

# DAI

## Práctica 1: Aplicaciones Web con ASP, JSP y Ajax

### Introducción

En esta práctica vamos a desarrollar una **aplicación web para una agencia de viajes** que permita la realización de una serie de gestiones relacionadas con la **reserva de viajes on-line**. La web dispondrá de una **zona accesible a cualquier usuario**, con información de la empresa, ubicación, imagen corporativa, búsqueda de vuelos que cumplen una condición, etc. y **otra zona, accesible únicamente tras llevar a cabo un proceso de identificación**, que permitirá a un usuario consultar las reservas realizadas.

Para el desarrollo de la parte “estática” del sitio emplearemos **XHTML o HTML5 y CSS**, mientras que para la parte dinámica emplearemos las tecnologías **ASP** con lenguaje **VBScript** y **JSP**.

La aplicación empleará una sencilla **base de datos**, desarrollada en **Firebird** cuya plantilla básica está en la VM del Campus Virtual de la asignatura y a la que progresivamente le iremos añadiendo funcionalidad. El acceso deberá realizarse a través de **ODBC**. Todos los drivers y ejecutables necesarios se encuentran disponibles igualmente en la sección Prácticas del Campus Virtual.

Dispondrá únicamente de las siguientes tablas:

- ✚ **Ciudad:** con información de las diferentes ciudades desde las que, y hacia las que, se pueden realizar vuelos. Debe almacenar al menos el precio asociado a las tasas de aeropuerto de dicha ciudad.
- ✚ **Compañía:** información de las compañías aéreas con las que opera la empresa.
- ✚ **Vuelo:** contiene toda la información relacionada con los vuelos: Ciudad origen, Destino, Fecha y hora de salida, Compañía, duración del vuelo, N° plazas disponibles, Avión,...
- ✚ **Avión:** Tabla en la que se registran los diferentes modelos de avión que incluirá como mínimo el número de plazas del mismo y el precio base del vuelo.
- ✚ **Reserva:** Contiene un registro de las solicitudes de reserva de vuelo que se realizan. Cada nuevo registro deberá generar un número identificativo único del número de reserva, Nombre y Apellidos de la persona, DNI, Vuelo, N° de Asientos reservados...

### PARTE OBLIGATORIA

#### Desarrollo de la interfaz básica (4 puntos)

El sitio web de la agencia de viajes debe permitir:

##### GESTIÓN DE CIUDADES:

Para acceder a este apartado será necesario realizar previamente un **acceso identificado de administrador**. Una vez logueado correctamente, esta opción mostrará un **listado de todas las ciudades de la base de datos** desde (y a) las que se puede volar.

Al hacer clic sobre una de las ciudades de la lista, se accederá a la ficha **Detalles de la Ciudad**, donde podrá **modificarse** los datos de cada ciudad (**NO** se podrá modificar el **IDCIUDAD**, que **será asignado de forma automática mediante números enteros secuenciales**).

Además, se mostrará un botón que permita **añadir una nueva ciudad**.

También se incluirá en este apartado un botón que permita **Eliminar Ciudades** de la base de datos.

## GESTIÓN DE RESERVAS

**Sin necesidad de identificarse**, cualquier usuario podrá **consultar vuelos disponibles** mediante un formulario de consulta. **Ningún dato es obligatorio**, por lo que si se dejan en blanco, la aplicación deberá devolver una lista con todos los vuelos disponibles; o bien un listado de vuelos que cumpla las condiciones expresadas, si se rellenan los campos del formulario. Se debe mostrar el precio del vuelo según se a continuación: **Precio = (Nº Plazas del avión – Nº Plazas Disponibles) / Nº Plazas del avión \* Precio Base del vuelo + Tasas aeropuerto**. De este modo el precio se incrementa a medida que disminuye el número de plazas disponibles.

Una vez obtenido el listado de vuelos disponibles, el usuario podrá **realizar una reserva sobre alguno de los vuelos localizados**:

- ✚ Se deberá comprobar que quedan plazas suficientes y actualizar el número de plazas disponibles tras realizar la reserva.
- ✚ **Se devolverá al usuario un identificador único de la reserva** realizada para posteriores consultas y **se le indicará el precio tras seleccionar el número de asientos** que desea reservar.

Tras la reserva del vuelo de ida, la interfaz de consulta de vuelos **deberá mostrar también los posibles vuelos de vuelta**, permitiendo por tanto al usuario realizar una segunda reserva con el vuelo de vuelta.

Además, el usuario podrá consultar reservas realizadas **a partir de su DNI y del número de reserva generado en el paso anterior**. Una vez mostrados los datos de la reserva **se debe poder cancelar**, en cuyo caso el número de plazas disponibles en el vuelo se debe incrementar. Si faltan menos de 48 horas para la salida del vuelo, la aplicación mostrará un mensaje de error indicando que ha finalizado el plazo máximo de cancelación.

## Inclusión de Técnicas Ajax en la interfaz de usuario (2 puntos)

Una vez finalizada la versión básica de la interfaz, tal y como se pide en el apartado anterior, se desarrollará una versión alternativa de la misma [se entregarán ambas versiones, por separado], **que permita la búsqueda de vuelos empleando técnicas AJAX**. De manera que:

- ✚ La **recarga de datos** en el cuadro de lista desplegable de **ciudad de destino**, se realizará en segundo plano, **sin recargar la página**, al cambiar la ciudad de origen, mostrando sólo aquellas ciudades para las que existan vuelos.
- ✚ Del mismo modo, **la tabla que contiene los resultados de la búsqueda de vuelos** deberá rellenarse en segundo plano **sin necesidad de recargar la totalidad de la página**.

## PARTE OPCIONAL

### Desarrollo de la interfaz para el administrador (2 puntos)

En esta parte de la práctica vamos a desarrollar la interfaz que **permitirá a los usuarios con perfil de administrador añadir, consultar y modificar información** de la base de datos. Para ello se diseñará, un **interfaz** mediante tecnologías **ASP + AJAX**, que recibirá la información del usuario y se encargará de **ejecutar las sentencias SQL** sobre la base de datos.

Se debe añadir al sitio desarrollado en el apartado anterior las siguientes páginas y clases de Java:

- ✚ **ServVuelos.asp**: Página ASP que recibirá, en forma de peticiones **GET**, información del interfaz y que devolverá, en segundo plano, a éste la información solicitada (si es una consulta) o se encargará de insertar o actualizar los registros (si se trata de una inserción o modificación). Los **parámetros** serán:
  - **Accion**: con tres posibles valores: **consulta, insercion, modificacion**.
  - **NombreCampo** = Valor: Lista de campos con sus respectivos valores para **insertar en la tabla de vuelos**.
    - Cada parámetro se separará de los demás empleando el carácter "&", según la especificación del protocolo HTTP.
- ✚ **GestionVuelos.asp**: interfaz que permita al administrador insertar, consultar o modificar vuelos. Empleando botones de comandos, menús, controles de edición de texto, cuadros de lista, checkbox, o cualquier otro componente disponible, debe permitir al administrador:
  - **Insertar un nuevo vuelo**
  - **Visualizar un vuelo a partir** del identificador del mismo
  - **Modificar cualquiera de los campos** de un vuelo (salvo el identificador de vuelo)

### Inserción de compañías y consulta de reservas en JSP (2 puntos)

En esta parte de la práctica vamos a **desarrollar dos nuevas funcionalidades** empleando tecnología **JSP**. Dado que ya tenemos un IIS escuchando en el puerto 80, será necesario configurar el servidor **Tomcat** en un puerto distinto, por ejemplo, **8081**.

#### Consulta de Reservas:

Se añadirá una nueva página, accesible desde la página principal, que muestre un listado de reservas realizadas a partir del **DNI** de la persona que la realizó y de una **ciudad de origen**. Será necesario implementar un formulario que solicite el DNI y la ciudad de origen del vuelo, y además muestre un botón que será el encargado de **llamar a la página JSP encargada de recuperar los datos**.

Los datos **se mostrarán en una tabla que incluirá**, al menos, columnas para IdReserva, IdVuelo, Apellidos, Nombre, Ciudad Origen, Ciudad Destino, Fecha del Vuelo, Compañía y Estado (Cancelado o No Cancelado); se puede añadir cualquier otro dato que consideréis interesante.

#### Inserción de Compañías:

Se añadirá una nueva página, **accesible desde el acceso identificado de administrador**, que permita **introducir compañías** en la base de datos. Se deberá implementar un formulario que solicite los datos de la compañía (IDCOMPANIA, junto con el resto de campos). Además, mostrará un botón que será el encargado de **llamar a la página JSP encargada de insertar los datos** de la compañía y mostrar un mensaje de confirmación al usuario.

### Normas de Entrega

---

La entrega de la práctica se hará en un **único fichero .ZIP**, que incluirá todas las páginas implementadas, ficheros java, jsp, asp, imágenes y breve memoria descriptiva del sitio desarrollado. Para ello se empleará el acceso habilitado en la sección Prácticas del Campus Virtual en la fecha prevista.

**ESTA PRÁCTICA SUPONE EL 24% DE LA NOTA FINAL DE LA ASIGNATURA.**