Desarrollo de Aplicaciones Multiplataforma

Actividad 1.

Reserva *Vuelos*

Desarrollo de Interfaces

Actividad

Interfaz *Reserva de Vuelos*.

**Objetivos**

* Usar controles básicos y avanzados para lograr los objetivos.
* Crear interfaces intuitivas, simples y adecuadas.

|  |
| --- |
| **¿Cómo lo hago?** |
| 1. Rellena los datos que se piden en la tabla “Antes de empezar”. 2. Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc. 3. Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar. 4. Recuerda entregar la actividad en formato PDF a no ser que el profesor o profesora indique lo contrario. 5. Recuerda nombrar el archivo siguiendo estas indicaciones:  * Ciclo\_Módulo o crédito\_Tema\_ACT\_número actividad\_Nombre y apellido   + Ejemplo: AF\_M01\_T01\_ACT\_01\_Maria Garcia |

|  |  |
| --- | --- |
| **Antes de empezar…** | |
| Nombre |  |
| Apellidos |  |
| Módulo/Crédito |  |
| UF (solo ciclos LOE) |  |
| Título de la actividad |  |

Actividad 1

***Interfaz de un programa de Reserva de Vuelos***

La actividad consiste en implementar la interfaz para un programa permite reservar billetes de avión. La interfaz debe permitir al usuario filtrar el listado de vuelos de la manera mas eficiente (salida, llegada, fecha, compañía, …) para seleccionar el vuelo deseado. Una vez elegido el vuelo se procederá a realizar la reserva. Para ello el usuario debe elegir la cantidad de billetes de adulo y de niño, así como la opción para equipaje extra (tipo si o no). Una vez obtenidos todos los datos deberá confirmarse la reserva.

**Modelo de Datos:**

Para ello se dispone de el siguiente modelo de datos:

|  |
| --- |
| **Aeropuerto** |
| Nombre : String  Siglas : String  Nacional : bool |
| importar(String) : List<Aeropuerto>  exportar(String) |

|  |
| --- |
| **Aerolinea** |
| Nombre : String  Codigo : String  Logo : Bitmap |
| importar(String) : List<Aerolinea>  exportar(String) |

|  |
| --- |
| **Vuelo** |
| Numero : String  Origen : Aeropuerto  Destino : Aeropuerto  Fecha : DateTime  aerolinea : Aerolinea |
| importar(String) : List<Aeropuerto>  exportar(String) |

|  |
| --- |
| **Reserva** |
| vuelo : Vuelo  adultos : int  ninyos : int  extra : bool |
| confirmar() |

**Importación del modelo de datos al proyecto:**

Para incorporar el modelo de datos al proyecto deben seguirse siguientes los pasos :

1. \*Desplegar el menú **Proyecto** y elegir la opción **Agregar referencia…**
2. \*En la ventana emergente marcar **Examinar** en la parte derecha y luego pulsar el botón **Examinar…** de la parte inferior.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. \*En el explorador de archivos que aparece localizar y elegir el fichero **Vuelos.dll** que viene con el enunciado de la actividad. Debería aparecer listada en las referencias del explorador de proyectos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. \*Añadir la clausula using correspondiente en nuestro proyecto.

Texto

Descripción generada automáticamente

**Carga de datos al proyecto:**

Junto al modelo de datos encontrareis 3 archivos de datos: **aerolíneas.dat**, **aeropuertos.dat** y **vuelos.dat**. Esos archivos contienen la base de datos necesaria para realizar la actividad. En ellos podemos encontrar 6 aerolíneas, 30 aeropuertos españoles, 6 europeos y 6 norteamericanos. También se han generado 1000 vuelos ficticios.

Para incorporar los datos al proyecto seguir los siguientes pasos:

1. Copiar los 3 archivos .**dat** en la carpeta **debug** del proyecto (se encuentra dentro de la carpeta **bin**).
2. Crear 3 objetos **List** para cada una de las clases del modelo de datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. A continuación, invocar el método estático **importar** de cada una de las clases almacenando su resultado en la lista correspondiente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**IMPORTANTE:** La importación de vuelos debe hacerse en último lugar ya que esta depende de las otras dos.

**Nuevos datos**

Si se desea se pueden añadir nuevos datos a los ya existentes. Para ello simplemente hay que construir una instancia del objeto que se quiere añadir (Aerolínea, aeropuerto o vuelo) y luego invocar su método **exportar** pasando el nombre del fichero como parámetro.

Imagen que contiene Diagrama

Descripción generada automáticamente

**Confirmación de la reserva:**

Una vez obtenidos todos los datos, para realizar el último paso que consiste en la confirmación de la reserva hay que construir una instancia de la clase Reserva pasando como parámetros de construcción el objeto Vuelo elegido, la cantidad de billetes de adulto, la cantidad de billetes de niño y si se desea equipaje extra. Una vez construido dicho objeto invocar su método **confirmar** y si todo funciona correctamente se verá una ventana con todos los datos con lo que se puede dar por finalizada la actividad.

Se pueden utilizar todos los controles estudiados para implementar la interfaz.

Hay que entregar todo el proyecto (la carpeta del proyecto y todo su contenido) comprimido en un único fichero zip.