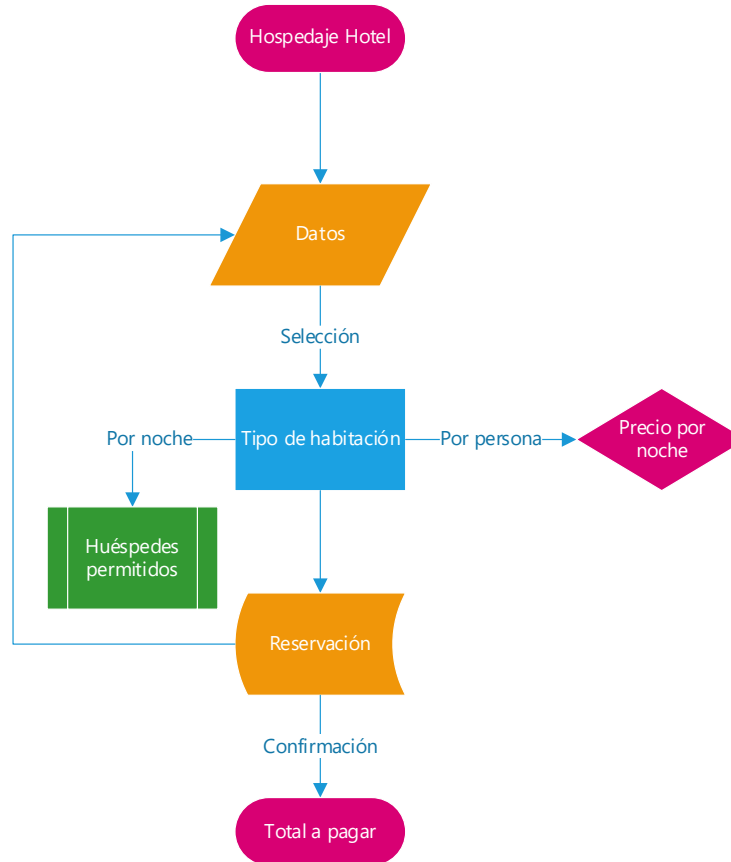


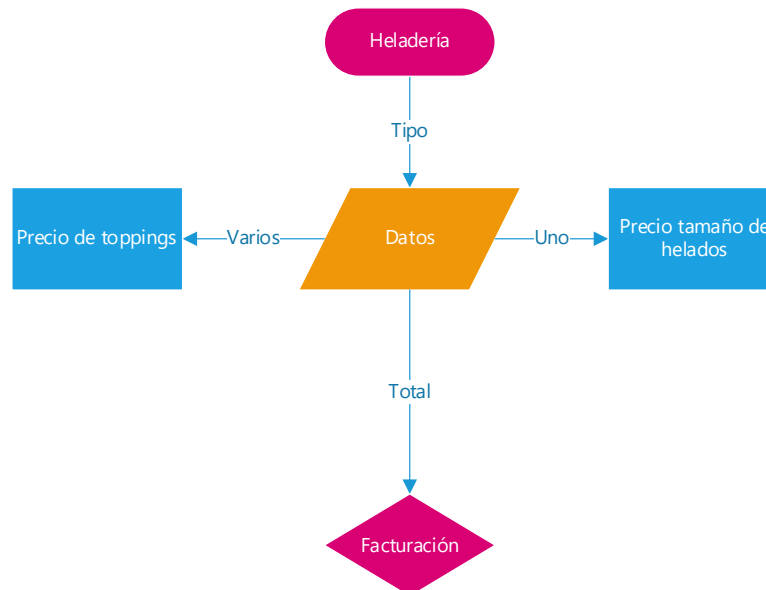
Primera Parte Segundo Examen Parcial

1. ¿Definir el proceso de su proyecto en un algoritmo y diagrama de flujo?

Programa 1:

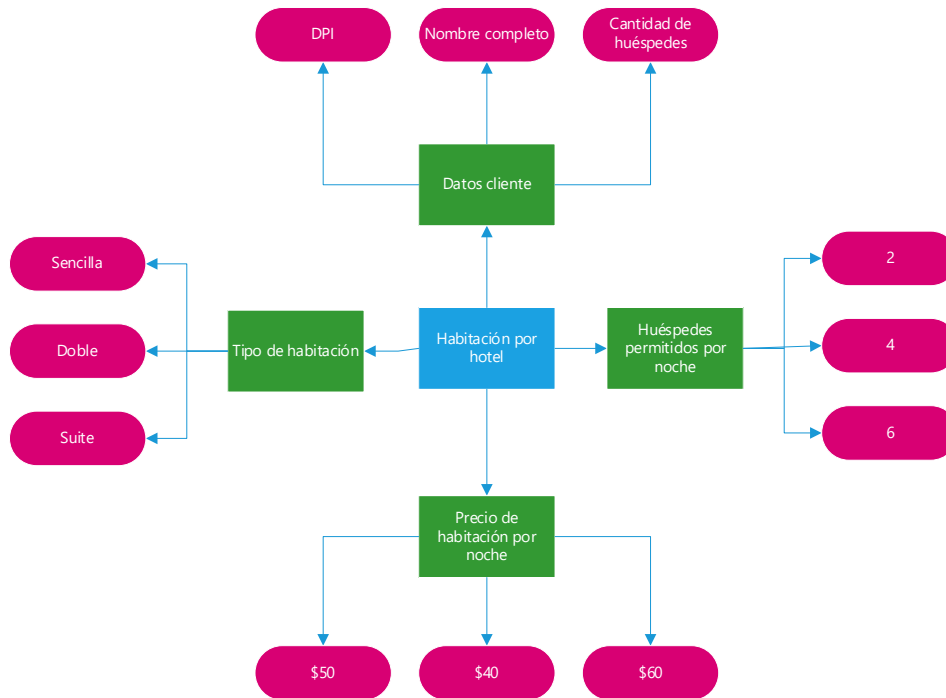


Programa 2:

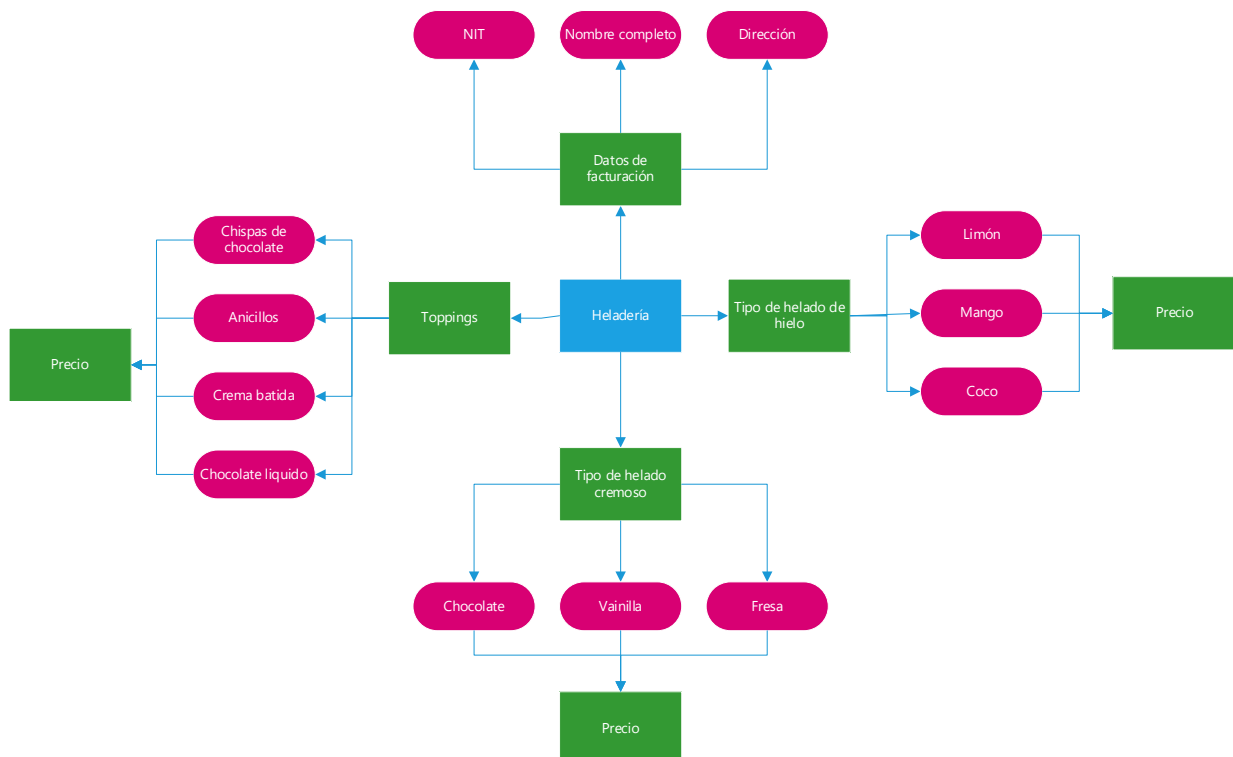


2.

Programa 1:



Programa 2:



3. ¿Comandos sql utilizados para crear la base de datos?

Programa 1:

CREATE TABLE DatosCliente (

DPI INT PRIMARY KEY,
nombre VARCHAR(50),
apellido VARCHAR(50),
email VARCHAR(50),
telefono VARCHAR(15)

);

CREATE TABLE TipoHabitacion (

idTipoHabitacion INT PRIMARY KEY,
tipo VARCHAR(50),
descripcion TEXT,
precioBase DECIMAL(10,2)

);

CREATE TABLE PrecioNoche (

idPrecio INT PRIMARY KEY,
idTipoHabitacion INT,
fecha DATE,
precio DECIMAL(10,2),
FOREIGN KEY (idTipoHabitacion) REFERENCES TipoHabitacion(idTipoHabitacion)

);

CREATE TABLE Reservacion (

idReservacion INT PRIMARY KEY,
idCliente INT,
idTipoHabitacion INT,

```
    fechaInicio DATE,  
    fechaFin DATE,  
    FOREIGN KEY (idCliente) REFERENCES DatosCliente(idCliente),  
    FOREIGN KEY (idTipoHabitacion) REFERENCES TipoHabitacion(idTipoHabitacion)  
);
```

Programa 2:

```
CREATE TABLE TipoHeladoHielo (  
    idTipoHeladoHielo INT PRIMARY KEY,  
    sabor VARCHAR(50),  
    descripcion TEXT,  
    precio DECIMAL(10,2)  
);
```

```
CREATE TABLE TipoHeladoHielo (  
    idTipoHeladoHielo INT PRIMARY KEY,  
    sabor VARCHAR(50),  
    descripcion TEXT,  
    precio DECIMAL(10,2)  
);
```

```
CREATE TABLE Toppings (  
    idTopping INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    descripcion TEXT,  
    precio DECIMAL(10,2)  
);
```

```
CREATE TABLE Facturacion (  
    idFactura INT PRIMARY KEY,  
    fecha DATE,
```

```
total DECIMAL(10,2)  
);
```

4. ¿Pasos de configuración de Django utilizado?

- Instalar Python.
- Instalar Django.
- Creación de un proyecto Django.
- Crear una aplicación Django.
- Configurar base de datos.
- Crear modelos de datos.
- Migrar base de datos.
- Crear vistas.
- Configuración de URL.

5. ¿Explicación del código utilizado?

Primero se debe crear la base de datos en PostgreSQL donde se deben de implementar todas las tablas a utilizar, luego de esto se debe enlazar la base de datos con el proyecto Django a realizar, para posteriormente introducir el código a elaborar, es importante definir muy bien la base de datos ya que esto puede generar problemas en la implementación del código en Django.

6. ¿Explicación de SQL injection y sus programas?

SQL injection consiste en una técnica de ciberataque utilizada para explotar vulnerabilidades en aplicaciones web que interactúan con una base de datos descrita en SQL. Esta técnica consiste en manipular la entrada de usuario que se envía a la base de datos, con el fin de ejecutar comandos maliciosos en el servidor de la aplicación web. Algunos de los programas más utilizados en la penetración y detección de vulnerabilidades utilizando SQL injection son SQLMap, OWASP ZAP, Burp Suite y Nessus.

7. ¿Por qué es importante asegurar el código de sus programas?

La importancia de asegurar un programa puede consistir en temas de seguridad debido a que muchas personas y entidades depositan sus datos en muchos programas utilizado en la actualidad y esa información puede ser sensible y no disponible para todo público. Fiabilidad: las personas y entidades esperan que su información este a salvo de cualquier ataque y no estar pendiente de que sus datos pueden verse comprometidos. Mantenimiento: muchas de las vulnerabilidades se deben a una falla encontrada en el código fuente que no se trata o se le presta la debida atención, lo cual provoca que personas mal intencionadas puedan aprovecharse de este error y vulnerar la seguridad fácilmente. Costos un ataque en un programa o código puede resultar demasiado costoso para empresas y organizaciones debido a que los atacantes pueden extorsionar a las organizaciones y solicitar fuertes cantidades de dinero a cambio de no publicar información sensible.

8. ¿Qué es Django y como se utiliza?

Django es un framework de alto nivel escrito en Python muy utilizado para construir aplicaciones web de manera eficiente y rápida. Este sigue el patrón de arquitectura Modelo-Vista-Controlador (MVC) donde ofrece una amplia variedad de características para el desarrollo web. Algunas de estas son: Enrutamiento de URL, capa de acceso a la base de datos, autenticación de usuarios y control de acceso a la aplicación, plantillas que facilitan la creación de interfaces de usuario y sistema de administración de contenido del sitio web.

Acceso a carpeta compartida para documentación de parte práctica:

https://drive.google.com/drive/folders/1D-YWWx2OMdluTfXqAwEh4dXZI4E7nLiE?usp=share_link