

# HACKA{ODE}



BY TODO{ODE}

## 2da Edición - Escenario

### Consigna

Una **agencia de turismo** necesita del desarrollo de una **aplicación web** para la venta de sus productos por parte de cada uno de sus empleados.

Para comprender cada parte de la empresa y cada uno de sus productos y servicios que ofrece, un **analista funcional** llevó a cabo el relevamiento de cada una de las acciones y cuestiones a tener en cuenta en el desarrollo del sistema, éstas se citan a continuación.

### Servicio Turístico

- La agencia de turismo ofrece diferentes tipos de servicios a sus potenciales clientes:
  - Hotel por noche/s
  - Alquiler de auto
  - Pasajes de colectivo
  - Pasajes de avión
  - Pasajes de tren
  - Excursiones
  - Entradas a Eventos
- Cada servicio puede ser contratado de dos maneras posibles:
  - De forma individual
  - En un paquete turístico (con otros servicios)
- Cada servicio tiene los siguientes datos:
  - `codigo_servicio`
  - `nombre`
  - `descripción_breve`
  - `destino_servicio`
  - `fecha_servicio`
  - `costo_servicio`

### Paquete Turístico

- Un paquete turístico es la combinación de dos o más servicios turísticos. Por ejemplo: un paquete puede contener una noche de hotel + un pasaje de avión.
- Los paquetes turísticos tienen los siguientes datos:
  - `codigo_paquete`
  - `lista_servicios_incluidos`
  - `costo_paquete`



@todocode



@todo\_code



@todo\_code



youtube.com/ToDoCode

# HACKA {ODE}

BY TODO {ODE}



## 2da Edición - Escenario

- Cabe destacar que el costo del paquete es igual a la suma de los costos de los servicios que lo componen menos un 10% de descuento por contratarlos en forma de paquete. De esta manera, por ejemplo, si la noche de hotel vale \$6000 y el pasaje de avión \$45.000 entonces la sumatoria de ambos es \$51.000, a esto, al descontar el 10% obtenemos el valor total del paquete ( $\$51.000 - 10\% = \$45.900$ ).

### CLIENTES:

- El sistema debe ser capaz de llevar a cabo el registro de nuevos clientes. De cada cliente se necesitan los siguientes datos:
  - id\_cliente
  - nombre
  - apellido
  - dirección
  - dni
  - fecha\_nac
  - nacionalidad
  - celular
  - email

### EMPLEADOS:

- El sistema debe ser capaz de llevar a cabo el registro de cada uno de sus empleados.
- Cada empleado deberá contar con los mismos datos que un cliente, agregando además también los siguientes datos propios:
  - Cargo
  - Sueldo

### VENTAS:

- El sistema debe ser capaz de llevar a cabo ventas de diferentes paquetes o servicios turísticos.
- Cada venta cuenta con los siguientes datos propios:
  - num\_venta
  - fecha\_venta
  - medio\_pago
- Al mismo tiempo, cada venta debe estar relacionada con:
  - Un cliente
  - Un servicio o un paquete turístico (recordar que un paquete contiene varios servicios).
  - Un empleado (el que realiza la venta)



@todocode



@todo\_code



@todo\_code



youtube.com/ToDoCode

# HACKA{ODE}

BY **TODO {ODE}**

## 2da Edición - Escenario

### ABML (Altas, bajas, modificaciones y lecturas necesarias)

- El sistema debe ser capaz de realizar las operaciones ABML de los diferentes servicios, paquetes, clientes, ventas y empleados. Para ello tener en cuenta que idealmente:
  - Todos los clientes, servicios, paquetes y ventas deberán poder tener su ventana o apartado para realizar altas.
  - Todos los clientes, servicios, paquetes y ventas deberán poder tener su ventana o apartado para realizar modificaciones.
  - Todos los clientes, servicios, paquetes y ventas deberán poder tener su ventana o apartado para realizar consultas/lectura de datos.
  - Todos los clientes, servicios, paquetes y ventas deberán poder tener su ventana o apartado para realizar bajas o eliminaciones.

### ENTREGABLES:

- **Diagrama de clases** del modelo de datos (UML) de la aplicación a desarrollar.
- **Repositorio** de GitHub del **Frontend**.
- **Repositorio** de GitHub del **Backend**.
- Especificación de las configuraciones de la **base de datos** (usuarios, contraseñas, etc para replicar)
- **Aplicación deployada** y en funcionamiento (tanto back como front).
- **Documento de supuestos**: Un documento con todas las suposiciones que tuvo el analista/programador. Por ejemplo, se supone que cada paquete es comprado por un solo cliente.

### SE VALORARÁ:

- **Interfaz gráfica** atractiva e interactiva.
- Correcta **funcionalidad** y **visualización de los datos** que se muestren por pantalla.
- **Eficiencia** del modelado lógico a partir de UML.
- **Creatividad**, aportes, ideas o funcionalidades extras proporcionadas por los miembros del equipo.
- **Uso de buenas prácticas**: código comentado, separación de la aplicación en capas, documentación, README, etc.



@todocode



@todo\_code



@todo\_code



youtube.com/ToDoCode

# HACKA {ODE}

BY **TODO {ODE}**

## 2da Edición - Escenario

### BONUS POINT

- El sistema que se planteó con anterioridad no tiene pensado el manejo ni de dinero ni de ganancias; sin embargo, el dueño de la agencia propuso una recompensa para aquel desarrollador que realice una propuesta que le permita, a través del sistema, conocer las ganancias diarias y mensuales en base a las ventas realizadas de los paquetes y servicios.
- Para ello tener en cuenta que las comisiones según medio de pago son:
  - Efectivo: Sin comisión
  - Tarjeta de Débito: 3%
  - Tarjeta de Crédito: 9%
  - Monedero Virtual: Sin comisión
  - Transferencia: 2.45%



@todocode



@todo\_code



@todo\_code



youtube.com/ToDoCode