

# Comunitat Valenciana Skills

---

## Modalidad de competición: Desarrollo Web

Fase Previa

Dirección General de Formación Profesional y Enseñanzas de Régimen Especial

**DIRECCIÓ GENERAL DE FORMACIÓ PROFESSIONAL I  
ENSENYAMENTS DE RÈGIM ESPECIAL**



## **ÍNDICE**

1. Introducció
2. Estructura de la Fase Previa
3. Descripció de la prova tècnica

## 1. Introducció

Este documento establece el desarrollo de la **Fase Previa** para la Modalidad de competición "Desarrollo Web" incluida en la competición Comunitat Valenciana Skills 2017.

Dicha fase es necesaria ya que hay inscritos 11 alumnos y en la competición únicamente pueden participar 10 alumnos.

Debido a la que la competición da preferencia a los alumnos que están cursando el ciclo formativo de grado superior de Desarrollo de Aplicaciones Web, los alumnos que estén cursando dicho ciclo participarán automáticamente en la competición Comunitat Valenciana Skills 2017.

Por lo tanto la Fase previa solo será realizada entre los alumnos de otros ciclos formativos. En este caso hay 4 alumnos de otros ciclos por lo que deberá seleccionarse 3 alumnos entre los 4 existentes.

Los alumnos que deberán realizar la fase previa son los siguientes:

Alumno	Centro	Ciclo	Tutor
Antonio Gallego Martínez	IES Severo Ochoa	ASIR	Daniel Calatayud
Miguel Pastor Sevilla	IES Severo Ochoa	DAM	Marina Navarro
Sergio Peris Tárrega	IES Dr Luís Simarro	SMR	Enric Giménez
Jordan Perucho Ortega	IES Dr Luís Simarro	SMR	Ana Gema Sanchis

Los alumnos excluidos de realizar la fase previa por ser alumnos de DAW son los siguientes:

Alumno	Centro	Ciclo	Tutor
Oscar Llorens I Lurbe	IES L'Estació	DAW	Pere Crespo
Sergio Martínez Guerrero	IES L'Estació	DAW	Yolanda Moneris
Oscar Seguí Guerola	IES L'Estació	DAW	Pepe Muñoz
Leandro Botella Penalva	IES Macià Abela	DAW	Carlos Fernadez
José Miguel Sánchez Alarcón	IES Mare Nostrum	DAW	Pedro Prieto
Víctor Lázaro Palmero	IES La Vereda	DAW	Carlos Gutierrez
Daniel Ortega Conesa	IES Severo Ochoa	DAW	Daniel Calatayud

## 2. Estructura de la Fase Previa

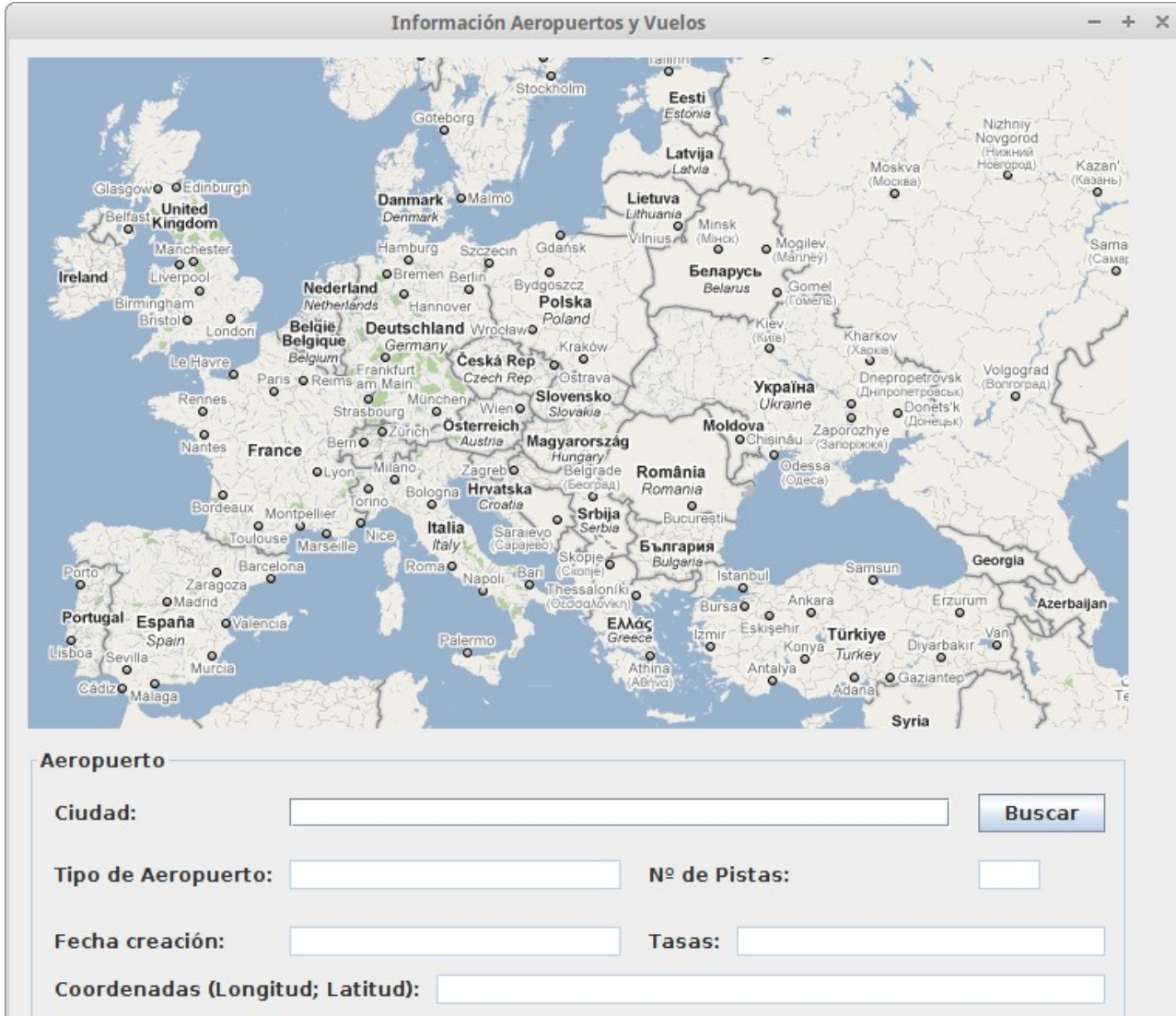
La fase previa es una prueba técnica que consistirá en realizar un programa en el lenguaje Java o .NET (C# o Visual Basic). El programa lo realizará el alumno en su propio centro bajo la supervisión de su tutor. La duración de la prueba será como máximo de 5 horas. Una vez finalizada la prueba, el tutor deberá enviar el resultado de la misma al experto del Skill de desarrollo Web, Lorenzo González al siguiente correo "[lorenzo.profesor@gmail.com](mailto:lorenzo.profesor@gmail.com)". La fecha límite para enviar la prueba será el 10 de Noviembre de 2017 a las 23:59. En caso de no haber entregado el resultado de la prueba antes de esa fecha, el alumno será automáticamente excluido de la fase previa, lo que le impedirá participar en la competición Comunitat Valenciana Skills 2017.

Una vez entregados el resultado de las pruebas, el experto evaluará cada una de las pruebas mediante una nota numérica de 0 a 10. El experto informará a los tutores antes del 16 de Noviembre del resultado de la misma. Los alumnos con las 3 mejores notas serán los alumnos que participarán en la competición Comunitat Valenciana Skills 2017. El resultado de dichas notas será definitivo y no podrán modificarse.

### 3. Descripción de la prueba técnica

El objetivo de esta prueba es simplemente crear una aplicación de escritorio que dado el nombre de un aeropuerto muestre información sobre dicho aeropuerto.

La pantalla principal de la aplicación será la siguiente:



**Información Aeropuertos y Vuelos**

Aeropuerto

Ciudad:

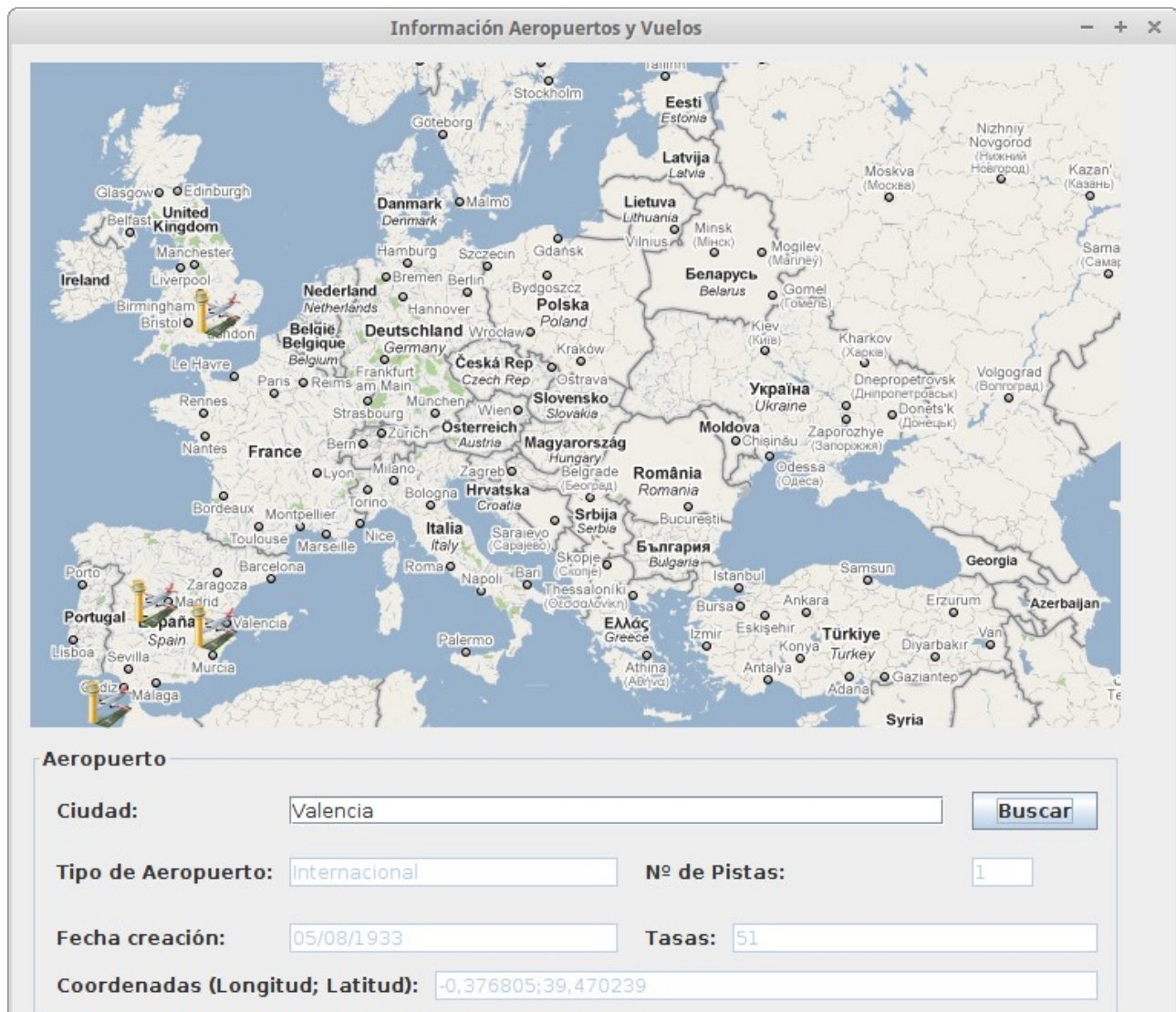
Tipo de Aeropuerto:  Nº de Pistas:

Fecha creación:  Tasas:

Coordenadas (Longitud; Latitud):

El usuario deberá escribir el nombre de un aeropuerto en la caja de texto llamada "Ciudad" y pulsar el botón de "Buscar". En caso de que el aeropuerto no exista se deberá mostrar un mensaje en una ventana modal indicando que el aeropuerto no existe. Si dicho aeropuerto si que

existe en el sistema, deberá mostrar en las otras cajas de texto la información del aeropuerto y en el mapa superior los aeropuertos a los que hay vuelo directo y el del propio aeropuerto. En la siguiente imagen puede verse un ejemplo para el aeropuerto de Valencia:



The screenshot shows a web application window titled "Información Aeropuertos y Vuelos". It features a map of Europe with several airports highlighted in yellow, indicating direct flights. Below the map is a form with the following fields:

Aeropuerto	
Ciudad:	<input type="text" value="Valencia"/> <input type="button" value="Buscar"/>
Tipo de Aeropuerto:	<input type="text" value="Internacional"/> <input type="text" value="Nº de Pistas: 1"/>
Fecha creación:	<input type="text" value="05/08/1933"/> <input type="text" value="Tasas: 51"/>
Coordenadas (Longitud; Latitud):	<input type="text" value="-0,376805;39,470239"/>

En el ejemplo podemos ver que el tipo de aeropuerto de Valencia es "Internacional", que tiene una pista, fue creado el 5/08/1933. Las tasas del aeropuerto son 51€ y sus coordenadas son -0.376805, 39.470239.

Por último en el mapa se muestra a Valencia junto a a las 3 ciudades a las que tiene vuelos , que son: Madrid, Cádiz y Londres.

Al final del enunciado están todos los datos de los aeropuertos.

Para ayudar a hacer la aplicación se deberán crear entre otras clases las siguientes:

1.- Crea un enumerado llamado “**TipoAeropuerto**”.

Los posibles valores del enumerado son:

National

International

2.- Crea una clase llamada “**Coordenada**”

a) Añadir las siguientes propiedades

longitud

latitud

b) Crea un constructor en el que se puedas pasar la longitud y latitud

c) Realizar las siguientes validaciones en la clase (donde sea necesario) de forma que siempre se cumpla que:

- La Latitud debe estar entre los valores -90 y 90
- La Longitud debe estar entre los valores -180 y 180

En caso de que alguno de los valores no cumpla la validación se deberá lanzar una excepción

3.- Crea una clase llamada “**Aeropuerto**”

a) Añadir las siguientes propiedades

nombreCiudad

tipoAeropuerto

numeroPistas:

fechaConstruccion

ciudadesDestino: que será un array ( o similar) de Strings con las ciudades a las que hay vuelos directos

coordenadas: Las coordenadas de Longitud y Latitud de la posición del Aeropuerto

b) Añadir un método llamado “getTasas()” que nos indicará el precio de las tasas .

El precio se calcula en proporción a los aeropuertos a los que tiene destinos y del tipo de aeropuerto.

Si el tipo de aeropuerto es “National” el importe de la tasa será de 8€ por aeropuerto destino. Si el aeropuerto es “International” el importe de la tasa será de 17€ por aeropuerto destino.

Es decir habrá que multiplicar el “precioPorDestino()” por el nº de aeropuertos a los que tiene destino este aeropuerto.

c) Crear un método privado llamado “precioPorDestino()” que nos retornará 8 si el tipo de aeropuerto es “National” o retornará 17 si el tipo de aeropuerto es “International”, en cualquier otro caso lanzar una excepción.

d) Añade un constructor de forma que en el constructor se puedan añadir todas las propiedades de la clase. Se declararán en el mismo orden que están en la lista del apartado a).

e) Realizar las siguientes validaciones en la clase (donde sea necesario) de forma que siempre se cumpla que:

- El nº de pistas debe ser mayor que 0
- El aeropuerto debe tener al menos un destino.
- La fecha de construcción debe ser menor que el día de hoy y no ser vacía.
- Las coordenadas no pueden estar vacías.
- El tipo de aeropuerto no puede estar vacío.
- El nombre del aeropuerto no puede estar vacío

En caso de que alguno de los valores no cumpla la validación se deberá lanzar una excepción

4. Crea una clase llamada “**InformacionTopografica**”.

Esta clase tendrá dos métodos:

- getPosicionAeropuerto: Método público y estático con un único parámetro que será el nombre del aeropuerto. Retornará un objeto Coordenada con las coordenadas donde se encuentra ese aeropuerto.



- `getPosicionesAeropuertos`: Método privado y estático que retornará un array asociativo (Map en Java y Dictionary en .Net) con las coordenadas de todos los aeropuertos.

5. Crea una clase llamada “**Aena**”.

Esta clase tendrá dos métodos:

- `getAeropuerto`: Método público y estático con un único parámetro que será el nombre del aeropuerto. Retornará un objeto `Aeropuerto` con toda la información de ese aeropuerto.
- `getAeropuertos`: Método privado y estático que retornará un array asociativo (Map en Java y Dictionary en .Net) con la información de todos los aeropuertos.

6. Crea la clase principal donde se mostrará la información del mapa y del aeropuerto y que hará uso de las clases antes expuestas. Y que debe realizar toda la funcionalidad que se ha explicado.

7. Datos de los Aeropuertos:

Nombre	Tipo	Nº Pistas	Fecha	Ciudades
Valencia	Internacional	1	05/08/1933	Madrid,Londres,Cádiz
Madrid	Internacional	4	22/04/1931	Londres,Roma,Valencia,Cádiz
Cádiz	Nacional	1	20/11/1975	Madrid,Valencia

8. Longitud y Latitud de los Aeropuertos:

Ciudad	Longitud	Latitud
Cadiz	-6.292409	36.529942
Valencia	-0.376805	39.470239
Madrid	-3.700345	40.416691
Londres	-0.126236	51.500152
Roma	12.482324	41.895466