



# Servicios y Sistemas Web

---

Manuel Méndez Calvo (manmend)

Miranda Silva del Valle (mirsilv)

Pablo Verdejo Santana (pabverd)

Édgar Díez Alonso (edgdiez)

19 de mayo de 2021

## Índice

<b>1. Presentación del equipo y estimación del trabajo realizado</b>	<b>3</b>
<b>2. Introducción: descripción del sistema web.</b>	<b>3</b>
<b>3. Usuarios objetivo.</b>	<b>3</b>
<b>4. Especificación de requisitos</b>	<b>3</b>
4.1. Funcionales . . . . .	3
4.1.1. Cliente . . . . .	3
4.1.2. Gestor . . . . .	4
4.1.3. Ambos actores . . . . .	4
4.2. No funcionales . . . . .	4
<b>5. Modelo del dominio y estructura de la base de datos</b>	<b>5</b>
5.1. Esquema relacional. . . . .	6
<b>6. Mapa del sitio web</b>	<b>7</b>
<b>7. Descripción de la funcionalidad aplicada</b>	<b>8</b>
7.0.1. Casos de uso implementados del gestor . . . . .	8
7.0.2. Casos de uso implementados del cliente . . . . .	8
7.0.3. Casos de uso implementados para ambos actores . . . . .	8
<b>8. Conclusiones</b>	<b>8</b>

---

## Índice de figuras

1. Modelo Entidad Relación . . . . .	5
2. Esquema Relacional de la base de datos . . . . .	6

## 1. Presentación del equipo y estimación del trabajo realizado

- Édgar Díez Alonso (edgdiez) 40 %
- Manuel Méndez Calvo (manmend) 20 %
- Miranda Silva del Valle (mirsilv) 20 %
- Pablo Verdejo Santana (pabverd) 20 %

Como nota aclaratoria queremos decir que hemos hablado entre todos los miembros del grupo, y creemos que el computo total final debería ser de 25 % para cada uno. Consideramos que el trabajo a lo largo de las entregas finalmente ha sido equitativo entre todos los miembros.

## 2. Introducción: descripción del sistema web.

El sistema que hemos desarrollado es una plataforma de gestión de anuncios de turismo. Tenemos 2 tipos de clientes, el **Gestor** y el **Cliente**. El gestor puede crear anuncios de ofertas (ya sean de desplazamiento, alojamiento y/o turismo), los cuales serán añadidos a la base de datos, que almacenará su historial de creación. Los clientes, por otra parte, podrán contratar los servicios ofertados por esos anuncios, los cuales serán añadidos a su historial de compra.

Tenemos 3 tipos de anuncios diferentes, **Desplazamiento**, **Alojamiento** y **Turismo**, los cuáles ofertan los servicios que su nombre indica.

## 3. Usuarios objetivo.

Como se ha comentado en la sección anterior, tenemos 2 tipos de clientes, el **Gestor** y el **Cliente**. Cada uno tendrá su login y registro por separado, para lo cual aportaran datos diferentes. Además tendrán funcionalidades diferentes a la hora de interactuar con la web, siendo el gestor el que creará los anuncios que podrá comprar el cliente en las diferentes categorías. Los gestores podrán acceder a las diferentes páginas de anuncios, pero no será posible que compren ninguno a no ser que se registren e inicien sesión como cliente.

## 4. Especificación de requisitos

### 4.1. Funcionales

Dividiremos los **Requisitos funcionales** dependiendo de si el actor es **Cliente** o **Gestor**

#### 4.1.1. Cliente

- **Elegir plan de viaje:** El sistema deberá permitir al actor cliente seleccionar una oferta
- **Correo de confirmación:** El sistema deberá comunicar al actor cliente la confirmación de compra de una de las ofertas
- **Nueva cuenta:** El sistema deberá permitir al actor cliente crear una nueva cuenta
- **Asegurar identificación de clientes:** El sistema deberá asegurar que al seleccionar una oferta, el actor cliente esté identificado

- **Ofertas anteriormente seleccionadas:** El sistema deberá permitir al actor cliente revisar las ofertas anteriormente seleccionadas
- **Aplicar filtros:** El sistema deberá permitir al actor cliente aplicar el filtro de ubicación a sus búsquedas.
- **Pop up:** El sistema deberá mostrar recomendaciones de alojamiento y desplazamiento al actor cliente al elegir una oferta de turismo

#### 4.1.2. Gestor

- **Crear un nuevo anuncio:** El sistema deberá permitir al actor gestor crear un nuevo anuncio
- **Revisar anuncio:** El sistema deberá permitir al actor gestor revisar todos los anuncios subidos

#### 4.1.3. Ambos actores

- **Cambiar idioma:** El sistema deberá permitir al actor cliente y al actor gestor cambiar el idioma
- **Identificación de cliente:** El sistema deberá permitir al actor cliente y al actor gestor identificarse
- **Registro de cliente:** El sistema deberá permitir al actor cliente y al actor gestor registrarse con el rol deseado.

#### 4.2. No funcionales

- **Anuncios únicos:** El sistema deberá identificar cada anuncio de forma única
- **Base de datos:** El sistema usará JDBC (Derby) como base de datos
- **Google maps:** El sistema deberá ofrecer links a los sitios anunciados siempre que se permita
- **Distinción de logins:** El sistema deberá permitir iniciar sesión de forma diferente a gestores y clientes
- **Distribución de la página:** El sistema incluirá el logo de la empresa que funcionará como un botón de redirección a la página principal
- **Anuncios de gestores:** El sistema solo permitirá al actor gestor crear nuevos anuncios
- **Distinción de registros:** El sistema deberá permitir registrar una cuenta de manera diferente a gestores y clientes

## 5. Modelo del dominio y estructura de la base de datos

Hemos decidido representar el modelo de dominio en conjunto con la estructura de la base de datos, ya que con un único diagrama podemos representar la información de ambos apartados. En el esquema se puede ver cómo se relacionan nuestros diferentes usuarios con los anuncios y sus históricos, y los atributos que tendrá cada clase.

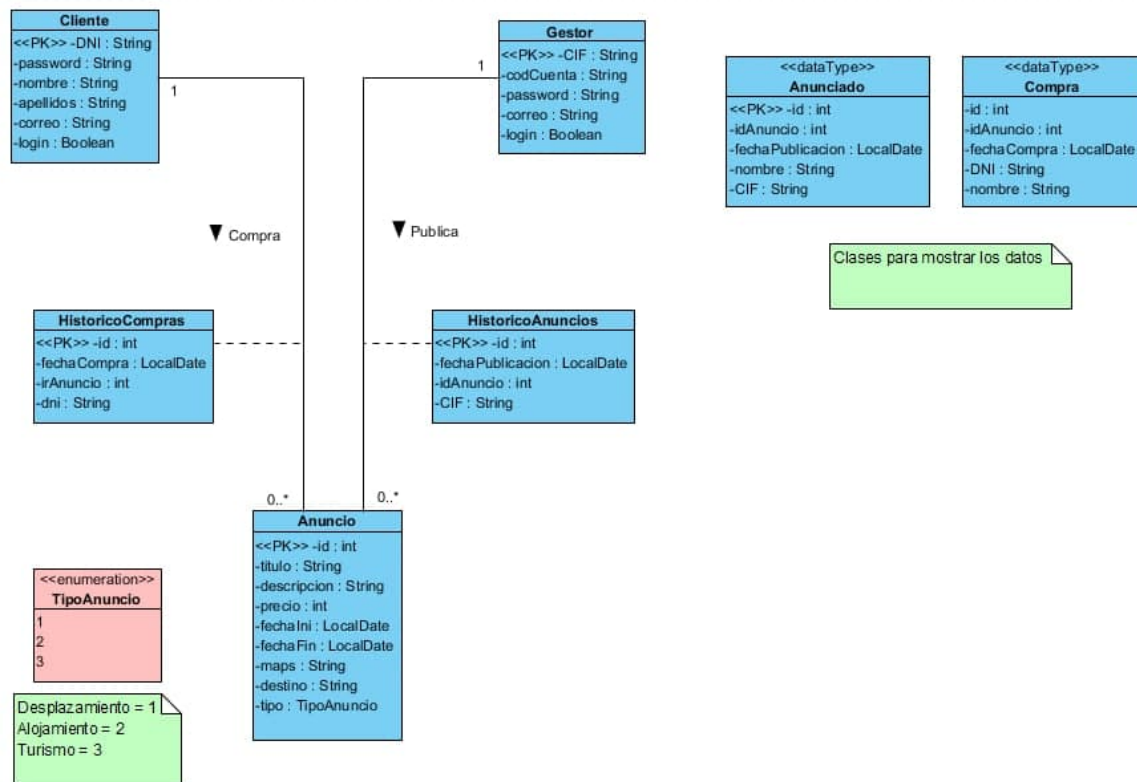


Figura 1: Modelo Entidad Relación

### 5.1. Esquema relacional.

El esquema relacional permite establecer las relaciones existentes entre las distintas tablas y sus claves primarias y foráneas. Además, adjuntamos el Modelo de Dominio de la Base de Datos en el apartado dedicado a ello.

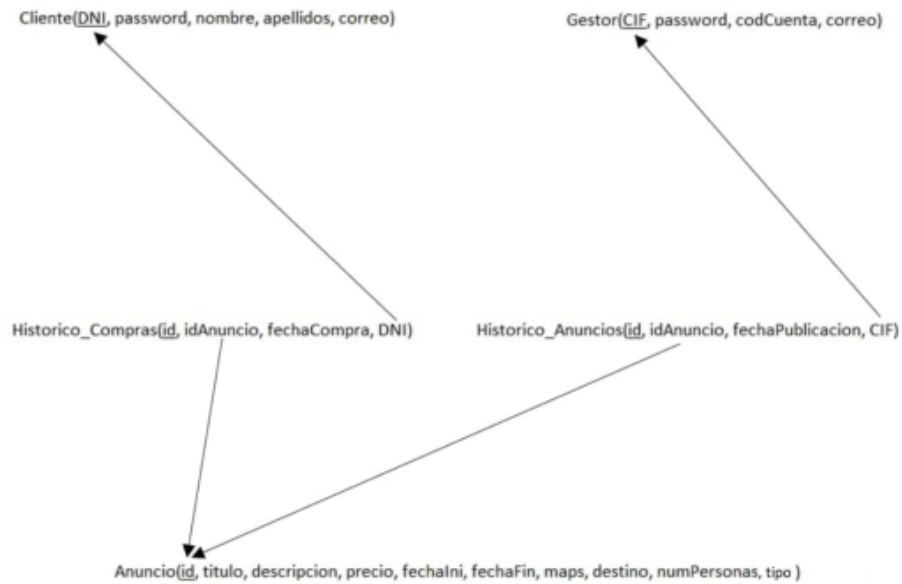
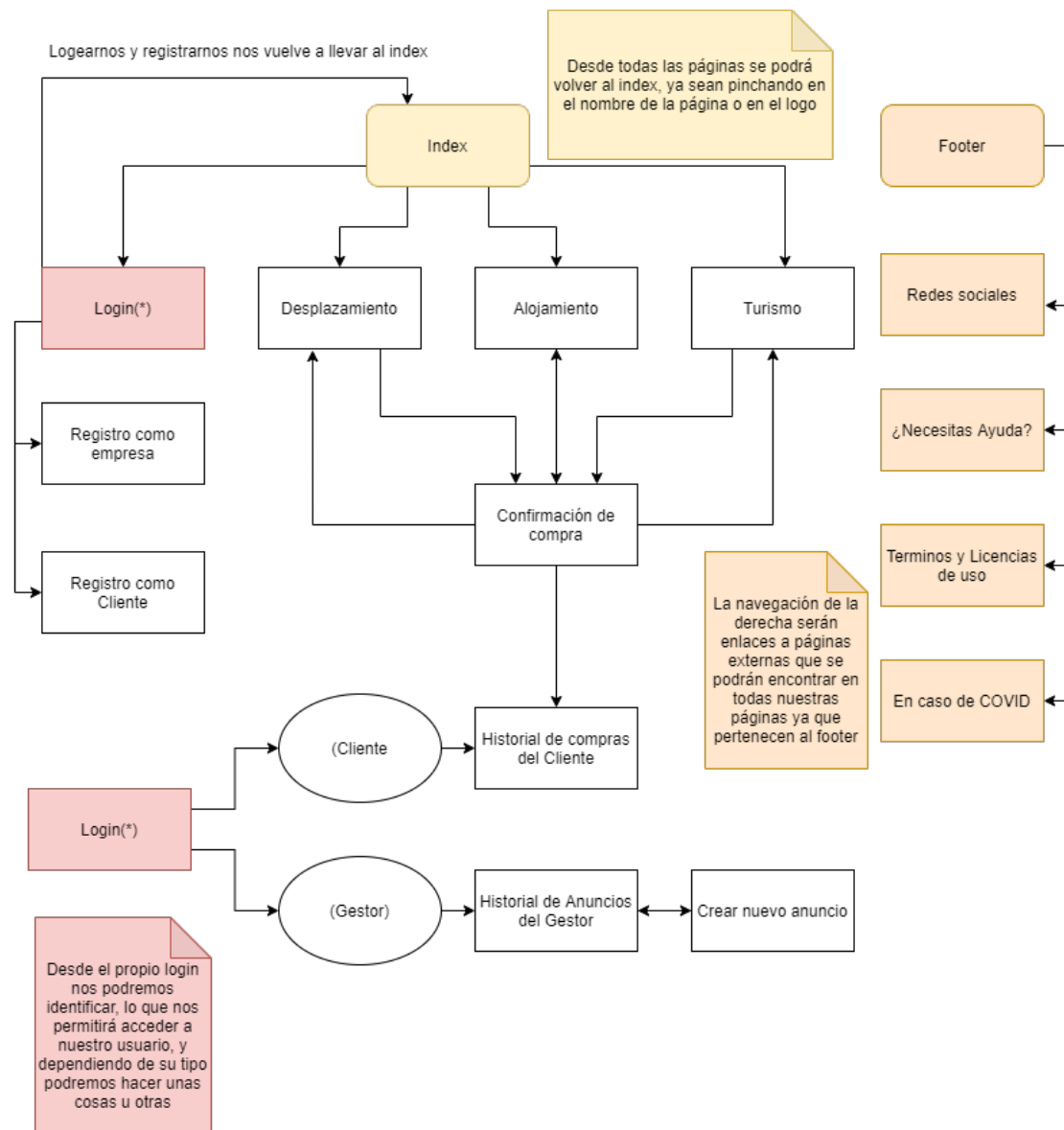


Figura 2: Esquema Relacional de la base de datos

## 6. Mapa del sitio web



## 7. Descripción de la funcionalidad aplicada

Para la realización de la práctica final hemos decidido implementar aquellos casos de uso que permitan asegurar una funcionalidad al 100 por ciento de la página. Dividimos los casos de uso implementados en:

### 7.0.1. Casos de uso implementados del gestor

- Login del gestor: El gestor inicia sesión y su información se almacena en la sesión.
- Registro de un nuevo gestor: El gestor se registra y su información se almacena en la base de datos.
- Publicación de un anuncio: El gestor rellena un formulario con la información de su producto y esta se almacena en la base de datos.
- Acceso al historial de publicaciones: El gestor puede acceder a su área personal clickando en su CIF en el header, donde puede ver toda la información relativa a sus productos publicados en nuestra empresa.

### 7.0.2. Casos de uso implementados del cliente

- Login del cliente: El cliente inicia sesión y su información se almacena en la sesión.
- Registro de un nuevo cliente: El cliente se registra y su información se almacena en la base de datos.
- Realización de una compra: El cliente puede contratar uno de los productos ofertados de las distintas categorías. El sistema le enviará un email con la información de la compra al email vinculado a su cuenta y el usuario confirmará la recepción del email.
- Acceso al historial de compras: El cliente puede acceder a su área personal clickando en su nombre en el header, donde puede ver toda la información relativa a los viajes contratados.

### 7.0.3. Casos de uso implementados para ambos actores

- Logout: el usuario cierra sesión y la información no persistente almacenada en la sesión se elimina.
- Visualización de anuncios disponibles de las tres modalidades.

## 8. Conclusiones

Para la realización de la práctica hemos empleado distintas herramientas software. Primero, programamos las páginas html y css con Visual Studio Code y realizamos los diagramas de modelado con Astah Profesional y Visual Paradigm. Posteriormente, para el desarrollo global de la práctica empleamos Netbeans ya que nos permitía una forma sencilla y efectiva de gestionar nuestro proyecto. Para ello empleamos Maven para gestionar dependencias, Derby (JDBC) para implementar la base de datos y Apache Tomcat para el servidor web. Por último, para la integración del proyecto hemos empleado la herramienta Gitlab.