

Módulo: Programador

Proyecto ABP

Introducción al proyecto

El **proyecto final** será compuesto de dos etapas, una correspondiente a la **evidencia 2** cuya consigna es detallada en este documento, y una siguiente etapa que será parte de la **evidencia 3** (consigna aún no publicada). La resolución de cada evidencia es especificada en detalle en el apartado de “Entregables” que posee cada consigna de evidencia. El trabajo desde evidencia 2 en adelante es grupal, si bien pueden dividirse tareas, cada uno debe saber de la totalidad de lo abordado en cada consigna.

Algunas fechas importantes:

- **Evidencia 2** tiene como fecha de entrega el **martes 27/5** a las 23:59hs.

Una vez presentada la consigna de la Evidencia 3, con ella también se compartirá el modelo del documento que deben elaborar, y que será parte de la entrega final.

- **Evidencia 3** tiene como fecha de entrega el **miércoles 04/6** a las 23:59hs. **Su consigna aún no es presentada.**
- **Proyecto final ABP: Documento** (basado en el modelo de proyecto abp) que deben elaborar, y que será parte de la entrega final (junto con las **correcciones de las evidencias** el **lunes 16/6** a las 23:59hs). Además del documento y las correcciones de las evidencias, se debe entregar el **póster** que será utilizado para la defensa del proyecto.
- **Defensa oral del Proyecto ABP** será el día **miércoles 18/6** en los horarios de sincrónico previa distribución de horarios (a confirmar).

Veamos ahora de que se trata la temática del proyecto:

Situación Profesional

Sistema básico de gestión de pasajes aéreos

La empresa ficticia SkyRoute S.R.L., una agencia que comercializa pasajes aéreos para empresas y particulares, busca implementar un sistema digital básico que le permita gestionar la venta de pasajes, el registro de clientes, los destinos disponibles y una funcionalidad especial conocida como botón de arrepentimiento, la cual permite a los usuarios anular una compra dentro de un período limitado desde su realización.

La empresa ha contratado al equipo de estudiantes como desarrolladores para construir un prototipo de aplicación por consola, simple pero funcional, que pueda simular el funcionamiento del sistema de ventas.

El proyecto estará basado en una aplicación de escritorio (consola) python que utilizará base de datos relacional y que integrará los saberes de todo el Módulo programador. La aplicación se presentará al usuario mediante un menú de opciones que permitirá resolver lo requerido por la empresa ficticia.

Objetivo Principal

Desarrollar una aplicación de escritorio en Python que permita:

1. Registrar usuarios de empresa (clientes).
2. Registrar destinos disponibles.
3. Registrar la venta de pasajes asociando cliente, destino, fecha y costo.
4. Permitir visualizar ventas y realizar consultas simples.
5. Implementar la función de “botón de arrepentimiento” para anular una compra reciente.

Requerimientos generales funcionales del sistema

Gestión de Clientes:

Permitir registrar nuevas empresas como clientes, ingresando datos como razón social, CUIT y correo de contacto. También se podrá modificar o eliminar clientes registrados.

Gestión de Destinos:

Permitir registrar destinos disponibles para la venta de pasajes. Cada destino debe incluir ciudad, país y costo base del viaje. Se debe poder modificar o eliminar destinos.

Gestión de Ventas:

Permitir registrar ventas de pasajes, asociando un cliente con un destino y registrando la fecha de la venta. Cada venta tendrá un estado ("Activa" o "Anulada").

Botón de Arrepentimiento:

Permitir al usuario anular una venta recientemente realizada mediante la funcionalidad de "botón de arrepentimiento", que dejará la venta en estado "Anulada". Este proceso simula el derecho a cancelar la compra dentro de un período determinado. Enmarcada en la Ley 24.240 de Defensa al Consumidor y en el Código Civil y Comercial de la Nación (Ley 26.994), la medida tiene como objetivo facilitar la devolución y reembolso de las transacciones. Debe incluirse en los servicios digitales de todos los proveedores de productos y servicios. Acorde a la regulación y para facilitar el proceso de devolución, el plazo que otorga la ley para dar marcha atrás en las operaciones aéreas es de 60 días hábiles.

Consulta de Datos:

Permitir visualizar:

- La lista de clientes registrados.
- Los destinos disponibles.
- Las ventas realizadas, filtradas por cliente, destino o estado.
- Ventas anuladas mediante el botón de arrepentimiento.

Requerimientos no funcionales

Modularidad: (en evidencia 3)

El código fuente debe estar organizado en **módulos y funciones independientes**, de forma que se facilite la escalabilidad, el mantenimiento y la reutilización del código.

Legibilidad:

El código debe utilizar nombres claros y coherentes para variables, funciones y estructuras. Debe incluir comentarios que expliquen las partes clave del sistema.

Eficiencia:

El sistema debe ser eficiente en el uso de recursos, evitando estructuras innecesarias o repetitivas. La manipulación de datos en memoria (diccionarios, listas) debe ser ágil.

Usabilidad:

La interfaz de consola debe ser **intuitiva y clara**, organizada con menús y submenús pensados para un flujo natural del usuario. Cada acción debe estar acompañada de mensajes que guíen y confirmen las operaciones realizadas.

Evidencia de aprendizaje N° 2 - Consigna

A partir de la **Evidencia de Aprendizaje N.º 2**, se inicia la construcción de un primer prototipo de aplicación de consola en Python, estructurado bajo el paradigma imperativo y utilizando estructuras básicas de programación. Este prototipo tiene como finalidad establecer las bases funcionales del sistema, ofreciendo un menú interactivo que simula el funcionamiento deseado. Paralelamente, se diseña un modelo relacional de base de datos que organiza de manera coherente toda la información relevante para la empresa.

Programación

Desarrollar un primer prototipo de la aplicación de consola, en un único archivo python (**main.py**), utilizando solo estructuras básicas de programación: estructuras secuenciales, condicionales e iterativas.

Contenidos de Programación Aplicados:

- Paradigma Imperativo: programas secuenciales.
- Estructuras de control: condicionales (**if/else**) y ciclos (**for, while**).
- Variables, constantes, entrada (**input**) y salida(**print**)
- (**opcional**) Control de versiones: Git y GitHub (creación de repositorio, push inicial).

Alcance funcional en esta evidencia:

- Implementar un **menú de opciones** interactivo, por ejemplo (NO necesariamente debe ser así):

Bienvenidos a SkyRoute - Sistema de Gestión de Pasajes

1. Gestionar Clientes
2. Gestionar Destinos
3. Gestionar Ventas
4. Consultar Ventas
5. Botón de Arrepentimiento
6. Ver Reporte General
7. Acerca del Sistema
8. Salir

y algunas opciones que tengan **submenú** como puede ser:

-- GESTIONAR CLIENTES --

1. Ver Clientes
2. Agregar Cliente
3. Modificar Cliente
4. Eliminar Cliente
5. Volver al Menú Principal

- Al elegir una opción deberá mostrar un mensaje representativo mediante prints. (No debe haber aún funciones ni módulos externos). Por ejemplo, una navegación posible al ejecutar `main.py` podría mostrar (mediante prints) lo siguiente:
 - *Bienvenidos a **SkyRoute - Sistema de Gestión de Pasajes** (y el menú de opciones)*
 - *Ingreso a la opción 1: Gestionar Cliente*
 - *eligio la opción 2 del submenú (Gestión Cliente) agregar cliente*
 - *ingreso los siguientes datos para el nuevo cliente:*
 - *Se guardó el cliente, sus datos son:*
 - *Volvió al menú principal*
 - *Ingreso a la opción 4: Consulta ventas*
 - *Se mostraron las ventas*
 - *Muestra un submenú con opciones de consultas.*
 - *Eligio la opción 3: Ventas de la última semana*

- Se mostraron las ventas de la última semana y luego volvió al menú principal
- Eligió salir (y sale de la aplicación)

Estructura mínima requerida:

- Carpeta de proyecto con:
 - (opcional) Repositorio Git inicializado y vinculado a GitHub.
 - Archivo `main.py` que contenga:
 - Un encabezado con:
 - El propósito del sistema.
 - Cómo instalar y ejecutar el programa.
 - Datos de los integrantes del grupo (nombre, apellido y DNI).
 - El menú de opciones.
 - Control de flujo de acciones a realizar según la opción seleccionada (sin modularización).

Base de Datos

1. Identificar las entidades principales necesarias para almacenar la información del sistema de Gestión de Pasajes (ej. Cliente, Ventas, Destinos, etc).
2. Definir los atributos relevantes para cada entidad, especificando el tipo de dato adecuado para cada uno (ej. texto, número entero, número decimal, fecha/hora).
3. Establecer las relaciones entre las entidades identificadas (uno a muchos, muchos a muchos), especificando las claves primarias y foráneas necesarias.
4. Crear el **diagrama entidad relación** completo, representando gráficamente las tablas, sus atributos, la cardinalidad y la dependencia (identidad, no identidad y generalización/especialización). Utilizar herramientas como draw.io, lucidchart u otras similares.

Ética y Ejercicio Profesional

1. El informe deberá considerar:
2. Que figura legal tendrá el grupo?
3. Explicar como se encuadrará legalmente la relación laboral entre el grupo y SkyRoute S.R.L.

4. Si SkyRoute S.R.L. decide cambiar de grupo para que le provea la aplicación, cómo debe actuar cada integrante del grupo?
5. Propiedad de quien son los datos que se comienzan a diagramar?
6. Propiedad de quien es el código desarrollado?
- 7.Cuál es la utilidad del Botón de Arrepentimiento?
8. Fundamentar cada punto del informe con los conceptos legales que hemos incorporado

Entregables

1. Programación I.

Link al repositorio GitHub con el proyecto (o entregar el archivo `main.py`).

2. Base de Datos.

Documento que contenga el diagrama del modelo entidad relacional generado.

3. Ética y ejercicio Profesional.

Documento que contenga el informe solicitado con referencias normativas claras y una sección final de conclusiones personales vinculadas al caso.

4. Nos adelantamos al Informe Final. Inicio de la elaboración del Documento Final del Proyecto.

Siguiendo la "Estructura para el diseño de Proyectos ABP" provista (Documento que pronto será subido en el mosaico Proyecto Final). Para esta entrega, un avance sugerido del informe es incluir los siguientes puntos **completados o en borrador inicial**:

- **5. Problemáticas/Necesidades:** Identificación y descripción inicial de la problemática a abordar. La misma surge de la situación profesional y el enunciado de la evidencia de aprendizaje 2.
- **6. Fundamentación:** Un borrador inicial explicando la importancia de abordar esta problemática/necesidad identificada
- **10. Producto Final:** Una descripción inicial de la parte del producto final que están construyendo en esta etapa.

Evidencia de aprendizaje N° 3

Se habilitará la consigna más adelante.

Documento del Proyecto Final ABP

Se habilitará más adelante el documento modelo con todos los puntos a considerar.