

Centro Integrado de Formación Profesional **AVILÉS**

Principado de Asturias

ACTIVIDAD INDIVIDUAL PRIMERA EVALUACIÓN

DESARROLLO DE INTERFACES

2º CURSO C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA

1. UNIDADES QUE SE TRABAJAN EN ESTA ACTIVIDAD

[UT1] Creación de interfaces de usuario.

[UT2] Diseño de interfaces gráficas. Usabilidad y accesibilidad.

[UT3] Diseño de componentes.

[UT4] Documentación de aplicaciones.

2. INTRODUCCIÓN

La actividad se basa en la creación de una aplicación para la gestión de vuelos de un aeropuerto. El sistema permitirá el registro de compañías aéreas y de vuelos, así como su relación. Existirá una funcionalidad que permitirá registrar los vuelos con sus orígenes y destinos pudiendo visualizarse paneles de próximas llegadas y salidas.

3. TIPOS DE DATOS

De un modo general, la aplicación manejará los siguientes datos:

- Gestión de compañías aéreas. De cada una de ellas se almacenarán los siguientes datos:
 - o Prefijo. Código numérico entero positivo cuyo máximo es 999. Este campo se puede utilizar como identificador de la relación subyacente en la base de datos.
 - Código. Cadena de dos caracteres de tamaño fijo. Puede contener dos letras mayúsculas o una mayúscula y un número (en este orden). Ejemplos válidos: IB, V7. Ejemplos no válidos: 7V, ib, Ib, iB, 8a, 7-
 - Nombre de la compañía.
 - o Dirección de la sede central.
 - o Municipio de la sede central.
 - Teléfono de información al pasajero. Se trata del teléfono al que tienen que llamar los pasajeros para obtener información. Se componen de un código de país con tres dígitos y el resto de teléfono contendrá un máximo de 12 dígitos más.
 - Teléfono de información a aeropuertos. Es el teléfono que tienen los aeropuertos para contactar con las compañías para gestiones, incidencias, etc. Mismo formato que el de información al pasajero.

Se pueden obtener ejemplos de códigos de compañías reales y sus nombres en la siguiente URL: http://rm-forwarding.com/i/prefijos-companias-aereas/

- Gestión de vuelos (datos base).
 - Código de vuelo. Se compone del código de la compañía y un número entero positivo cuyo máximo es 9999. No es necesario completar con ceros. Ejemplos de códigos de vuelos válidos: V73585, IB480
 - o Aeropuerto origen.

- Aeropuerto destino. Tanto para este campo como para el anterior, se dispondrá de una tabla o fichero que recoja sus datos estando estos precargados, es decir, no hará falta crear un interfaz para su gestión. Se puede partir de la tabla de aeropuertos según el código IATA: https://w.wiki/7iMF Debe contener los siguientes campos:
 - Código IATA.
 - Nombre.
 - Código de municipio. Si es español, código según el INE, el cual servirá para el servicio meteorológico. Se trata de una cadena de cinco caracteres numéricos. Si es internacional, el código será 0000.
- o Número de plazas. Las que tiene disponibles el vuelo en total.
- o Hora oficial de salida.
- o Hora oficial de llegada.
- o Días que opera. Cadena con siete caracteres que contiene qué días de la semana opera el vuelo. Por ejemplo, si lo hace todos los días: "LMXJVSD". Se da por hecho que no hay variaciones anuales, ni en temporada alta/baja ni en festivos.
- Gestión de vuelos diarios. Una vez que se tiene un registro de los vuelos que operan sobre el aeropuerto, es preciso añadir la información precisa para cada vuelo que sale o llega al aeropuerto.
 - Código de vuelo. Relacionado con el vuelo concreto incluido en el apartado anterior.
 No es necesario incluirlo como tal en el modelo, basta con crear una clave que establezca una relación con vuelo.
 - o Fecha del vuelo. Hay que comprobar que es coherente con los días que opera el vuelo.
 - o Hora de salida real.
 - Hora de llegada real. Tanto para esta como para la anterior, por defecto, se registrarán las horas previstas pudiendo cambiarlas para reflejar si el vuelo llega o sale tarde.
 - o Número de plazas ocupadas.
 - o Precio del vuelo. Se considera que es el precio medio por asiento sin tener en cuenta si es Business o turista.

4. FUNCIONALIDAD

Como ya se ha explicado antes, la aplicación será capaz de gestionar un aeropuerto. Existirá una pantalla principal desde la que se podrán realizar las principales funcionalidades de la aplicación. En un archivo de recursos se debe registrar el código del aeropuerto que se está gestionando, lo que condicionará si un vuelo es llegada o salida.

4.1. GESTIÓN DE COMPAÑÍAS AÉREAS

La aplicación permitirá dar de alta, baja, modificar y consultar compañías aéreas. Todos los formularios estarán validados. Las compañías estarán registradas en la aplicación y podrán ser dadas de alta al crear un vuelo. El formato de guardado de datos de las compañías será un archivo CSV.

4.2. GESTIÓN DE VUELOS

Se permitirá dar de alta, baja, modificar y consultar vuelos. Al igual que en el apartado anterior, se registrarán los distintos vuelos en un archivo CSV. Hay que distinguir entre los datos base de los vuelos (los que aparecen en la sección Gestión de datos base de vuelos del apartado 3) y los de los vuelos diarios; ambos datos deben ir en ficheros separados. Por una parte, los datos base determinan la información que tiene un vuelo independientemente de las veces que salga. En la parte de vuelos diarios, se registra la información de cada vuelo concreto que opera en cada momento. Por tanto, los datos base serán completamente modificables mientras no haya un vuelo concreto. Si ya ha salido o llegado un vuelo en una fecha concreta no se podrá modificar ningún dato. Si se requieren cambiar fechas, horarios, plazas, etc. habrá que crear un nuevo vuelo.

4.3. PERSISTENCIA DE DATOS

El sistema permitirá almacenar los datos en ficheros, serializando para ello la información contenida en las estructuras de datos utilizadas. Al arrancar el sistema se cargará toda la información almacenada en estos ficheros. El sistema guardará la información en dichos ficheros cuando el usuario realice modificaciones en el sistema. La ruta de los ficheros estará dentro del proyecto de NetBeans para evitar problemas al mover la aplicación.

El sistema permitirá consultar la siguiente información:

- Panel de salidas. Mostrará las salidas del aeropuerto para el día seleccionado (por defecto, la fecha en la que se consulta) ordenadas por hora de salida ascendente.
- Panel de llegadas. Mostrará las llegadas del aeropuerto para el día seleccionado (por defecto, la fecha en la que se consulta) ordenadas por hora de llegada ascendente.
- Vuelos por compañía que operarán un día concreto (por defecto, la fecha en la que se consulta).
- Recaudaciones de vuelos completados para un día concreto. Si se trata del día actual, se toma
 la suma de los vuelos que ya han salido de su aeropuerto origen (tomando la fecha real, no
 la prevista). Se trata de la suma del importe recaudado de todos los vuelos que cumplan con
 ese criterio teniendo en cuenta el número de plazas ocupadas y el precio previsto.
- Lista de vuelos previstos a un destino concreto los siguientes siete días a la fecha actual ordenados por fecha y hora de salida.

Aunque el sistema está previsto para su uso con ficheros, se puede desarrollar enteramente con una conexión a una base de datos PostgreSQL. En caso de optar por esta posibilidad, hay que incluir archivos .sql con la creación de la base de datos y la inserción de los registros base para comenzar a trabajar. En cualquier caso, es preciso conocer cómo trabajar con ficheros ya que la prueba práctica se realizará utilizando ese origen de datos.

5. FASES DE DESARROLLO

Este proyecto se irá desarrollando según se vayan viendo los diferentes contenidos en las clases presenciales. Por tanto, habrá que desarrollar las siguientes tareas que estarán asociadas a las diferentes unidades de trabajo de la evaluación:

- 1. [UT1 y UT2] Creación de la aplicación con interfaz gráfico en Swing. En esta parte se diseñará la aplicación cubriendo toda su especificación utilizando los componentes gráficos vistos en la UT1 y siguiendo los criterios de usabilidad de la UT2.
- 3. [UT3] Creación de un componente que permita mostrar la temperaturas mínimas y máximas previstas para un aeropuerto en el día actual. Este componente hará uso de una llamada a la API de servicios REST de la AEMET que puede consultarse en la siguiente dirección: https://opendata.aemet.es/centrodedescargas/inicio

La consulta se hará por municipio obteniendo este de los datos del aeropuerto. El componente simplemente recogerá el código del municipio y la API Key de conexión. La recogida de esta información se hará utilizando un **editor de propiedades**. Se incluirá en la interfaz de consulta de un vuelo como un elemento más mostrándose las temperaturas del aeropuerto origen, así como las del destino.

4. [UT4] Creación de un sistema de ayuda sensible al contexto. Se introducirán al menos tres páginas de ayuda en los puntos claves de la aplicación. La ayuda será accesible al pulsar la tecla F1 y a través de una opción de menú en el programa.

La aplicación se desarrollará obligatoriamente en capas, de manera que la parte del interfaz sea totalmente independiente de la lógica de negocio y del acceso a los datos. Concretamente será esta división en capas lo que te permitirá programar de manera correcta la persistencia de la aplicación, tanto usando una base de datos como usando ficheros.

6. ENTREGA Y EVALUACIÓN DE LA PRÁCTICA

La práctica estará completamente alojada en un repositorio de GitHub. Formará parte de la evaluación el trabajo realizado sobre esta plataforma. Esto quiere decir que no se admitirán prácticas que no se hayan subido a GitHub una vez acabadas. La actividad debe estar desarrollada y subida progresivamente al repositorio para poder ver el trabajo realizado durante toda la evaluación.

Se valorarán los siguientes aspectos:

- Grado de consecución de la funcionalidad de la práctica [80% de la nota].
- Calidad e idoneidad del código fuente [20% de la nota].