

# TALLER DE JAVA EE 7

Una importante Empresa de “Comida de Perros”, ha decidido organizar “Campeonatos de Perros” para acercar su marca a los consumidores, por lo que ha solicitado la construcción de un sistema que les permita registrar participantes a un campeonato y poder realizar consultas por raza de los mismos.

## ***Requerimientos funcionales***

Después de algunas entrevistas con el cliente, el analista ha definido los siguientes requisitos.

Tabla de Requisitos Funcionales.

Nro.	Requisito
F001	Debe permitir el ingreso de Participantes, solicitando el nombre, Raza y el ID de Registro del Participante.
F002	No debe permitir el ingreso de Participantes con mismo ID de Registro.
F003	Debe permitir mantener una lista de Razas. (*)
F004	Debe permitir consultar todos los Participantes existentes.
F005	Debe permitir consultar los Participantes que son de una misma Raza.
F006	Debe permitir eliminar Participantes.

(\*) Para el examen, la lista de Razas viene ya con datos.

## Prototipos de Interfaz de Usuario (vistas)

Basado en los requisitos funcionales, el analista ha propuesto las siguiente interfaces de usuario.

**Opciones:**      [Agregar Participante](#)      [Listar Participantes](#)

**Agregar Nuevo Participante**  
  

Nombre

Raza

▼

ID Reg.

Registrar

<b>Nombre Interfaz</b>	Agregar Participante
<b>Requerimientos Funcionales</b>	F001 F002
<b>Observaciones</b>	<ul style="list-style-type: none"><li>• Se debe validar que el usuario ingrese valores para todos los campos.</li><li>• Debe indicar al usuario si el registro fue creado o tuvo problemas de validación.</li><li>• Se debe registrar automáticamente la fecha de grabación de registro.</li><li>• Se debe registrar un log con las actividades del usuario, ya sean de éxito o fallidas.</li></ul>

**Opciones:**
[Agregar Participante](#)
[Listar Participantes](#)

Listado de Participantes

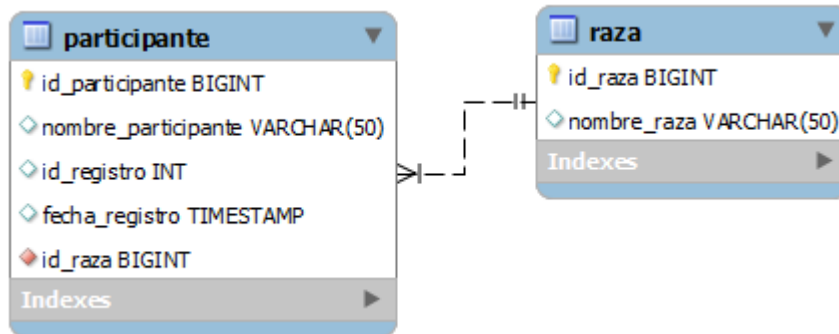
Seleccione una Raza:

Nombre	Raza	ID Reg.	Fecha Registro	Acción
NOMBRE	RAZA	####	dd/mm/aaaa hh:mm	<input type="button" value="Eliminar"/>
...				<input type="button" value="Eliminar"/>
...				<input type="button" value="Eliminar"/>

<b>Nombre Interfaz</b>	Listar Participantes
<b>Requerimientos Funcionales</b>	F004 F005 F006
<b>Observaciones</b>	<ul style="list-style-type: none"> <li>Al buscar, si no se selecciona una Raza, debe mostrar todos los Participantes.</li> <li>Al eliminar el Participante, se debe indicar al usuario que el participante fue eliminado.</li> <li>Se debe registrar un log con las actividades del usuario, ya sean de éxito o fallidas.</li> </ul>

## Modelo de Datos

Basado en la documentación del proyecto, el arquitecto ha desarrollado el siguiente modelo de datos.



## Descripción de Tablas

**Tabla: PARTICIPANTE**

Nombre Atributo	Tipo Dato	Observación
id_participante	BIGINT	Clave Primaria de la tabla - Auto numérico
nombre_participante	VARCHAR(50)	Nombre del Participante
id_registro	INT	ID de Registro de inscripción del Participante
fecha_registro	TIMESTAMP	Fecha / Hora en que se crea el registro
id_raza	BIGINT	Clave Foránea a la tabla RAZA

**Tabla: RAZA**

Nombre Atributo	Tipo Dato	Observación
id_raza	BIGINT	Clave Primaria de la tabla - Auto numérico
nombre_raza	VARCHAR(50)	Nombre de las posibles Razas de Perros  Ej. Beagle, Chihuahua, Pekinés, etc...

## ***Consideraciones al construir la aplicación.***

El arquitecto ha solicitado que tenga presente las siguientes consideraciones de diseño.

- Utilizar patrón de arquitectura MVC.
- Utilizar JSP como Vista.
- Utilizar JSTL + EL para el despliegue de información.
- Utilizar Servlet como Controladores.
- Utilizar Clases Java para el Modelo.
- Utilizar Excepciones para indicar problemas en el Modelo.
- Utilizar patrón DAO para el acceso a datos.
- Utilizar framework de Logging para registrar la actividad del usuario y excepciones ocurridas en el sistema.