Proyecto 1ero - Técnico Superior en Desarrollo de Aplicaciones Multiplataforma

Caso de estudio

Asociación contra el cáncer de testículo.

Renaud Bronchart

INDICE

[Resumen 5](#_Toc167011430)

[1 Introducción 6](#_Toc167011431)

[1.1 Justificación del proyecto 6](#_Toc167011432)

[1.2 Descripción General 7](#_Toc167011433)

[1.3 Objetivos principales 8](#_Toc167011434)

[2 Requisitos 8](#_Toc167011435)

[2.1.1 Requisitos funcionales 8](#_Toc167011436)

[2.1.2 Requisitos específicos 8](#_Toc167011437)

[3 Análisis 10](#_Toc167011438)

[3.1 Casos de uso 10](#_Toc167011439)

[3.1.1 **Anónimo**: visitante estándar sin estar registrado 10](#_Toc167011440)

[3.1.2 **Usuario**: Usuario registrado 11](#_Toc167011441)

[3.1.3 **Administrador**: Administrador del portal web 12](#_Toc167011442)

[3.2 Diagrama Entidad-Relación 13](#_Toc167011443)

[3.2.1 Paso a Tablas - Asociación contra el cáncer 13](#_Toc167011444)

[3.2.2 Diagrama de clases 14](#_Toc167011445)

[3.3 Diagrama de secuencia 16](#_Toc167011446)

[3.3.1 Visualizar contenido 16](#_Toc167011447)

[3.3.2 Insertar contenido 16](#_Toc167011448)

[3.3.3 Editar contenido 17](#_Toc167011449)

[3.3.4 Borar contenido 17](#_Toc167011450)

[4 Diseño 18](#_Toc167011451)

[4.1 Arquitectura cliente – servidor 18](#_Toc167011452)

[4.2 Capa de presentación 20](#_Toc167011453)

[4.2.1 Front-end 20](#_Toc167011454)

[4.2.2 El Back-end 22](#_Toc167011455)

[5 Implementación 26](#_Toc167011456)

[5.1 Tecnologías 26](#_Toc167011457)

[5.1.1 HTML 26](#_Toc167011458)

[5.1.2 CSS 26](#_Toc167011459)

[5.1.3 JAVA 26](#_Toc167011460)

[5.1.4 JAVASCRIPT 27](#_Toc167011461)

[5.1.5 MySQL 27](#_Toc167011462)

[5.1.6 JSON 27](#_Toc167011463)

[5.2 Herramientas utilizadas 28](#_Toc167011464)

[5.2.1 Dia Diagram 28](#_Toc167011465)

[5.2.2 Eclipse Java Web Developers 28](#_Toc167011466)

[5.2.3 Apache Tomcat 10: 28](#_Toc167011467)

[5.2.4 Java doc 28](#_Toc167011468)

[5.2.5 Git 28](#_Toc167011469)

[5.2.6 VISUAL CODE 28](#_Toc167011470)

[5.3 Detalles implementación 29](#_Toc167011471)

[5.3.1 Interfaz Front-end (índex ) 29](#_Toc167011472)

[5.3.2 Añadir Carrera 32](#_Toc167011473)

[5.3.3 Listar carreras 34](#_Toc167011474)

[5.3.4 Connexion a la base de datos 36](#_Toc167011475)

[5.3.5 DAO 37](#_Toc167011476)

[5.3.6 Zona usuarios 38](#_Toc167011477)

[6 Conclusiones 38](#_Toc167011478)

[7 Bibliographia 38](#_Toc167011479)

*Me gustaría mostrar mi más sincero agradecimiento y gratitud a Laurence P. Sin su apoyo, no habría conseguido empezar a estudiar de nuevo y llegar a hacer este proyecto.*

# Resumen

Esta es la memoria del Proyecto 1er curso Técnico Superior en Desarrollo de Aplicaciones Multiplataforma cursada en el Instituto Formación Profesional Nebrija.

El proyecto está basado en una aplicación web diseñada y desarrollada para poder mejorar la gestión de contenido y el control de inventario en un entorno de una asociación.

La gestión será completamente editable por medio de menús y formularios los cuales han sido realizados con las tecnologías web actuales como HTML, JAVA, MySQL.

El objetivo principal de esta aplicación web es permitir la difusión de informaciones, la gestión de carreras y dar visibilidad a asociación.

# Introducción

En nuestros días, el contenido, la gestión y la difusión de informaciones sobre una organización como una asociación es primordial. Es por ello, que cualquier organismo que no tenga una aplicación web de calidad, para facilitar la interactuación con sus usuarios le puede impactar negativamente.

En este proyecto, se pretender crear una aplicación web que permite dar visibilidad a asociación y permitir una gestión eficaz de sus contenidos con una interfaz fácil de usar y de mantener.

El portal va a permitir la creación, la modificación, la difusión y la gestión de contenidos sobre una asociación contra el cáncer de testículo.

## Justificación del proyecto

En nuestra época, es muy importante de poder apoyar las diferentes asociaciones que existen. Una aplicación web como asociación contra el cáncer de testículo permite la difusión de informaciones, y dar más visibilidad a este tema tan importante.  
  
Decidí crear una aplicación web sobre el tema del cáncer de testículo porque tuve un cáncer que me ha afecto en mi vida y desde entonces participo mas en todo en relación con el cáncer.

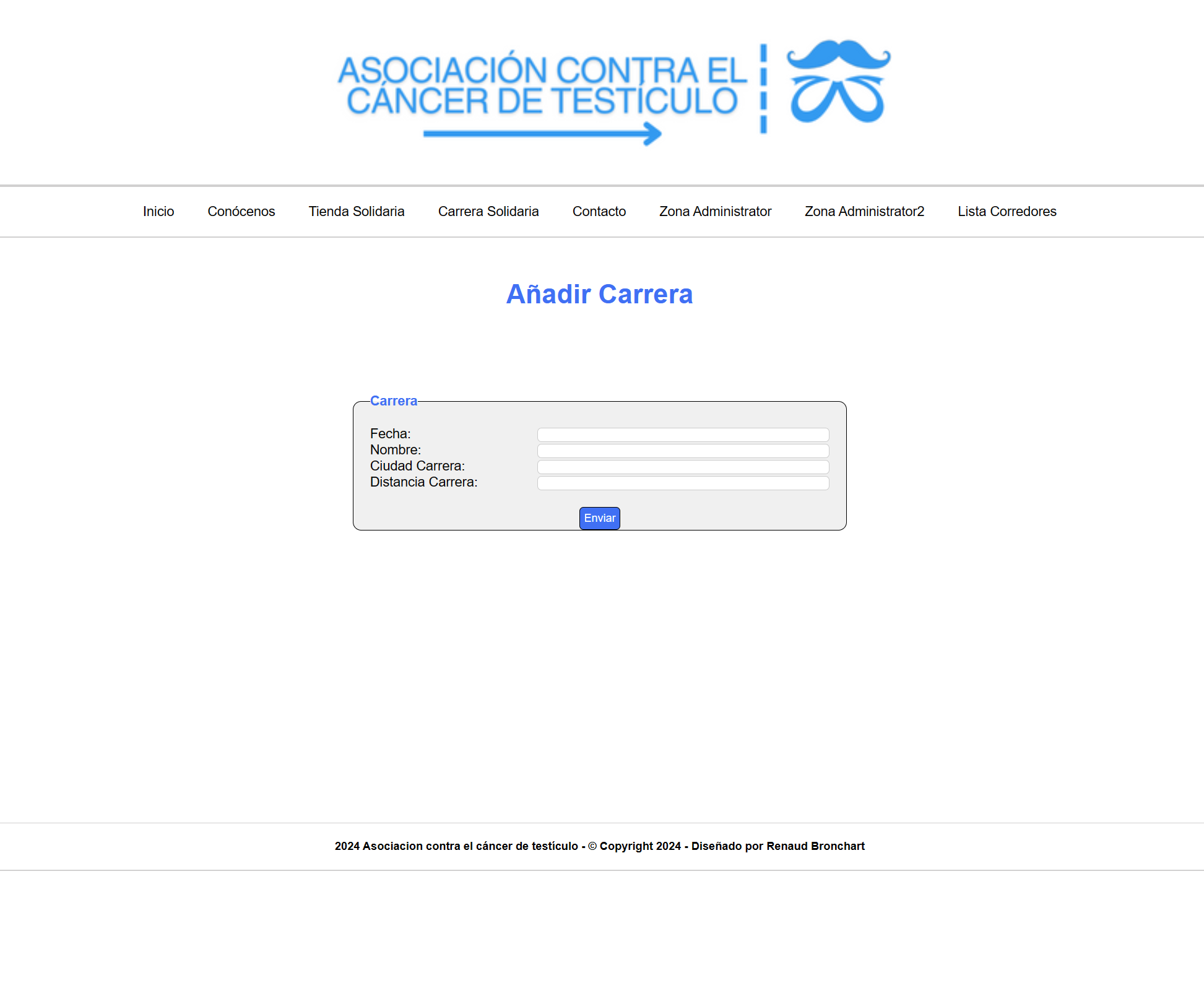
Como soy un aficionado del deporte y más del correr, quería combinar una aplicación que permite la difusión de noticias sobre asociación e informar sobre el tema. También se podría crear eventos como carrera de 10k para promover asociación y permitir dar más visibilidad sobre el cáncer de testículo.

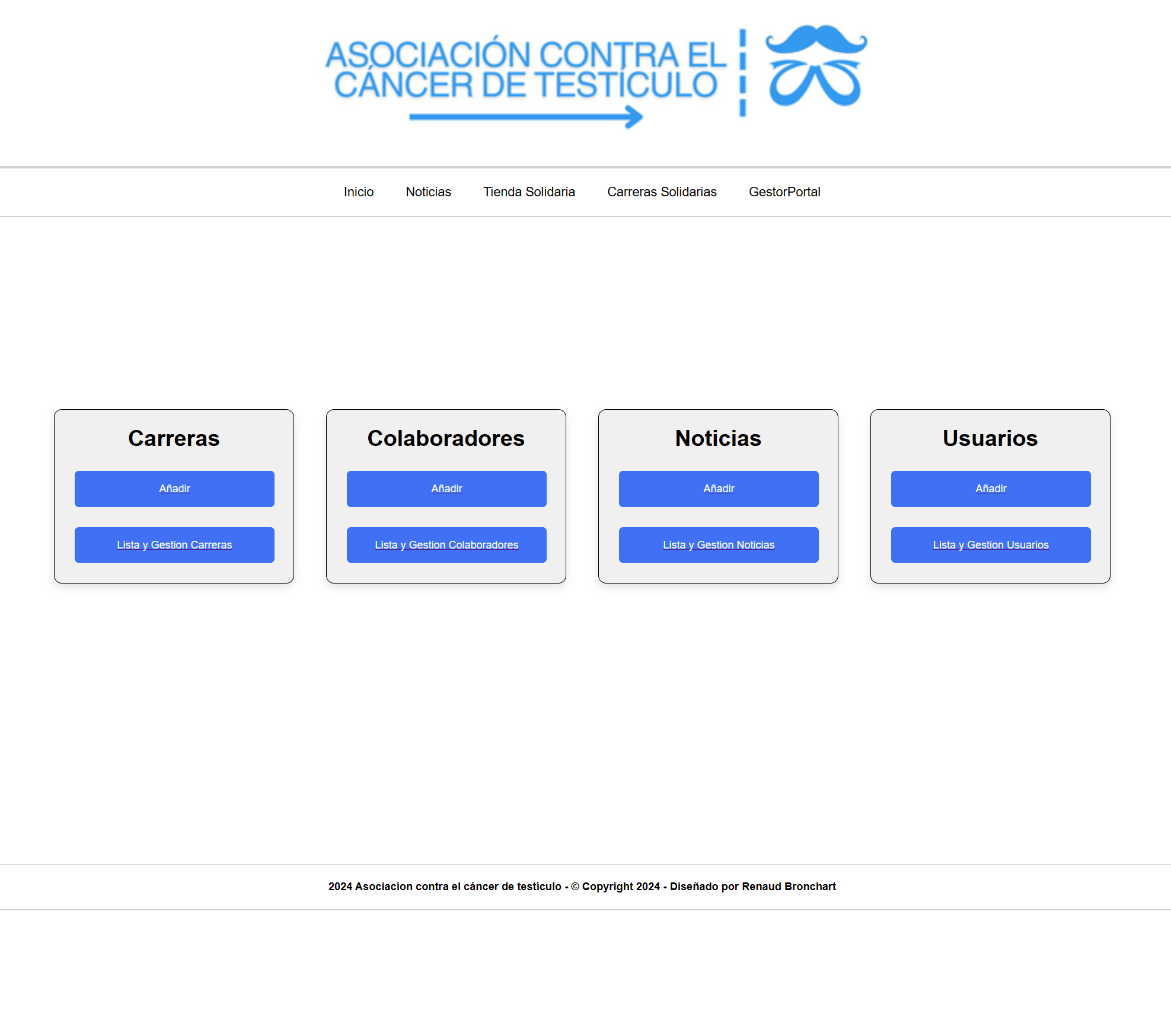
## Descripción General

La estructura del portal se basará en dos partes: en una parte pública (Front-end) y en una privada (Back-end).

La aplicación web permitiría a los usuarios de acceder a las informaciones de asociación(Noticias,Colaboradores, Carreras, etc…)

El back-end es el gestor donde se puede crear, editar y borrar las carreras, las noticias y los colaboradores.





## Objetivos principales

El objetivo principal del proyecto es la construcción de un portal web capaz de publicar eventos, noticias, los colaboradores y que podemos editar, borrar los contenidos que tengamos creados.

Los elementos principales son:

* Un portal web donde el usuario puede acceder a eventos creados.
* Un portal web donde el usuario puede registrase para los eventos creados.
* Un portal web donde el usuario puede ver las noticias de la asociación.
* Un portal web donde el usuario puede ver los colaboradores.
* Gestor de contenido que permite la creación, la modificación y la eliminación de eventos o noticias

# Requisitos

### Requisitos funcionales

* RF1 – El sistema mantendrá la disponibilidad del servicio para garantizar un acceso constante a los recursos
* RF2 - El sistema debe ser capaz de adaptarse rápidamente a los cambios de demande.
* RF 3 La plataforma deberá ser accesible desde cualquier lugar con conexión a internet

### Requisitos específicos

Parte pública

* REP1 – Cualquier visitante de la web podrá visualizar el contenido de la parte publica de la web
* REP2 - Los usuarios anónimos podrán registrarse en el sistema utilizando el mecanismo de registro de usuarios establecido.

Parte privada

* REU1 – Los usuarios podrán inscribirse a unas carreras vía un formulario
* REU2 – Los usuarios registrados podrán acceder su perfil y realizar cambios.

Gestor

* REG 1 - Esta parte será únicamente para los usuarios guardado como administradores
* REG2 - El administrador podrá introducir nuevos colaboradores y nuevas noticias o carreras en la sesión de aplicación web
* REG3 - El administrador podrá editar o eliminar colaboradores, noticias, y carreras y en la sesión de aplicación web

# Análisis

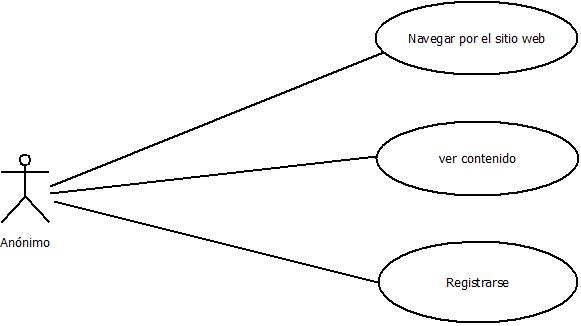
En este capítulo, se describe la fase de análisis del proyecto, su estructura y funcionalidad.

## Casos de uso

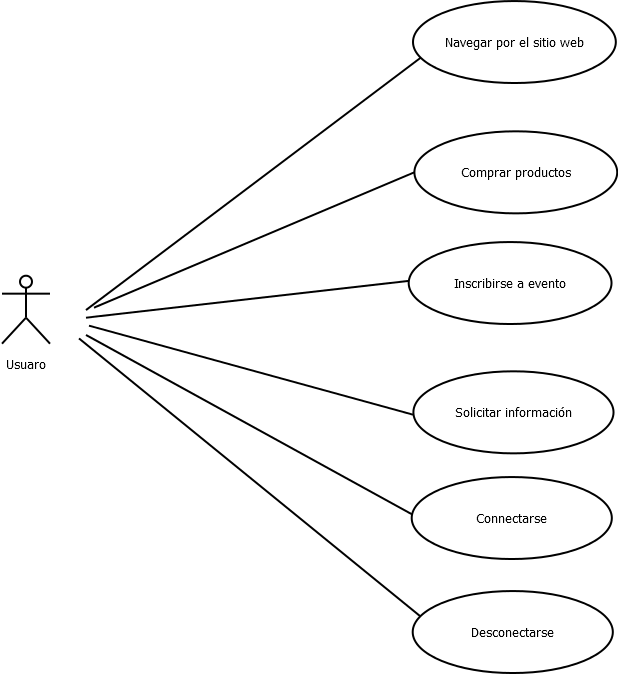
Un **caso de uso** es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso.[[1]](#footnote-1)

A continuación, se muestran los casos de uso de la parte del portal web, con los actores:

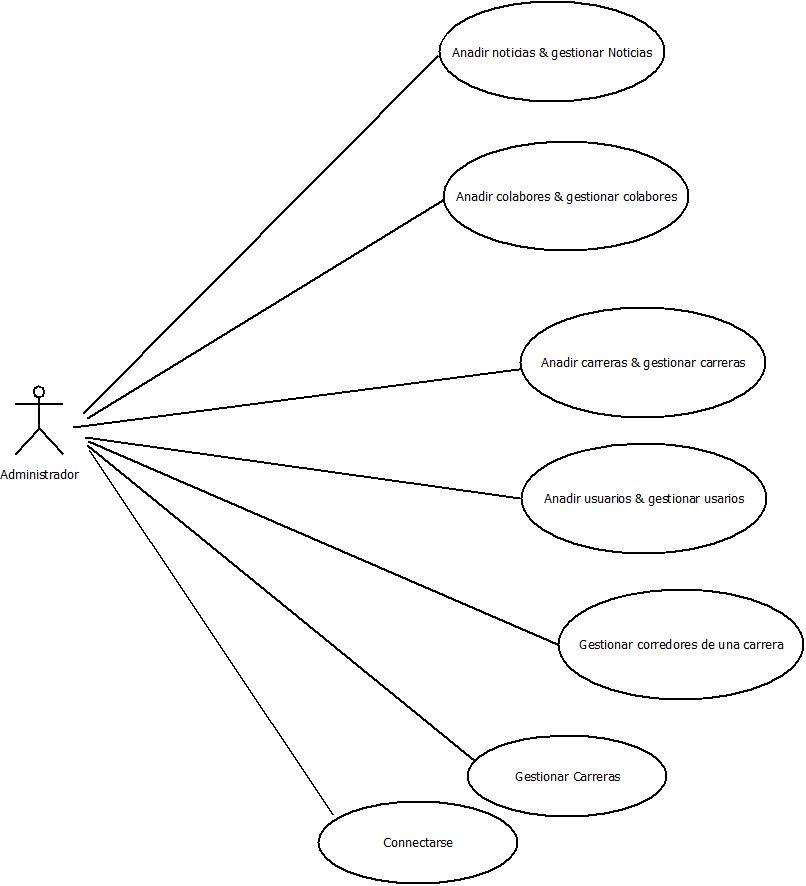
### **Anónimo**: visitante estándar sin estar registrado



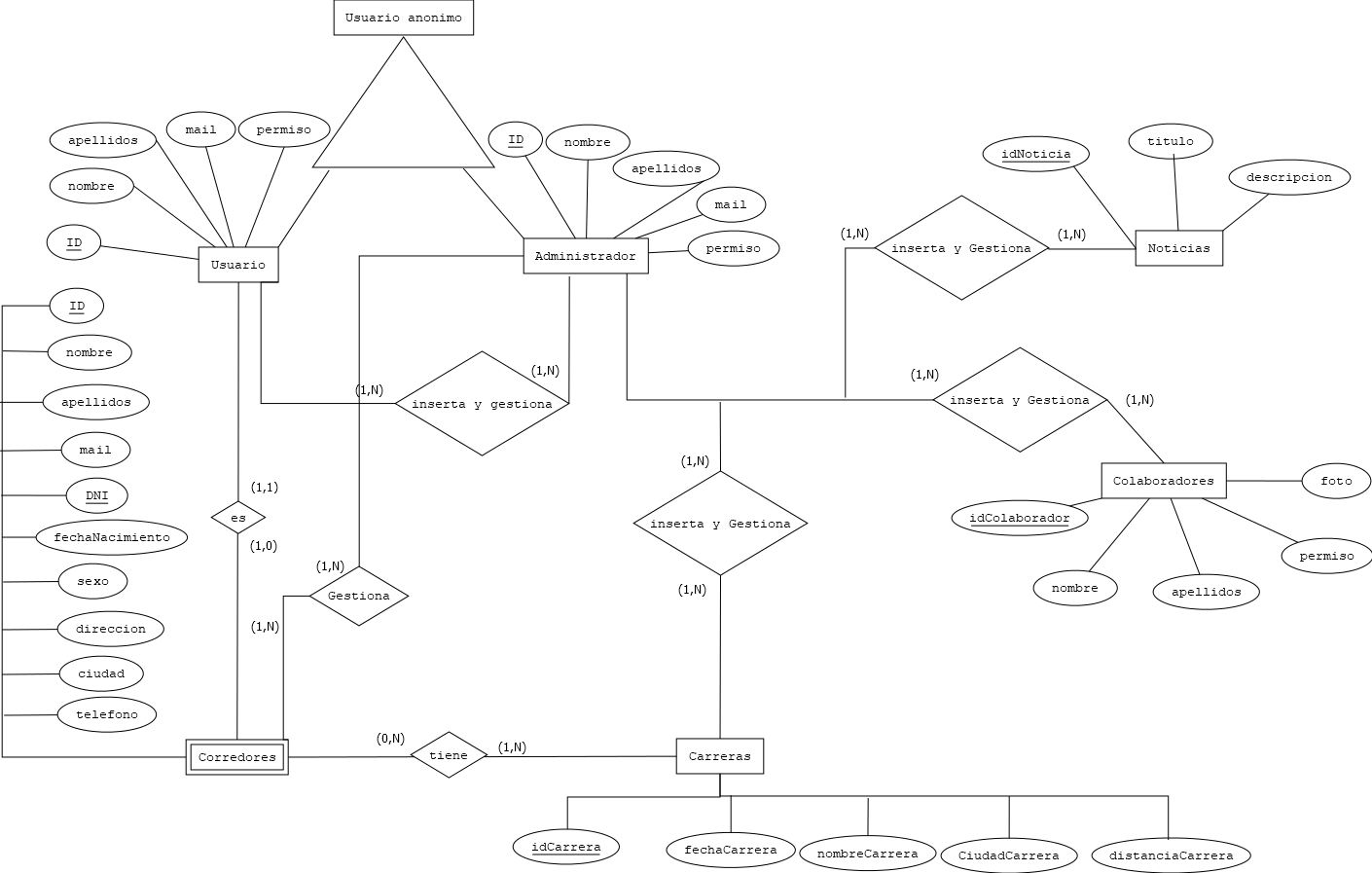
### **Usuario**: Usuario registrado



### **Administrador**: Administrador del portal web



## Diagrama Entidad-Relación



### Paso a Tablas - Asociación contra el cáncer

Usuarios (id, nombre, apellidos, mail, permiso, pass)

Noticias (idNoticia, titulo, descripción)

Colabores(idColaboradores, nombre, apellidos, puesto, foto)

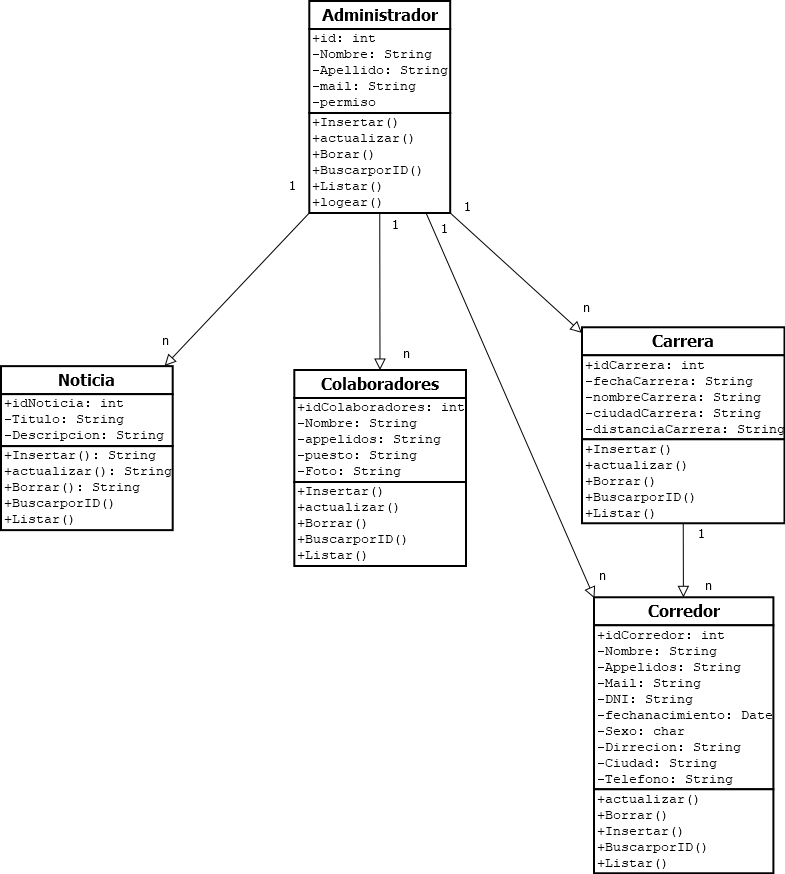
Carreras (idCarrera, fechaCarrera, nombreCarrera, ciudadCarrera, distanciaCarrera )

Corredores(id, nombre, apellidos, mail, dni, fechanacimiento, sexo, dirección, ciudad, teléfono, )

* Id(FK) > por ser PK de Usuarios

### Diagrama de clases

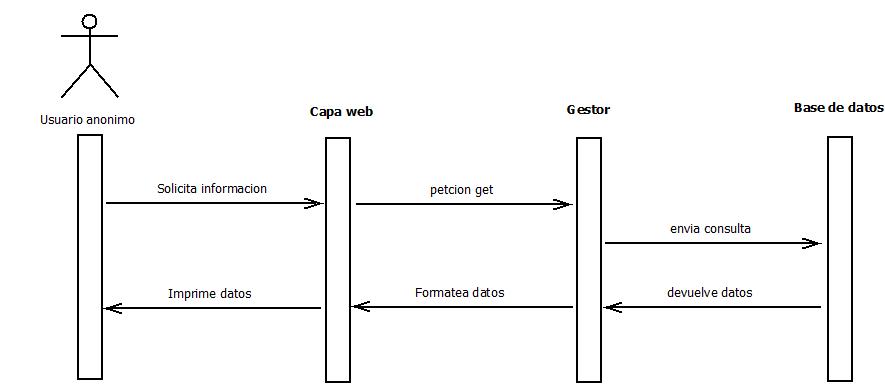
Un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.[[2]](#footnote-2)



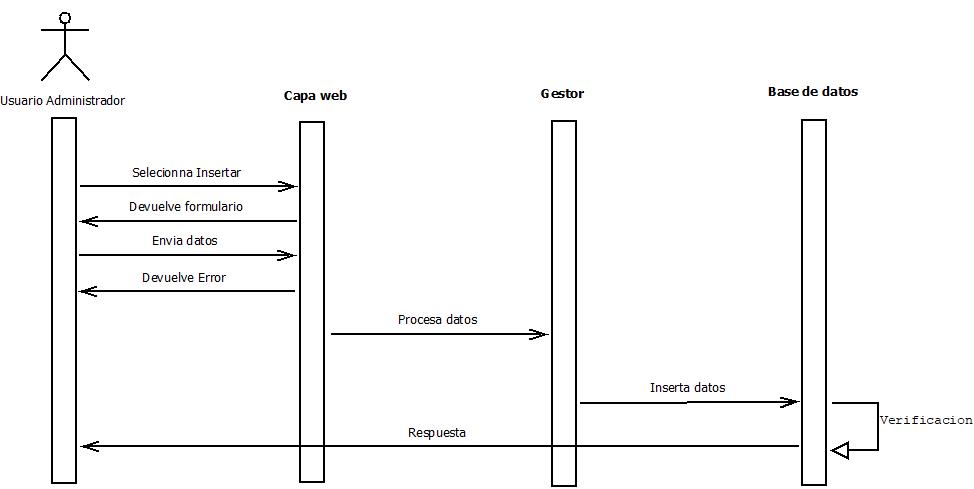
## Diagrama de secuencia

Un diagrama de secuencias muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo.

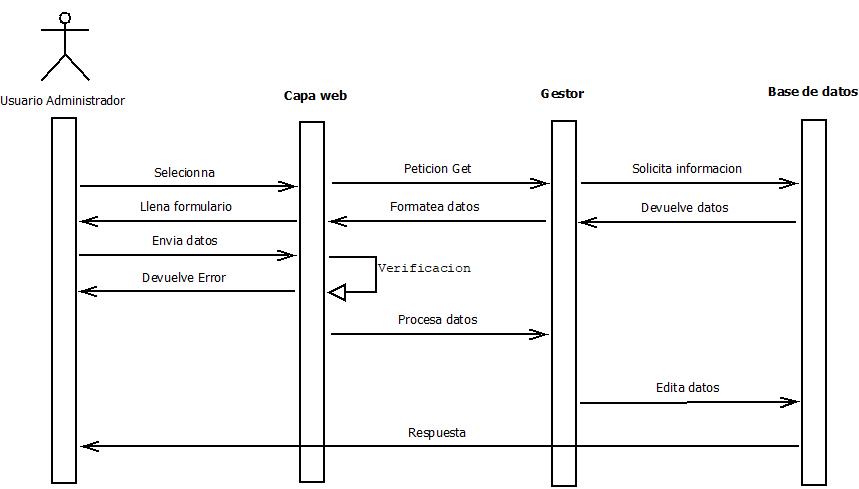
### Visualizar contenido



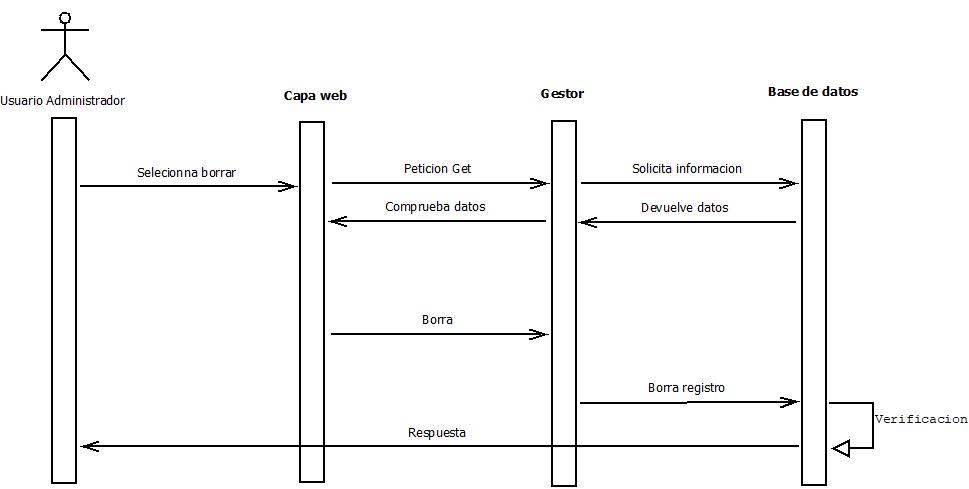
### Insertar contenido



### Editar contenido



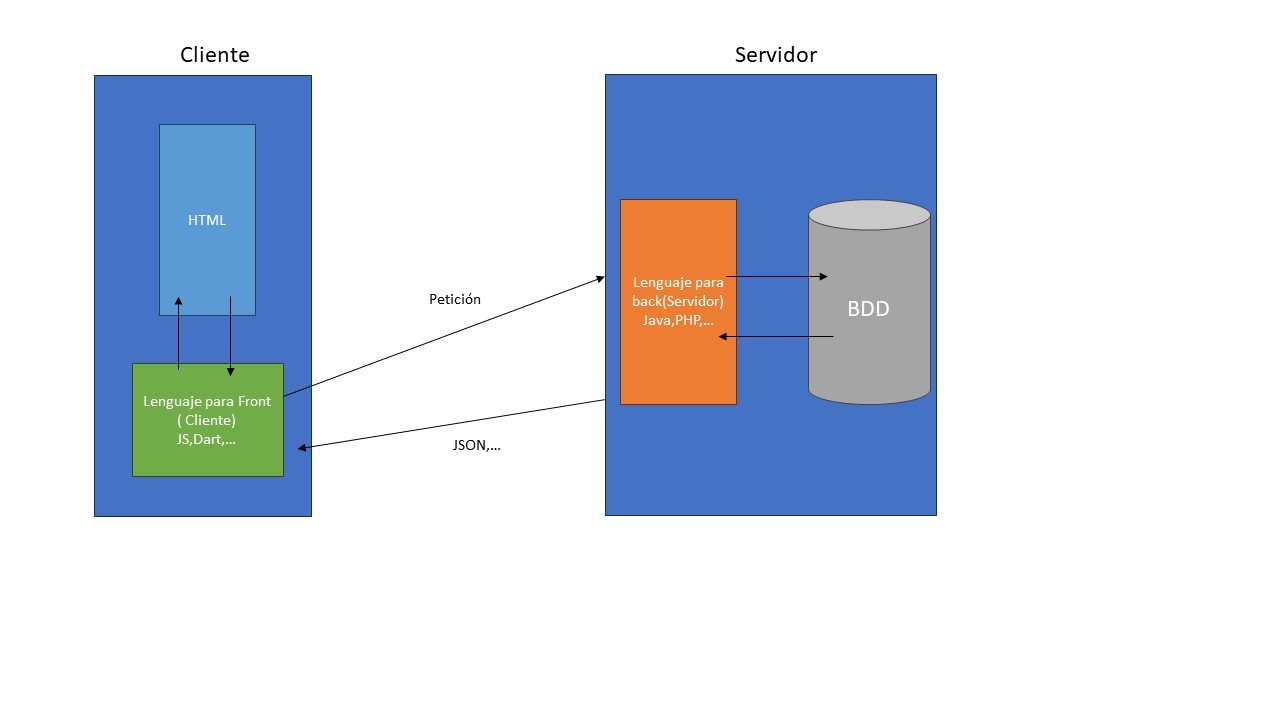
### Borar contenido

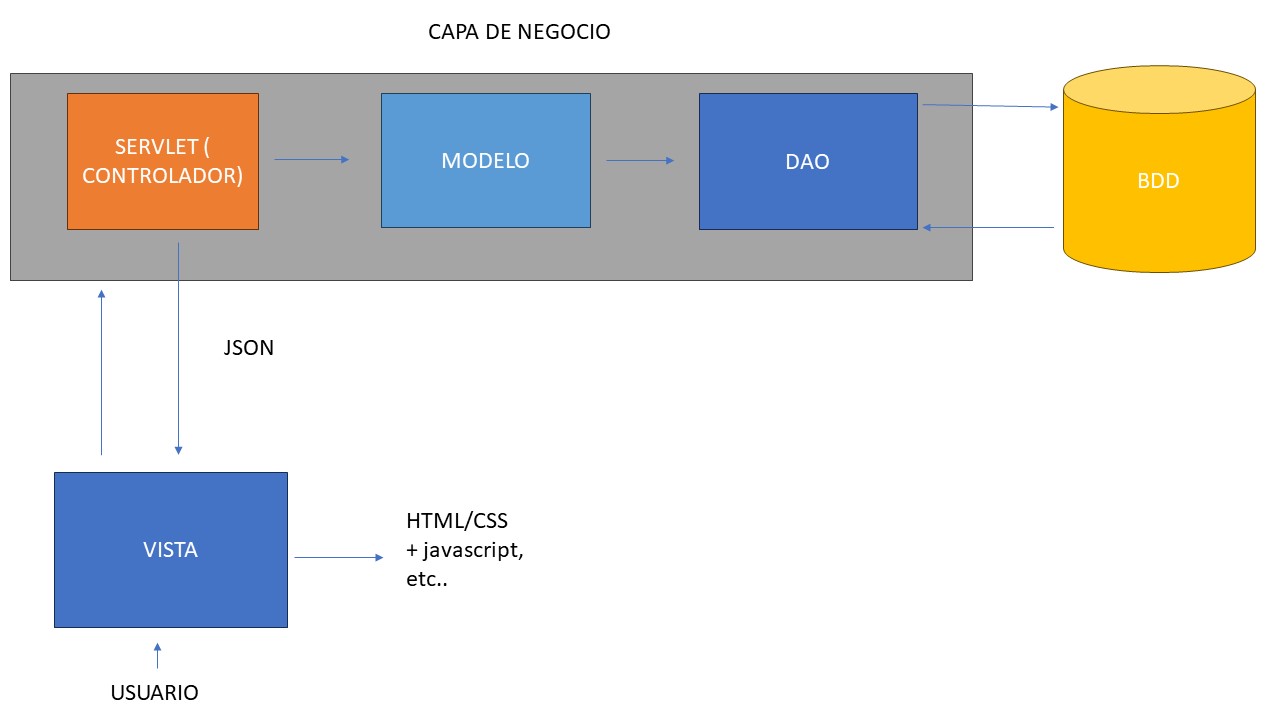


# [Diseño](https://es.wikipedia.org/wiki/Dise%C3%B1o)

## Arquitectura cliente – servidor

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.[[3]](#footnote-3)





-> Usuario inserte datos en el formulario creado en HTML (VISTA)

-> El servlet recibe los datos y los convierte en datos para el modelo y decide lo que va hacer con ellos

-> Las clases dao son los que van a coger los datos y se van a ocupar de gestionar esos datos y mandarlos a la BDD

-> La base de datos devolverá los datos

-> Gracias a Json, se permitirá de poner los datos a la vista para el usuario.

## Capa de presentación

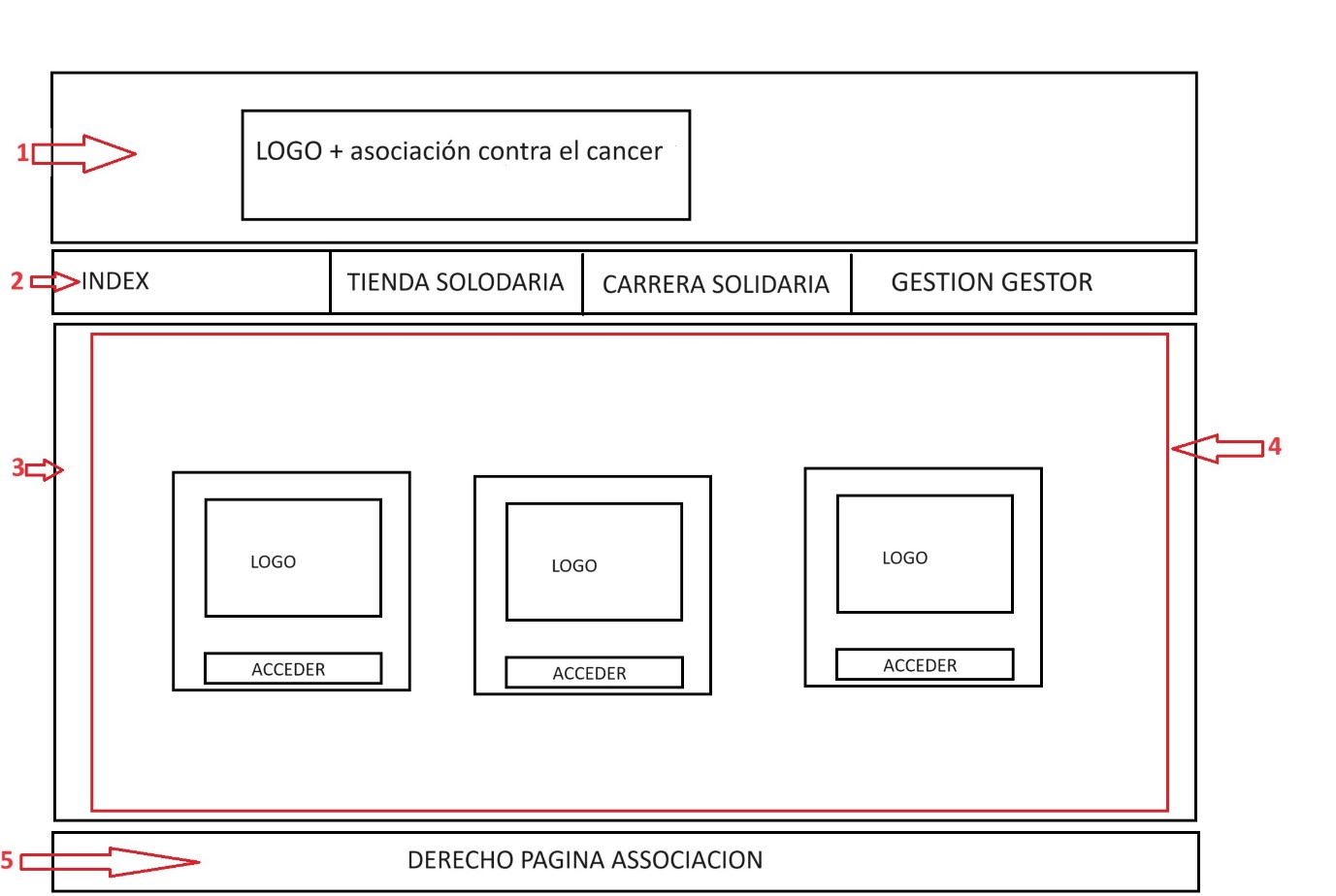
La capa de presentación, también conocida como capa de interfaz de usuario, es la interfaz gráfica de nuestro sistema, que proporciona la interacción entre los usuarios y el sistema.

### Front-end

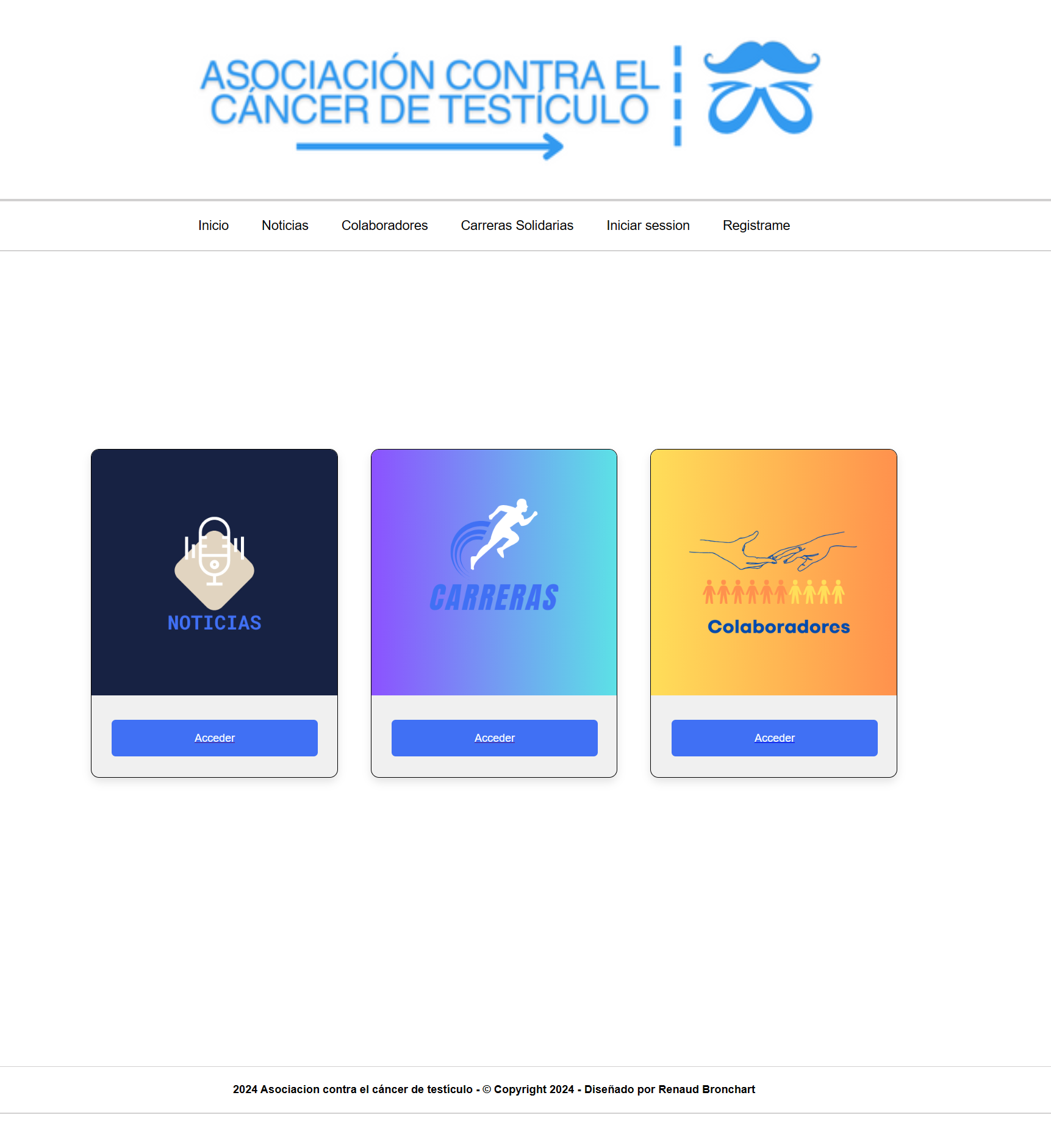
El front-end del portal es lo primero que ve el usuario al entrar en nuestra pagina web.

A continuación, mostraremos el diseño preliminar de la parte visual de nuestra aplicación con su explicación y el resultado final.

1. **CABECERA** : Logo y nombre de nuestra aplicación web
2. **Menú**: Lista de secciones disponible
3. **BODY** : parte principal donde se puede modificar el fondo de la pagina
4. **CUERPO** : parte principal de la vista donde se puede ver elementos como noticias, carreras…
5. **PIE** : datos de la pagina



**Resultado final de la pagina:**

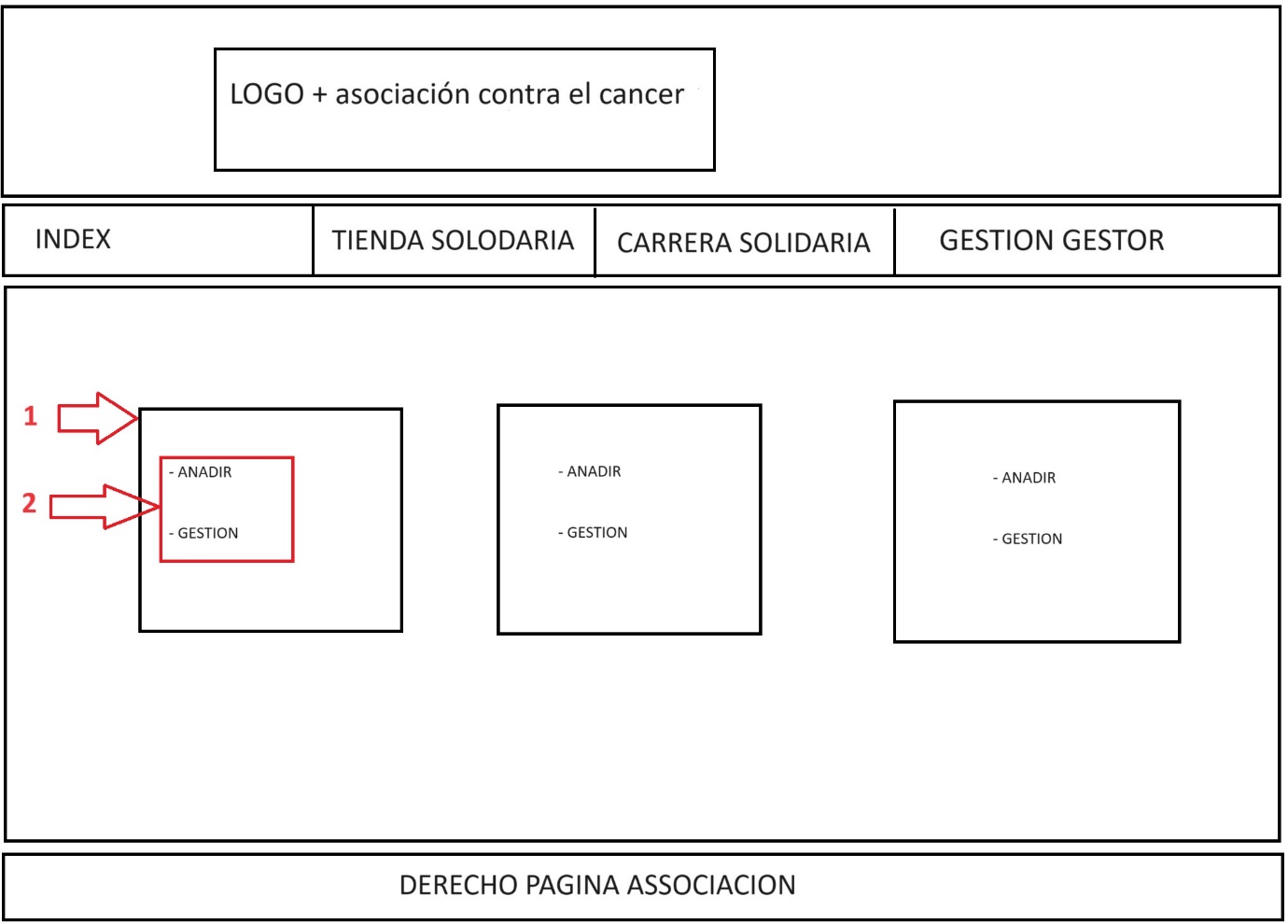


### El Back-end

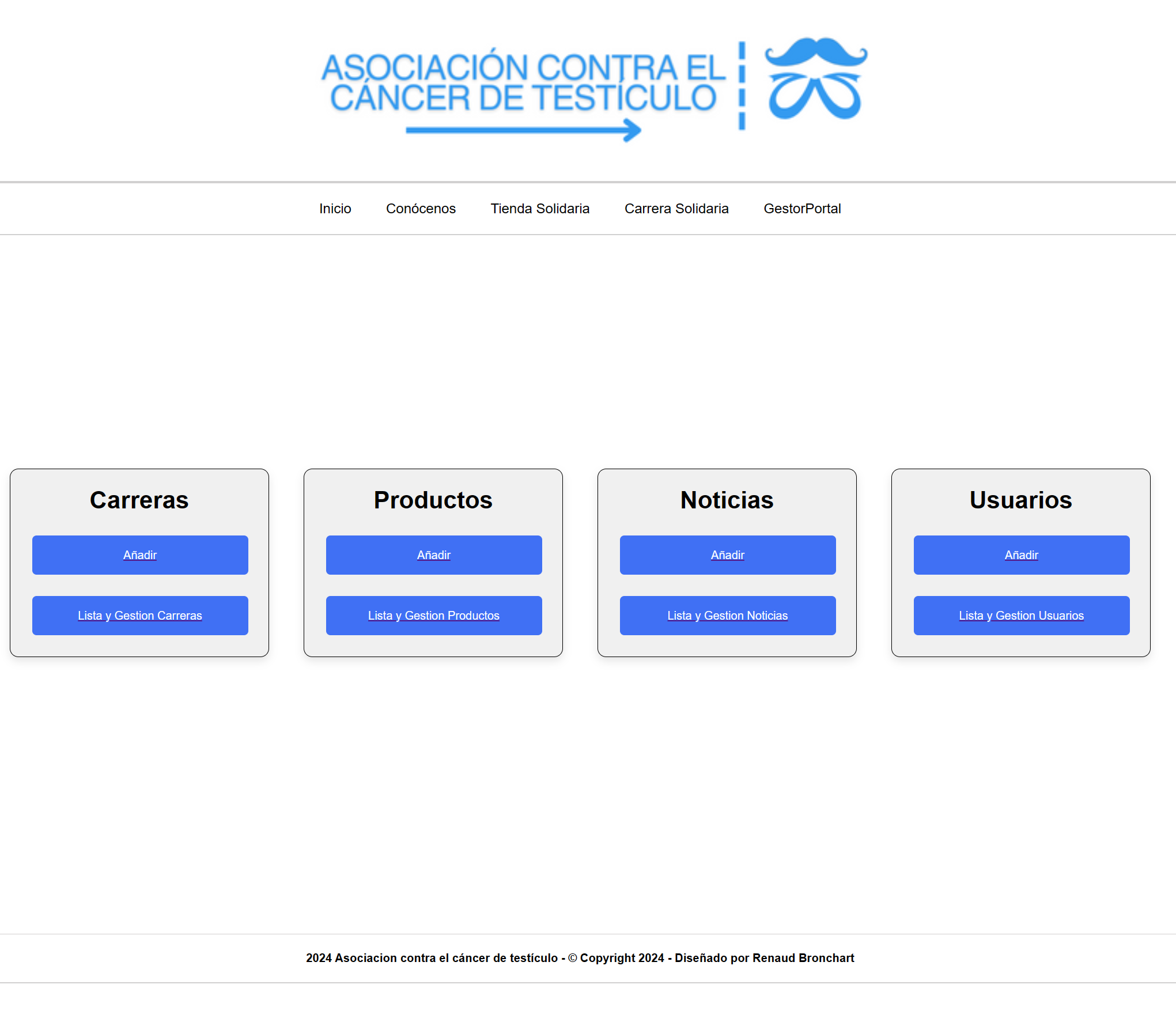
El Back-end es el gestor al que solo tendrán acceso los administradores, mediante el cual podrán mantener actualizado y configurado el portal. Es decir, podrán añadir y gestionar(editar, borrar) las carreras, las noticias, los corredores y los productos.

Mostraremos el diseño preliminar de la parte visual de nuestra aplicación con su explicación y el resultado final.

1. **Menú Gestor :** Menú del gestor donde se puede acceder a todas las secciones del portal
2. **Cuerpo Menú Gestor :** Zona de ladiferentes secciones del menú gestor que permite añadir o gestionar la sección

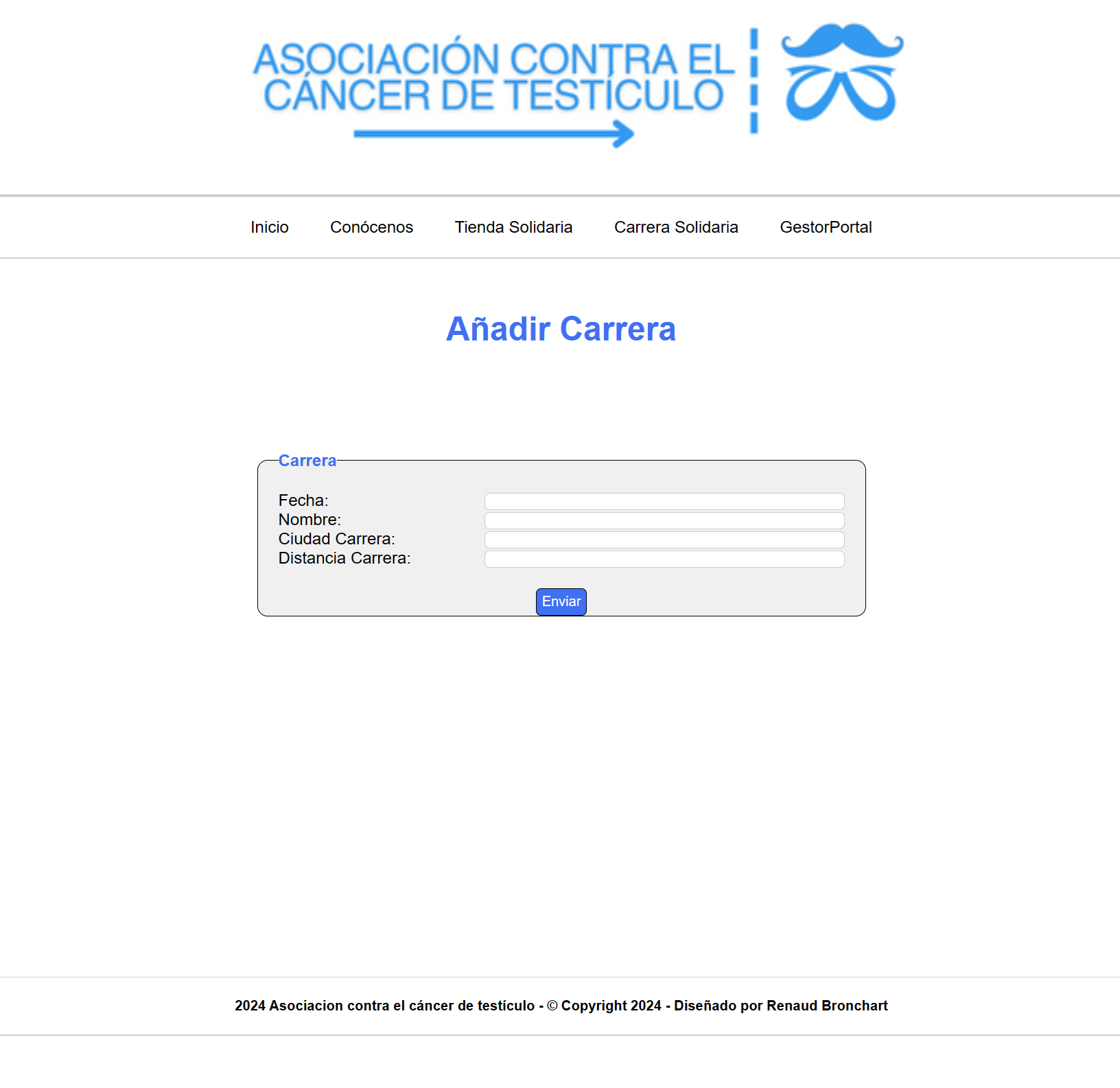


**Resultado final de la página:**



A continuación, se podría ver las diferentes secciones como añadir o gestionar.

**Añadir**



**Lista y Gestión**

1. **Editar :** permite cambiar datos de la carrera
2. **Borrar :** permite borrar una carrera
3. **Lista corredores :** permite ver los corredores de la carrera



# Implementación

## Tecnologías

### HTML

Html (HyperText Markup Language) es un lenguaje de etiquetas usado por los diseñadores y programadores para crear páginas web, este lenguaje es un estándar reconocido por en todo el mundo y cuyas normas las define un organismo sin ánimo de lucro llamado World Wide Web Consortium (http://www.w3.org/). Un documento Html está formado por su contenido y etiquetas que delimitan sus partes pudiendo formatear la página aplicando el diseño o estructura deseada

Actualmente HTML5 es la última actualización de este lenguaje de marcas, aunque realmente es un termino que sirve para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript, sin las cuales ya no se concibe una pagina web.

### CSS

Con las Hojas de estilo CSS (Cascading Style Sheets) podemos dotar a nuestras páginas HTML con diseños mas estéticos, mas hoy en día que con HTML5, la tendencia es usar el menor número de atributos, dejando así todo lo que se refiere a la maquetación, aspecto, diseño, etc. a CSS. Con las hojas de estilo no solo podremos hacer que nuestros textos y fotos se vean del modo que deseemos, además controlaremos toda la estructura de nuestra página, menús, columnas, etc.

### JAVA

Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems.2​3​

El lenguaje de programación Java fue desarrollado originalmente por James Gosling, de Sun Microsystems (constituida en 1983 y posteriormente adquirida el 27 de enero de 2010 por la compañía Oracle),4​ y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a bytecode (clase Java), que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.[[4]](#footnote-4)

### JAVASCRIPT

En contra de los lenguajes vistos hasta ahora, JavaScript (abreviado comúnmente “JS”) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas4 aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. Con él conseguiremos mayor interacción entre las páginas y podemos usarlo junto a HTML creando HTML dinámico DHTML. Otra de las particularidades de javasript es su flexibilidad, así podemos incluirlo en el código hasta en 3 formas diferentes para crear la página

### MySQL

MySQL es un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) de código abierto desarrollado por Oracle. Se ha ganado su lugar en el mundo digital como una base de datos relacional que permite almacenar, organizar y recuperar datos de manera eficiente. MySQL es utilizado por una amplia variedad de organizaciones y aplicaciones en todo el mundo.

Además, utiliza el lenguaje SQL (Structured Query Language) para acceder y manipular los datos.[[5]](#footnote-5)

### JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript.

JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.[[6]](#footnote-6)

## Herramientas utilizadas

### Dia Diagram

Creación de diagrama UM, modelo entidad relación, diagramas de casos de uso

### Eclipse Java Web Developers

Eclipse es un entorno de desarrollo software. Eclipse fue utilizado para el desarrollo del back-end con Java

### Apache Tomcat 10:

Permite la generación de un servidor que gestiona la aplicación java

### Java doc

Documentación de las clases utilizado en eclipse.

### Git

Para almacenar archivos de código y del proyecto

### VISUAL CODE

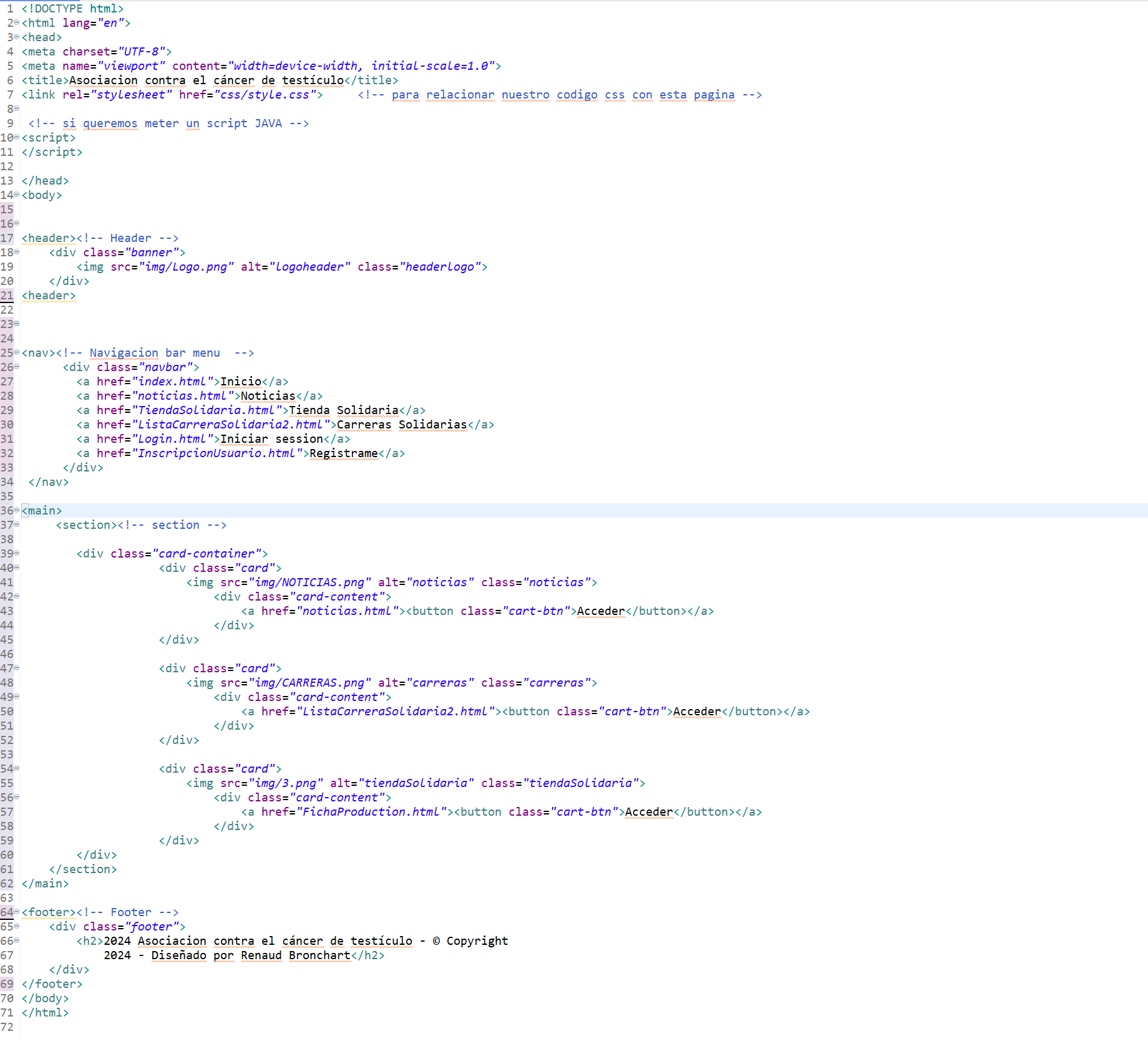
Para el desarrollo del Front-end (HTML+ CSS + JAVASCRIPT)

## Detalles implementación

En esta sección vamos a comentar algunas partes más destacadas del código de nuestro portal.

### Interfaz Front-end (índex )

Índex será la primera página de nuestra aplicación web. A continuación, vamos a ensenar una parte del código HTML utilizado para diseñar la pagina y una parte del código CSS utilizado para permitir un diseño más estético.



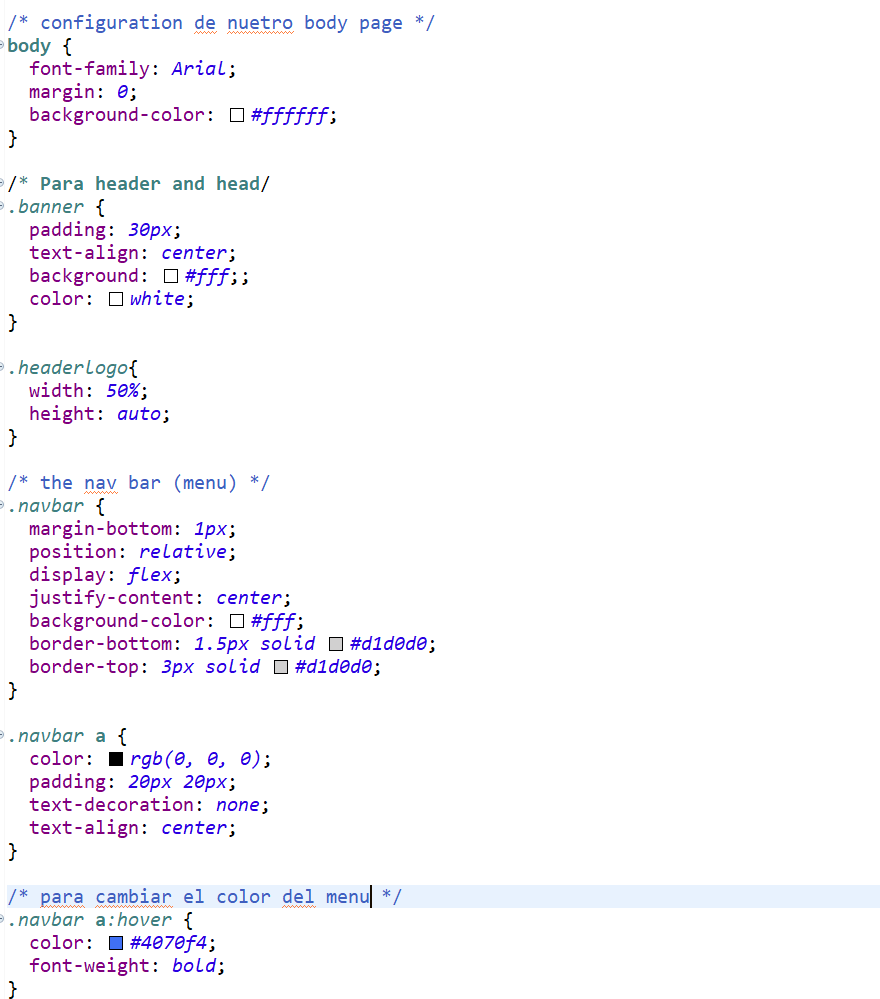
**Léxico de nuestra página web**

* **Html** : Es el elemento raíz del documento. Todo el contenido del sitio web está contenido dentro de este elemento.
* **Head**: Contiene el titulo, y el link que permite relacionar nuestro código CSS y si queremos, los scripts de JavaScript como comentado en la página.
* **Body**: Este elemento será el contenido principal de nuestra página web.
* **Header**: Este elemento será el contenido para contener el logo del sitio.
* **Nav**: Este elemento se utiliza para contener la barra de navegación del sitio web
* **Main**: Este elemento contiene el contenido principal único a la página.
* **Section**: Se utiliza para agrupar contenido relacionado. Dentro de una sección, normalmente se encuentran elementos de encabezado y otros bloques de contenido.
* **Footer**: Este elemento contiene información acerca del documento como autor, derechos de autor y enlaces a políticas de privacidad y términos de servicio.
* **Script**: Este elemento se utiliza para incrustar o referenciar código JavaScript en el documento.

**Léxico no utilizado en esta página web pero útil**

* **Aside**: Se utiliza para contener contenido que está relacionado con el contenido principal, pero que puede ser considerado separado del contenido principal.
* **Label**: Este elemento se utiliza para proporcionar una descripción a los elementos de entrada de un formulario.

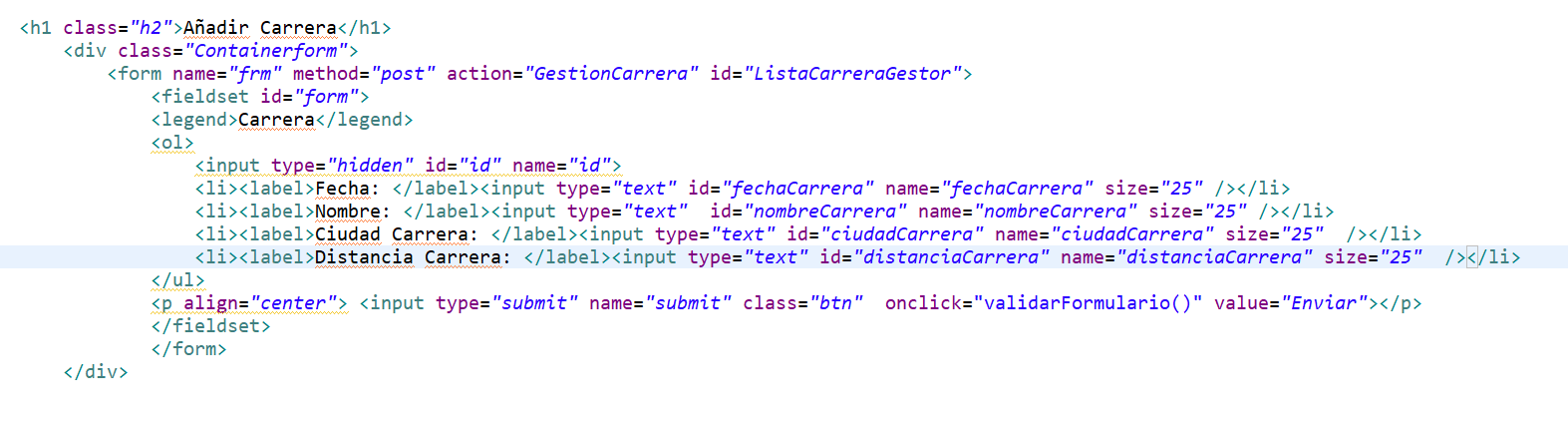
En la siguiente página, se podría ver una parte del código CSS comentada para entender mejor cada código para cada parte de nuestra pagina web.



### Añadir Carrera

Uno de los procedimientos más habituales en el portal es la inserción de contenido a través de un formulario, creado a tal efecto, un proceso de validación en JavaScript y guardado en la base de datos para que la información sea persistente.   
Como ejemplo hemos cogido “Anadir una carrera”.

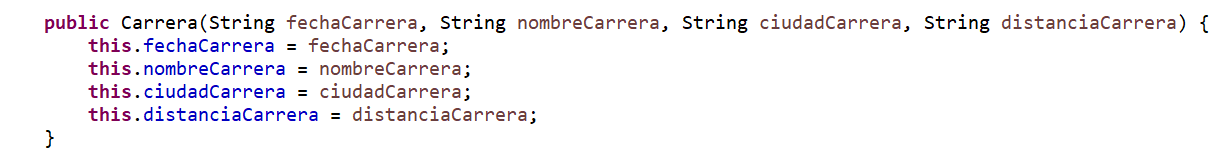
Un formulario fue creado para permitir al Gestor de crear una carrera.

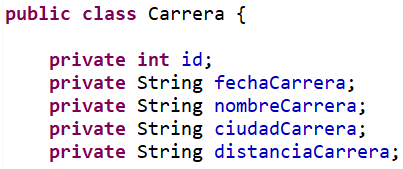
El formulario HTML es el siguiente :  
  


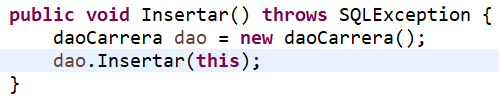


El proceso de inserción de una carrera se iniciará por rellenar el formulario por el gestor.  
  
Este envía una petición por post (method=”post”) como se puede ver en la captura del formulario a un Servlet que hemos creado y que se llama GestionCarrera.

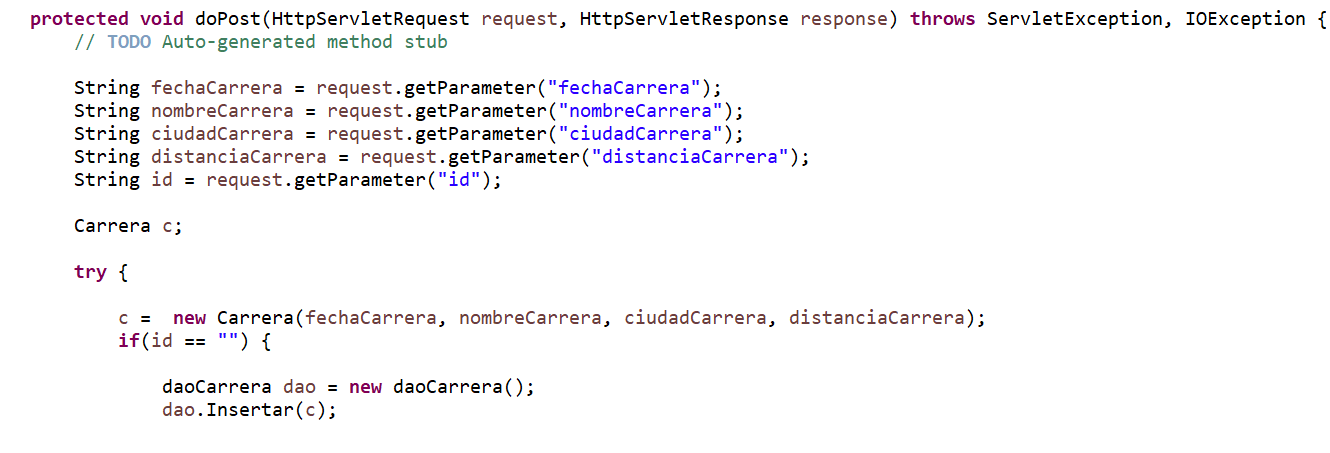
El constructor de la clase carrera va a crear un objeto con los datos de la carrera y mediante el método de insertar se va a proceder a la inserción de los datos.

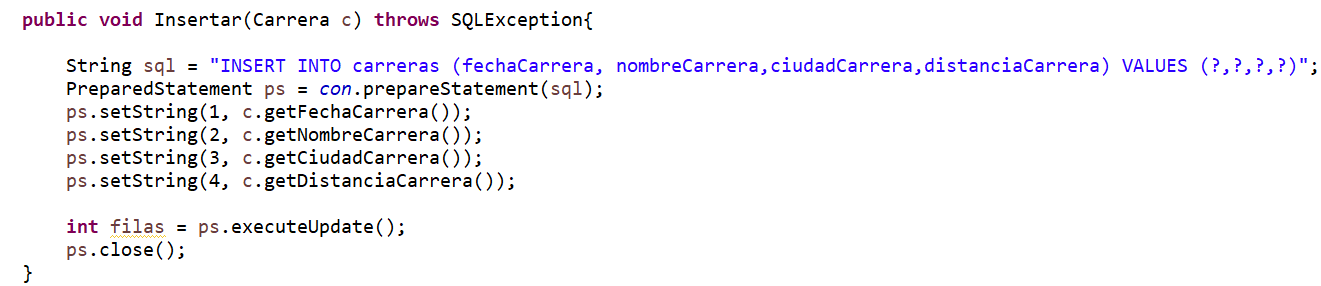




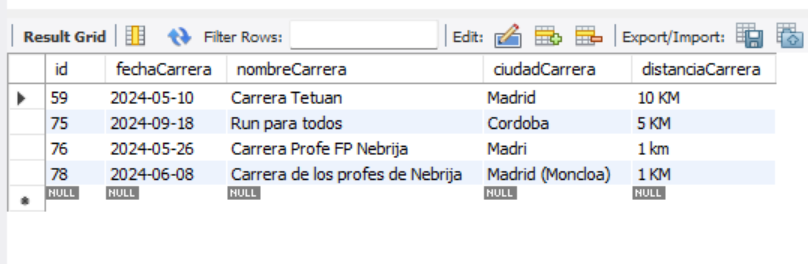


Mejor hacerlo con Singleton

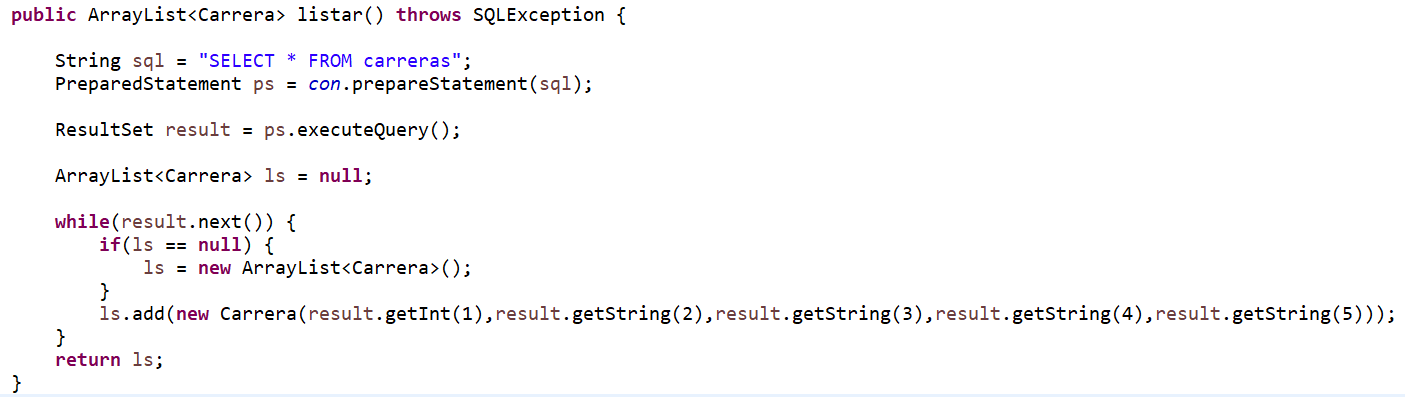




Esos datos van a poder estar en la BDD como se puede ver con el ID:78 que es lo que hemos añadido cuando el gesto ha completado el formulario.

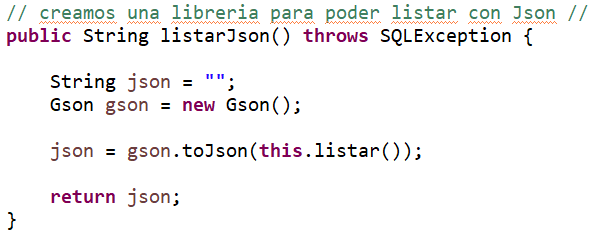


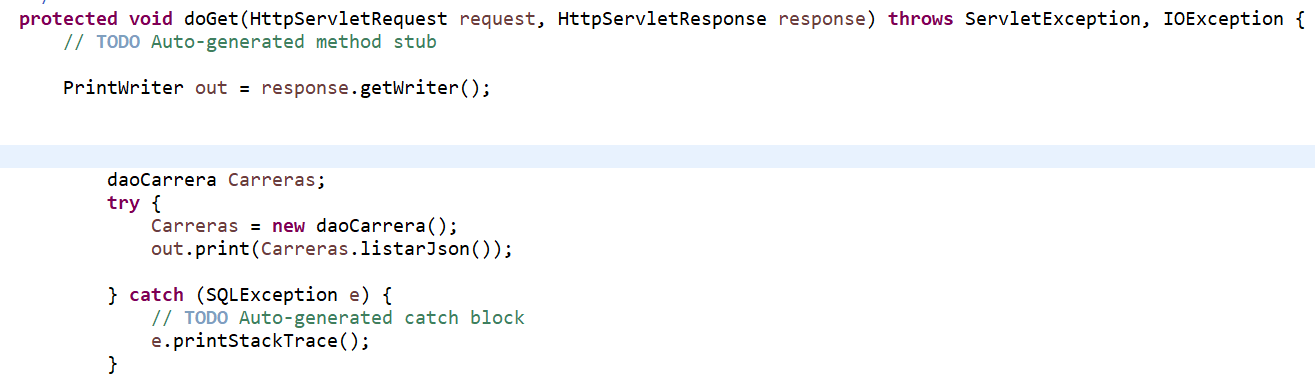
### Listar carreras

Un método fue creado para ir a la base de datos que permite devolver un Arraylist de la carreras creadas y insertadas en la BDD.  


Como fue explicado en el punto de **3.1 Arquitectura cliente – servidor**, utilizamos una librería JSON .

Es decir, un método String fue creado para poder listar los datos en formato Json.





Para poder mostrar esta lista, necesitamos pasar los datos de Json via HTML.  
  
Por eso, vamos a utilizar un script con JavaScript que permite de mostrar los datos.  
  
Entonces, cuando la pagina HTML se va a cargar, para poder mostrar la lista de carreras, unas funciones en JavaScript fue creado.



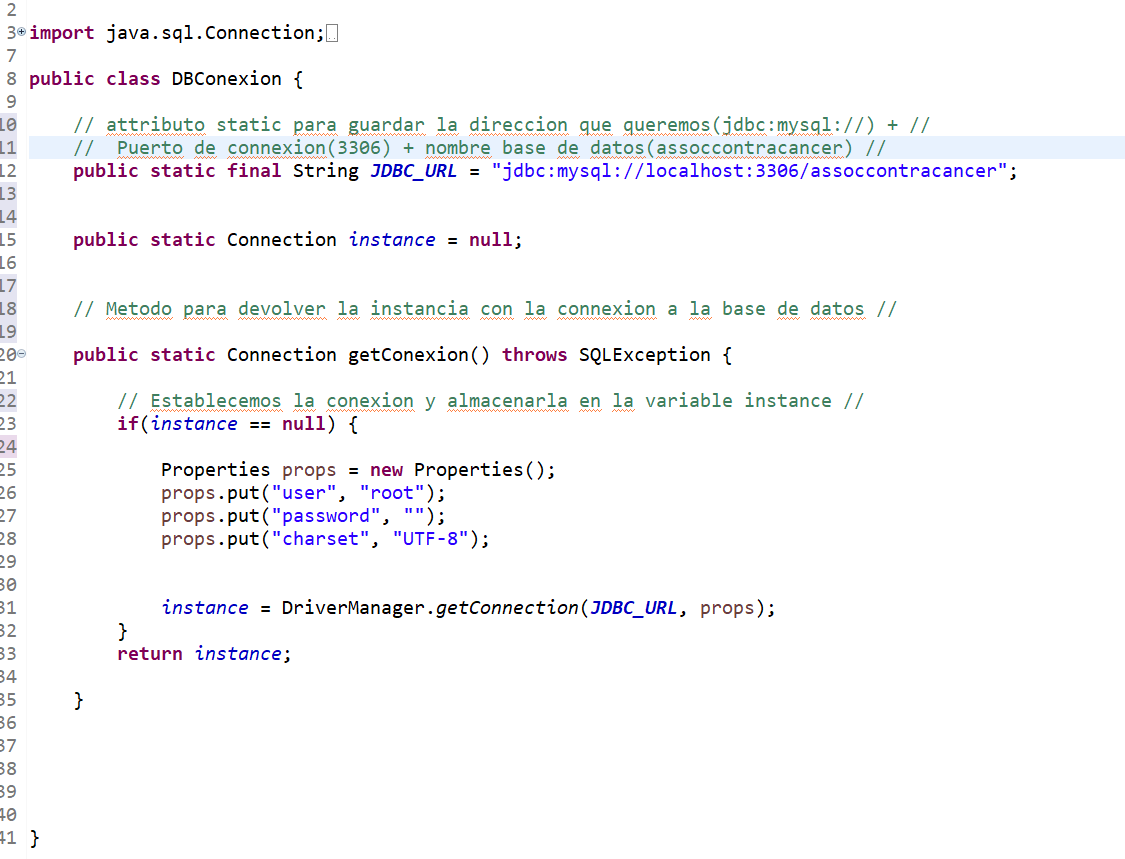
Gracias a este código de JavaScript, y los pasos hechos anteriormente, vamos a poder mostrar la lista de carreras que el gestor ha creado.



### Connexion a la base de datos

Hemos creado un Objeto conexión genérico que va a permitir la conexión a la BDD.

Hacemos la conexión a través de una instancia que gestiona la clase.

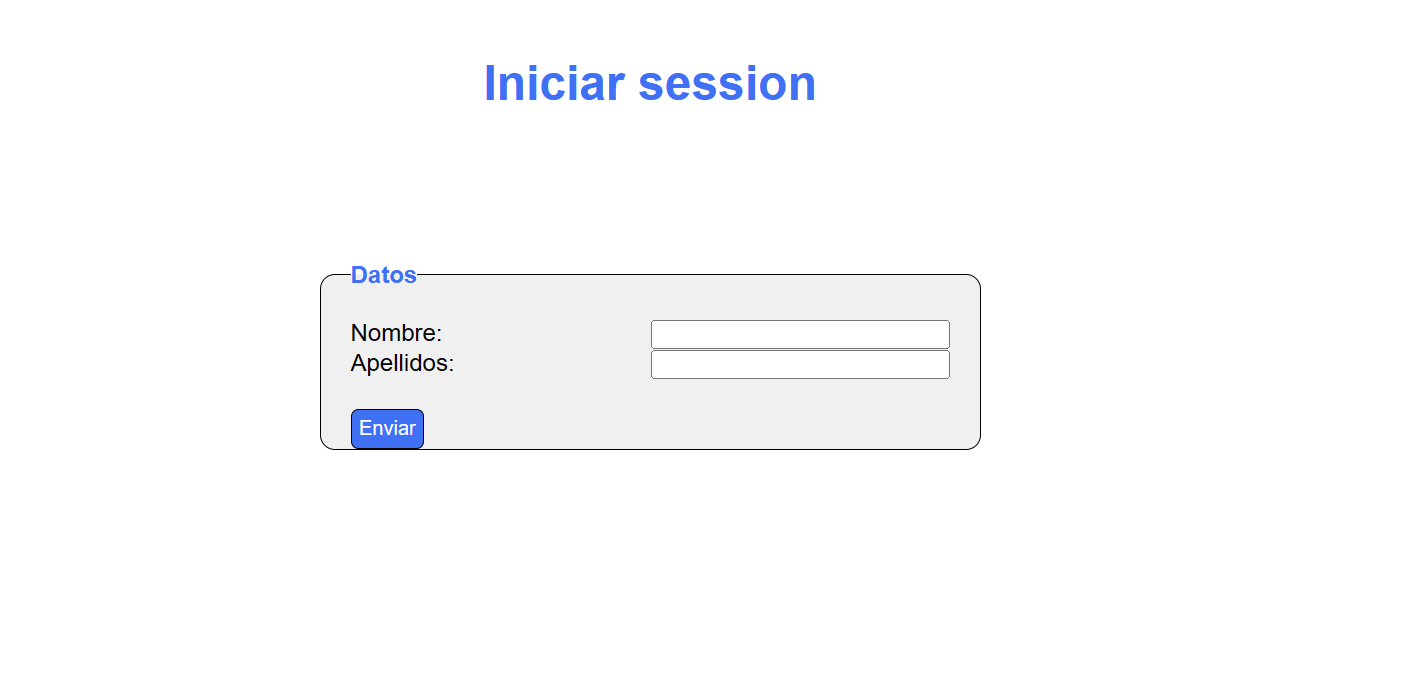


### DAO

Utilizamos el patrón DAO( Data Access object) en nuestro proyecto.  
El DAO va a permitir la interactuación con la base de datos.( Consultas, inserciones, actualización, eliminación de datos).

### Zona usuarios

Una zona usuario ha sido creado y que permite distinguir un usuario administrador y un usuario registrado.

  
  
Para acceder a esta parte, el usuario debería registrarse vía un formulario.



# Conclusiones

Participar y terminar este proyecto es un reto importante en muchos sentidos y una satisfacción porque significa que la mayoría de los requisitos fueron aplicados y que he cumplido las diferentes etapas del proyecto para generar un producto final.

La realización del proyecto fue una terea difícil debito a los diferentes obstáculos encontrados como la gestión de tiempos, la metodología de trabajo, poner en práctica los conocimientos aprendidos.

Antes de empezar el proyecto, muchos aspectos de html,CSS,JAVA,MYSQL, etc, eran complejos.

Durante el proceso he podido aprender, comprender y perfeccionar más los conocimientos adquiridos durante el curso. También fue una oportunidad para aplicar los conocimientos y habilidades.

Además de utilizar los diferentes documentos que tenemos a disposición gracias al curso, una propia investigación personal fue necesaria, y uno de los recursos utilizados fue internet.

En conclusión, la realización de este proyecto ha permitido meter en práctica los conocimientos obtenidos durante todo el año y aprender como manejar un proyecto con sus problemas y acercase el más posible a los proyectos que se puede tener en una empresa.

# Bibliographia

Para la realización de esta memoria se han consultado las siguientes fuentes de contenido**.**

* https://es.wikipedia.org/wiki/Caso\_de\_uso
* <https://es.wikipedia.org/wiki/Diagrama_de_clases>
* [UML: Diagrama de Secuencia – INGENIERÍA DEL SOFTWARE (wordpress.com)](https://ingsotfwarekarlacevallos.wordpress.com/2015/07/07/uml-diagrama-de-secuencia/)
* [Cliente-servidor - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/Cliente-servidor#:~:text=La%20arquitectura%20cliente%2Dservidor%20es,servidor%2C%20quien%20le%20da%20respuesta.)
* [¿Qué es MySQL? Explicación y características | Arsys](https://www.arsys.es/blog/mysql)
* [JSON](https://www.json.org/json-es.html)

1. Fuente : Wikipédia (ver bibliografía). [↑](#footnote-ref-1)
2. Fuente : Wikipédia (ver bibliografía). [↑](#footnote-ref-2)
3. Fuente : Wikipédia (ver bibliografía). [↑](#footnote-ref-3)
4. Fuente Wikipedia Java [↑](#footnote-ref-4)
5. Fuente Arsys mysql [↑](#footnote-ref-5)
6. Fuente Json.org json [↑](#footnote-ref-6)