

**ESCUELA DE INGENIERÍA**  
**DEPARTAMENTO DE COMPUTACIÓN, ELECTRÓNICA Y MECATRÓNICA**

**BASES DE DATOS**  
**LIS-2082**

**PROYECTO FINAL**  
**AGENCIA DE VIAJES PONCHITO**

## **1. Objetivo**

La agencia de viajes PONCHITO desea informatizar su actividad "reservación de viajes" utilizando un SGBD relacional. Para ello, esta agencia dispone de un folleto automatizado que permite simular y reservar un viaje. El folleto se presenta bajo la forma de relaciones que agrupan la información sobre ciudades, circuitos, monumentos, museos, hoteles, etc. También se debe definir las relaciones para autorizar, a un cliente potencial de la agencia, realizar una simulación de viaje y dirigirse posteriormente a la agencia para validar las opciones de su simulación.

La aplicación debe proveer funciones que permitan consultar, modificar y agregar información a esta base. Se debe, por ejemplo, poder aceptar nuevos clientes y autorizar una simulación de viaje después de su validación.

## **2. Definición del esquema de la base**

La información de la aplicación será manejada por una base de datos relacional. Una parte del esquema relacional de esta base estará predefinido. Usted no debe modificar los esquemas de las relaciones que se le proporcionan, pero puede agregar nuevas relaciones para considerar la información adicional necesaria a la implementación de la aplicación.

### ***Esquema proporcionado***

La parte del esquema relacional proporcionado corresponde al catálogo de los viajes (folleto). Esta parte concierne la información sobre las posibilidades de viajes ofrecidos por la agencia PONCHITO. Aquí se encuentran las relaciones que describen los países y las ciudades considerados por la agencia así como los lugares a visitar (monumentos, museos, etc.), los circuitos y los hoteles propuestos en las ciudades. Las relaciones son:

*Ciudad*( nombre char(12), país char(12) )

Una ciudad tiene un nombre y pertenece a un país. Se supone que no hay dos ciudades con el mismo nombre en un país dado.

*LugarAvisitar*( nombre char(12), ciudad char(12), país char(12), dirección char(20),  
descripción char(40), precio int )

Un lugar (un sitio, un monumento, un museo) a visitar tiene un nombre y una descripción. Se encuentra en una *dirección* en una *ciudad* de un *país* y su visita cuesta cierto *precio*. En el caso de algunos lugares el precio puede ser igual a 0.

*Circuito*( identificador char(5), descripción char(20), ciudadSalida char(12),  
paísSalida char(12), ciudadLlegada char(12), paísLlegada char(12),  
duración int, precio int )

Un circuito tiene un *identificador* y una *descripción*. Se realiza entre una ciudad de salida (en un país de salida) y una ciudad de llegada (en un país de llegada), tiene una *duración* y cuesta un *precio*.

*FechaCircuito*( *identificador char(5)*, *fechaSalida date*, *nbPersonas int* )

Un mismo circuito puede ser propuesto en fechas diferentes (*fechaSalida*) para un cierto número de personas.

*Etapas*( *identificador char(5)*, *orden int*, *nombreLugar char(12)*, *ciudad char(12)*, *país char(12)*, *duración int* )

Un circuito corresponde a una lista de lugares a visitar (etapas) en cierto *orden*. Cada etapa hace referencia al lugar que se visita en una *ciudad* y tiene una *duración*.

*Hotel*( *nombre char(12)*, *ciudad char(12)*, *país char(12)*, *dirección char(20)*, *numCuartos int*, *precioCuarto int*, *precioDesayuno int* )

Un hotel está situado en una dirección en una *ciudad* y tiene *numCuartos* cuartos con un costo *precioCuarto*. Propone un desayuno a un precio *precioDesayuno*.

### ***Esquema a definir***

Con la información que tiene, la agencia PONCHITO puede proponer a sus clientes diferentes simulaciones de viaje. Una simulación incluye:

- el nombre de la persona (cliente conocido o potencial) que pide la simulación,
- las fechas de salida y de llegada consideradas para el viaje,
- el número de personas que participan en el viaje,
- las diferentes opciones de reservación tomadas por el cliente al examinar la información sugerida, es decir el país considerado para el viaje, las ciudades a visitar, los lugares, circuitos y hoteles seleccionados,
- el costo del viaje (calculado).

Después de validar una simulación, el usuario obtiene un número de simulación con el cual puede presentarse en la agencia para realizar la reservación del viaje. Para hacer la reservación, es necesario entonces validar la información obtenida a la hora de la simulación (comprobar que las fechas y las opciones de la reservación son siempre válidas), realizar las reservaciones con los organismos correspondientes.

La agencia administra en su base de datos las reservaciones de cada cliente. Para dar de alta una reservación, se tendrá que dar de alta al cliente en la base, si éste no existe todavía. Un cliente se registra con su nombre, un tipo (compañía, grupo, individual) y el año de registro. Sobre el conjunto de los clientes, no pueden existir clientes sin reservación. Un cliente puede ser empleado de la agencia (usuario de la categoría Agencia) y se beneficia entonces de una reducción particular en la cantidad de su viaje.

Es importante notar que, pasado un plazo de dos días, se considera que una simulación ya no es válida: la información de la simulación almacenada en la base se destruye.

Para completar el esquema relacional dado en la sección 2.1, usted debe crear las relaciones necesarias para la administración de la información relativa a las simulaciones y a las reservaciones.

### 3. Funciones de la aplicación

La aplicación debe proporcionar las funciones necesarias para la gestión de las simulaciones y de las reservaciones así como de ciertas funciones de administración. A continuación se describen las funciones esenciales.

#### *Consulta del folleto*

Se pide que usted desarrolle el(los) programa(s) de la aplicación PONCHITO que permitan obtener información sobre los viajes sugeridos por la agencia.

Se trata de permitirle a un usuario autorizado (verificación de derechos de acceso) obtener la información sobre los países a visitar, las ciudades en los países, los circuitos en las ciudades, los hoteles en las ciudades, etc.

Las funciones principales de este programa son las siguientes:

- Permitir que el usuario seleccione el tipo de información que desea ver.
- El programa solicita al usuario que especifique la información necesaria para consultar la base. Inicialmente se limitará a consultas expresadas a través de mecanismos de diálogo de tipo texto.

#### *Simulación de una reservación de viaje*

El programa de la simulación debe permitir simular una o más reservaciones de viaje para un usuario y validar simulaciones. Las funciones principales de este programa son las siguientes:

- Capturar la información del usuario: mientras que el usuario no haya reservado, se le considera como cliente potencial. La información mínima que le concierne será almacenada en la base con sus simulaciones. Para simplificar, sólo se le pedirá su nombre. Esta información permitirá encontrar las simulaciones de un cliente.
- Crear una nueva simulación de viaje mientras que "navega" en el folleto (se pueden ejecutar las funciones de consulta del folleto). Para simplificar, se podrá considerar que una simulación concierne a un país con una o varias ciudades del mismo. Para cada ciudad, se puede obtener los lugares a visitar así como los hoteles. Se puede también elegir los circuitos que salen de cada ciudad. A partir de una información (un lugar, un circuito, un hotel) el usuario puede elegir e incluirla en la simulación.
- Evaluar el costo de un viaje (el costo de una simulación).
- Presentar la información relacionada con su simulación de viaje.
- Validar una simulación: la aplicación regresa el número de la simulación. Este número es asociado al nombre del cliente y a la fecha actual (se supone que no pueden existir dos clientes con el mismo nombre). Este será utilizado por el cliente a la hora de ir a efectuar una reservación en la agencia con base en una simulación previamente definida.

#### *Reservación de un viaje*

El programa reservación permite confirmar una simulación de viaje. Las funciones principales de este programa son las siguientes:

- Capturar la información sobre el cliente: si la persona que se presenta en la agencia no es un cliente conocido, el programa debe entonces pedirle información adicional (su nombre, su dirección, tipo de pago, etc.). Si al contrario, la persona es un cliente conocido, el programa debe cerciorarse de que la información almacenada referente a él esté actualizada.
- Crear una nueva reservación de viaje a partir de una simulación: se trata de la lista de reservaciones de circuitos, de lugares a visitar, de hoteles, de tours, etc. para validar cada opción. Por ejemplo, en el caso de indisponibilidad de ciertos hoteles o circuitos (falta de cuartos, exceso de participantes, etc.) el cliente tendrá que elegir otras opciones o cancelar todo. De manera general, es importante asegurar la coherencia de la información de la reservación.
- Evaluar el costo de un viaje y pedir la información necesaria para el pago.
- Presentar la lista de las reservaciones.

- Validar una reservación, la aplicación regresa un número de la reservación.

### ***Interfaces (opcional)***

Para la aplicación PONCHITO se pueden distinguir tres interfaces principales, cada una relacionada con un tipo de usuario:

- *Interfaz Pública:* permite que un usuario de tipo público (no conocido) consulte la información relacionada con los viajes sugeridos por la agencia y simule una reservación.
- *Interfaz Cliente:* permite que un usuario de tipo cliente (conocido) consulte sus reservaciones. Una persona cliente de la agencia tiene por lo menos una reservación en la agencia.
- *Interfaz Agencia:* permite a una persona que trabaja en la agencia (1) consultar el folleto y las reservaciones del cliente; (2) crear clientes; y (3) realizar reservaciones de viajes (simular y/o validar reservaciones). Tiene el derecho además de modificar las reservaciones ya validadas.
- *Interfaz Global:* agrupa las interfaces descritas arriba. Para utilizar las funciones de la aplicación, el usuario tendrá que acceder al sistema como usuario público, cliente o trabajador. Por defecto cualquier usuario (conocido o no) debe poder acceder la interfaz pública. Sin embargo, si un cliente desea utilizar la interfaz cliente para obtener la información sobre sus reservaciones de viaje, entonces debe conectarse como usuario de tipo cliente. Para ello, es necesario estar registrado en la base con la categoría *Cliente*. De manera similar, para que un usuario tenga acceso a la interfaz *Agencia*, debe estar registrado en la base con la categoría correspondiente. Note que se debe asociar una palabra clave a los usuarios a la hora de su creación.

## **4. Para concluir...**

Se precisa que la aplicación debe asegurar la coherencia de la base de datos. Por lo tanto, piense en definir transacciones y tanto como sea posible utilizar los servicios de apoyo del SGBD, para asegurar la coherencia (restricciones de integridad). Para el desarrollo de su aplicación, se debe dar prioridad a los aspectos relacionados con la administración de la base de datos. Se tiene toda libertad para los aspectos de interfaz, pero estos **no son el punto central del proyecto**. Una buena interfaz permitirá obtener puntos extras solamente si todos los aspectos de bases de datos se trataron bien.

Por otra parte, se debe ser riguroso al concebir el esquema de la base de datos:

- construir un esquema conceptual Entidad-Relación,
- transformar este esquema a un esquema relacional en 3FN,
- especificar las restricciones de integridad (restricciones de dominios, llaves primarias y extranjeras) e implementarlas!

## 5. Evaluación

Entregar un reporte técnico en el que se debe incluir la información señalada en los siguientes puntos:

- Su habilidad para **aplicar el diseño de ingeniería para producir soluciones** que satisfagan necesidades específicas teniendo en cuenta la salud pública, la seguridad y el bienestar, así como factores globales, culturales, sociales, ambientales y económicos. Para organizar la información previa, el reporte (10 página máximo) debe tener secciones en las que se detallen los siguientes aspectos:
  - Título del proyecto.
  - Breve descripción.
  - Análisis de viabilidad del proyecto.
  - Objetivos, metas e hipótesis.
  - Especificaciones técnicas:
    - Dar el esquema conceptual ER o UML de la base de datos.
    - Dar en un ANEXO su traducción a esquema lógico usando SQL.
    - Describir las funciones de su sistema (especificación de las transacciones).
  - Normativa y reglamentos para el desarrollo del software, ver ISO/IEC 12207:2008.
  - Metodología utilizada para el desarrollo.
  - Resultados obtenidos.
  - Conclusiones.
  - Referencias y bibliografía.
- Su habilidad para adquirir y **aplicar nuevos conocimientos según sea necesario**, utilizando estrategias de aprendizaje adecuadas. Agregar a su reporte otro ANEXO en el que se describan y detallen los siguientes puntos:
  - *Gap awareness*: Identificar el conocimiento técnico o de diseño de sistemas que le faltaba para abordar la práctica y que tuvo que desarrollar para desarrollarla. De qué manera el nuevo conocimiento le permitió mejorar sus habilidades para diseñar e implementar la base de datos.
  - *Growth awareness*: ¿Qué habilidades técnicas y de diseño de sistemas que ya había adquirido anteriormente usó para desarrollar la práctica? ¿Cuál fue su estrategia para desarrollar nuevas habilidades o qué habilidades tuvo que mejorar para desarrollar el proyecto?
  - *Research*: ¿Qué fuentes, tutoriales, guía de patrones de diseño de sistemas, guías técnicas usó para apoyar el desarrollo del proyecto?
  - *Apply knowledge*: Explicar cómo está aplicando la teoría y la práctica de la filosofía de las bases de datos para resolver su problema de manera precisa y efectiva.

Ajuntó encontrarán las rúbricas que serán utilizadas para evaluar cada punto:

- “SO2.pdf” (aplicar el diseño de ingeniería para producir soluciones) y
- “SO7.pdf” (aplicar nuevos conocimientos según sea necesario).