de datos creada:

```
# GRANT CONNECT ON DATABASE "efifadmindb" TO dvaca;
```

19) Asignamos permisos solo de select al usuario:

```
# GRANT SELECT on ALL TABLES IN SCHEMA public TO dvaca;
```

20) Nos descargamos el Pg_admin, para poder gestionar la base de datos,

21) Para conectarnos ponemos la ip del servidor, nombre de la base de datos, nombre de usuario y contraseña

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Grupo2

General Connection Parameters SSH Tunnel Advanced

Host name/address 192.168.1.22

Port 5433

Maintenance database efifadmindb

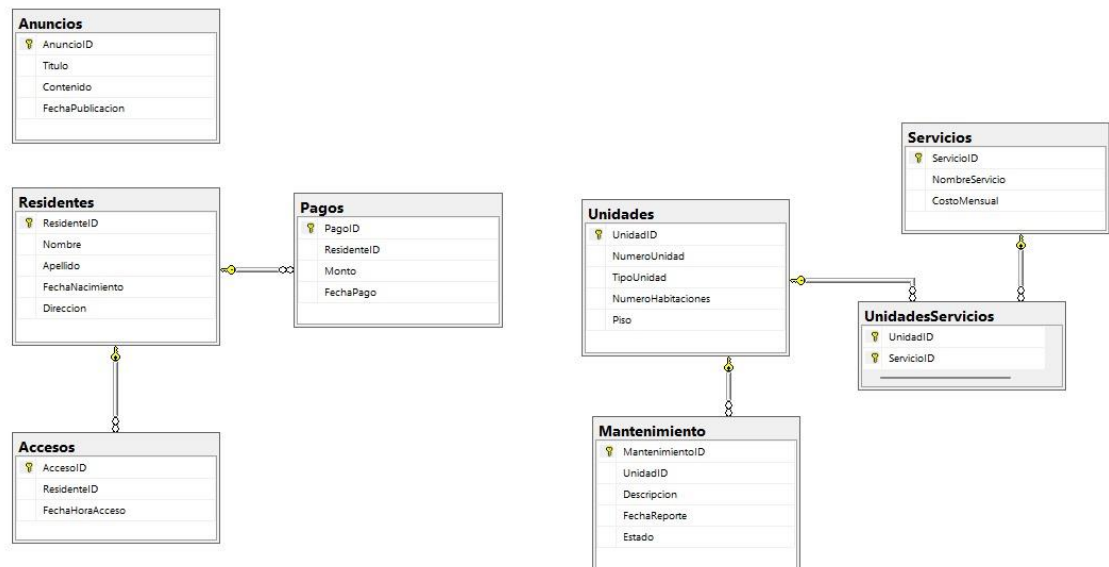
Username amogro

Kerberos authentication? ☐

Role

Service

13. Diagrama Relacional



14. Diccionario de datos

- Varchar: almacenar cadenas de texto
- INT: almacena números enteros
- Date: almacenar fechas
- Create table: creación de la tabla
- Primary Key: Identificación única
- Foreign Key: establece una relación entre 2 tablas
- Datetime: almacena fecha y hora
- Text: almacena grandes cantidades de texto



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS