

## Proyecto semestral parte 2 - Aplicación y consultas

A continuación se presenta la descripción de un problema, que deberá ser resuelto en el semestre usando tecnología Web y el Sistema Administrador de Bases de Datos PostgreSQL. En esta segunda etapa se solicita la construcción de una aplicación web en Python en conjunto con el DBMS y los datos proporcionados. Esta actividad debe ser elaborada por los mismos grupos de 3 alumnos anteriores.

### El Proyecto

Actualmente existen múltiples plataformas para solicitar el servicio de envío de comida a domicilio, lo cual dificulta la experiencia de los usuarios al pedir comida. Muchas veces el mismo pedido de comida tiene diferentes precios en las distintas aplicaciones o algunos restaurantes sólo ofrecen en una aplicación.

En un acto de creatividad, tu grupo ha decidido construir la aplicación web *Easy Food* cuyo objetivo es concentrar en una sola plataforma todas las ofertas de comida disponible en las distintas aplicaciones de delivery, para darles una gran experiencia a los hambrientos usuarios.

### Easy Food

La aplicación *Easy Food* debe dar una gran experiencia a sus clientes, facilitándoles de manera extraordinaria el proceso de búsqueda y selección de comida, despacho, seguimiento y pago de ella. Para poder cumplir con las expectativas, debe contar con las siguientes características:

### Descripción del trabajo a entregar

1. Su grupo deberá modificar el modelo E/R de la E1 para adaptarse a los nuevos requerimientos
2. Traspasar el modelo Entidad/ Relación al modelo relacional especificando tanto el esquema, el tipo y dominio de cada atributo, incluyendo llaves primarias y foráneas así como restricciones de integridad. Puede que se necesiten hacer algunos ajustes a tu modelo relacional para normalizarlo. El modelo final es el que debe ser entregado en el reporte.
3. Debe mostrar que tu modelo está en BCNF. Si no se puede, entonces deben mostrar que esta en 3NF. Para ello debe listar las dependencias funcionales no triviales de cada relación y mostrar que está todo normalizado.
4. Como en todo sistema real las claves se manejan encriptadas, sin embargo en el set de datos vienen en texto plano. Para la carga inicial de datos deberán encriptarlos y en la aplicación, tanto el ingreso, actualización y comparación de las claves a la base se deben manejar encriptadas.
5. La carga de datos debe ser automatizada (scripts en Python), sin ningún tipo de procesamiento manual.
6. En la carga inicial de datos se deberá chequear la integridad de los mismos y cuando haya falla tomar la acción más idónea (corregir o eliminar) la cual debe ser justificada en el informe.
7. Finalmente, la entrega consiste en una base de datos, un conjunto de programas en Python (cargadores y aplicación Easyfood) y un informe. Tanto la Base de datos como los programas se entregan en el la cuenta del grupo en el servidor, mientras que el informe se entrega en Canvas. el informe deberá contener el modelo E/R el esquema relacional, las instrucciones de ejecución de los cargadores, las violaciones a la integridad y la solución aplicada; las instrucciones para la ejecución de Easyfood.

## Funcionalidades a implementar

El grupo deberá implementar las siguientes funcionalidades

1. Cargadores de datos python: Los datos deberán ser cargados automáticamente, desde los archivos de datos proporcionados. sin pre-procesamiento previo.
2. Implementar una página con las consultas presentadas a continuación
3. Implementar una página de consulta inestructurada "SELECT" que reciba 3 atributos: Atributos A, tablas T y condiciones C y la ejecute de la siguiente forma: `SELECT A FROM T WHERE C`. Tanto A,T y C son string, A y T no nulos. Ud debe verificar que el formato de A,T y C sean correctos antes de ejecutar la consulta.

## Consultas Sql a implementar

En esta sección se detallan las consultas a implementar. Estas consultas pueden tener un input dado que se realizará mediante una página web.

1. Muestre todos los restaurantes que ofrezcan un plato específico (string) ingresado y que esté disponible.
2. Muestre todos los pedidos de un usuario específico ingresado (email string) y su gasto mensual (solo los pedidos concretados)
3. Muestre todos pedidos concretados y cancelados y el valor total de ellos
4. Dado un estilo de plato ingresado por el usuario (string) , muestre todas los platos con ese tipo, los restaurantes que las ofrecen y las opciones de delivery.
5. Dado un estilo de plato ingresado (string) por el usuario, muestre todas las platos que pertenezcan a ese estilo y sus restricciones.
6. Dado un cliente ingresado por el usuario (email string), muestre todas las restaurantes a las que tiene acceso con sus suscripciones.
7. Muestre la suma de dinero gastada por cada cliente en pedidos no incluidas en planes de suscripción.
8. Muestre todos los platos y los restaurantes que los ofrecen.
9. Dado un número (1-5)ingresado por el usuario, muestre todos las evaluaciones de Clientes, Delivery, Despachador superiores o iguales a él.
10. Dado una un alergeno ingresado por el usuario (string), muestre todos los platos que lo contengan en sus ingredientes (Ejemplo: maní).

## Reglas de negocio

- Los Restaurantes pueden tener varias Sucursales.
- Los Restaurantes/Sucursal pueden tener Delivery propio.
- Los Platos que se llaman igual pueden ser diferentes en cada Restaurant
- Los Restaurantes/Sucursal pueden tener solo algunos Delivery.
- Los Restaurantes que tienen despacho usan a los mismos Despachadores de éstos.

- Los Despachadores pueden atender a múltiples Delivery y Restaurantes/sucursales.
- Cada Despachador solo puede atender máximo 2 pedidos del mismo Restaurant a la misma comuna.
- El Despachador califica al cliente, el cliente califica el Pedido. La calificación del pedido se aplica al Delivery y Despachador
- El tiempo de despacho es la suma del máximo tiempo de preparación de los platos y el máximo tiempo de despacho.
- El Cliente tiene una dirección de correo única un teléfono único pero varias direcciones
- Una Suscripción se considera cancelada si es que el cliente la cancela explícitamente o no paga la cuota mensual, en ambos casos el servicio se presta hasta el día del próximo pago. En caso de las renovaciones anuales no se devuelven los meses pagados por adelantado.
- Las claves de los clientes vienen en texto plano en los archivos de datos, sin embargo al cargarse en el sistema y de ahí en adelante se manejan encriptadas.

## Diccionario de Datos

- - Nombre Restaurant: string (30 caracteres, no nulo, único)
  - Restaurante vigente Boolean no nulo
  - Restaurante estilo string no nulo
  - Preci o m nimo reparto gratis: int (nulo= no tiene)
  - Sucursal: string
  - Direccion sucursal: string (30 caracteres, no nula, calle n mero y comuna)
  - Tel fono sucursal: string (formato +PP y 9 d gitos, no nulo)
  -  rea de despacho: string comunas de despacho
  - Nombre Plato: string (30 caracteres, no nulo)
  - Estilo Plato: string (30 caracteres, no nulo)
  - Restriccion Plato: int (vegano, vegetariano, alerg enos, etc.)
  - Descripcion Plato: string (30 caracteres, no nulo)
  - Ingredientes Plato: string (30 caracteres)
  - Disponibilidad Plato: boolean no nulo
  - Porciones: int (min 1 default 1)
  - Precio Plato: int (no nulo)
  - Tiempo de Preparacion plato (minutos): int (1-60, default 5, no nulo)
  - Nombre Cliente: string (30 caracteres, no nulo)
  - email Cliente: string (30 caracteres, no nulo,  nico)
  - Tel fono Cliente: string (formato +PP y 9 d gitos, no nulo)
  - Clave Cliente: string (encriptada, no nula)
  - Direccion: string (30 caracteres, no nula, calle n mero y comuna) el cliente puede tener varias direcciones.
  - Nombre Delivery: string (30 caracteres, no nulo)

- Delivery vigente Boolean no nulo
- Telefono Delivery: string (formato +P y 9 dígitos, no nulo, único)
- Tiempo de despacho: int (minutos, no nulo)
- Precio por unitario despacho: int
- Precio por suscripción mensual:  $\text{int} \leq 4 * \text{preciodespacho}$
- Precio por suscripción anual:  $\text{int} \leq 12 * \text{preciomensual}$
- Suscripción monto ultimo pago: int (debe corresponder a uno de los valores definidos anteriormente)
- Suscripción fecha ultimo pago: date
- Estado suscripción: Vigente, Cancelada
- Ciclo facturación: string (Mensual/Anual, no nula, debe coincidir con el precio correspondiente)
- Nombre Despachador: string (30 caracteres, no nulo)
- Teléfono Despachador: string (formato +PP y 9 dígitos, no nulo, único)
- Fecha pedido: date (no nula)
- Hora pedido: time (incluye TMZ)
- Estado pedido: string (pendiente, en preparación, entregado a despachador, entregado a cliente, Cliente cancela, delivery cancela, restaurant cancela)
- Calificación al Cliente: int (1-5)
- Calificación del pedido: int (1-5)

## **Entrega 2, domingo 26 de mayo 23:59, consultas hasta viernes 24 de mayo a las 18:00**

El informe4 debe identificarse con el nombre grupoXXX.pdf donde XXX es el número de grupo asignado y subido solo por uno de los integrantes del grupo a canvas. Las consultas se reciben por ISSUES de Github, con el label ENTREGA 2. Los datos de conexión al server y archivos de datos se anunciarán por Canvas.