

HANDITOUR

DOCUMENTACIÓN



Curso 2020/21

alejandro gonzález damín

# 

# ÍNDICE DE CONTENIDO

[ÍNDICE DE CONTENIDO 2](#_Toc74134200)

[1. INTRODUCCIÓN. 3](#_Toc74134201)

[2. DESCRIPCIÓN. 4](#_Toc74134202)

[2.1. INDEX. 4](#_Toc74134205)

[2.2. CIUDAD. 5](#_Toc74134206)

[2.3. DETALLE MONUMENTO. 6](#_Toc74134208)

[2.4. ALOJAMIENTOS. 7](#_Toc74134209)

[2.5. DETALLES DE UN ALOJAMIENTO. 8](#_Toc74134210)

[2.6. APARCAMIENTOS 9](#_Toc74134211)

[2.7. TRANSPORTE. 10](#_Toc74134212)

[2.8. LOG IN Y REGISTRO. 10](#_Toc74134213)

[3. INSTALACIÓN Y PREPARACIÓN. 12](#_Toc74134214)

[4. GUÍAS DE ESTILOS Y PROTOTIPADO. 13](#_Toc74134215)

[ Prototipo index 14](#_Toc74134216)

[ Prototipo ciudad. 14](#_Toc74134217)

[ Prototipo monumento 14](#_Toc74134218)

[ Prototipo alojamientos. 16](#_Toc74134219)

[ Prototipo aparcamientos. 16](#_Toc74134221)

[ Prototipo transportes. 17](#_Toc74134222)

[5. DISEÑO. 18](#_Toc74134224)

[ Diagrama entidad relación. 19](#_Toc74134225)

[6. DESARROLLO 21](#_Toc74134226)

[7. DESPLIEGUE. 22](#_Toc74134227)

[8. MANUAL. 23](#_Toc74134228)

[9. CONCLUSIONES 25](#_Toc74134229)

[10. ÍNDICE DE IMÁGENES. 26](#_Toc74134230)

[11. BIBLIOGRAFÍA 27](#_Toc74134231)

# INTRODUCCIÓN.

La idea inicial que tuvimos para realizar este proyecto fue, hacer una aplicación web para el beneficio del colectivo de discapacitados. Con esta premisa nació la idea de crear un portal para facilitar el hecho de poder viajar. De ahí, nació la idea de handitour, que es un acrónimo de las palabras inglesas handicap (minusválido) y tour (viaje).

Para crear haditour nos hemos basado en ciudades las cuales hemos podido visitar con anterioridad, como son: Cádiz, Sevilla, Valencia, Madrid, Barcelona y Londres. En cada una ellas, se podrá ver los puntos de interés de cada una, como monumentos, alojamientos, transportes…

El proyecto está programado en Django, un framework de Python que usa MVC, además de Django, handitour contiene en casi todas sus pantallas un mapa dinámico programado en Javascript junto con peticiones Ajax y jqury que muestran el contenido dinámico deseado dependiendo de las premisas de cada pantalla.

En cuanto al estilo de la aplicación hemos usado y personalizado el sistema de rejillas y las clases de bootstrap 4.

# DESCRIPCIÓN.

En esta sección del documento, desarrollaremos el resultado del proyecto. Iremos viendo página por página y analizaremos su contenido. Para empezar, vamos a describir el index de la aplicación web.



## INDEX.



#### Ilustración 1. Index

En la ilustración 1, se muestra la disposición de nuestro index. En la parte superior de la pantalla se encuentra el header. Este se compone del título en el centro del mismo y la barra de navegación que muestra el título de nuestra aplicación web, que cuando se hace click redirecciona hacia el index, al lado nos encontramos un ítem del menú que nos llevará a hacer log in o log out según si encuentra o no sesión del usuario.

En la parte central de la ilustración, encontramos una carta por cada ciudad insertada en nuestra base de datos. Al hacer el evento hover sobre la carta apreciamos un cambio en la opacidad de la imagen mostrando el nombre de la ciudad donde se encuentre el puntero del ratón.

Para acabar con la descripción de nuestra pantalla inicial, explicaremos la composición de nuestro footer. Su composición es simple, está compuesto por varios iconos que apuntan a las redes sociales del contenido de nuestra aplicación junto con el nombre del autor del portal.

## CIUDAD.

## 

#### Ilustración 2. Ciudad.

En la ilustración 2, podemos apreciar la vista de una ciudad. En el header nos encontramos prácticamente con el mismo header que en la ilustración 1, salvo que esta vez la barra de navegación tiene diferentes ítems con las diferentes secciones que más adelante explicaremos detenidamente.

En el cuerpo de esta página tenemos dos grandes partes muy bien diferenciadas:

1. En la parte izquierda de la imagen podemos apreciar un listado con los monumentos que podemos visitar en la ciudad seleccionada. Este listado está compuesto por un tipo de carta con más elementos que las cartas desarrolladas en el index. Estas se componen de título, que muestra el nombre del monumento; subtítulo, en el que aparece una breve descripción de la historia del monumento, y un botón de más información que redirecciona a la pantalla de detalle, que ampliaremos en posteriores pantallas.
2. En la parte de la derecha de la imagen nos encontramos con un mapa que sitúa todos los monumentos del listado comentado anteriormente. El mapa está programado en javascript usando la api de google maps, concretamente la de geolocalización. Para mostrar todos los monumentos insertados en la base de datos, también hemos tenido que usar jquery junto con Ajax, con todo esto conseguimos enviarle al mapa la posición exacta de cada monumento.

Para acabar con la explicación de esta pantalla también cabe destacar la parte del footer que es igual que hemos expuesto anteriormente en la ilustración 1, correspondiente al index.

## DETALLE MONUMENTO.

#### 

#### Ilustración 3. Detalle monumento.

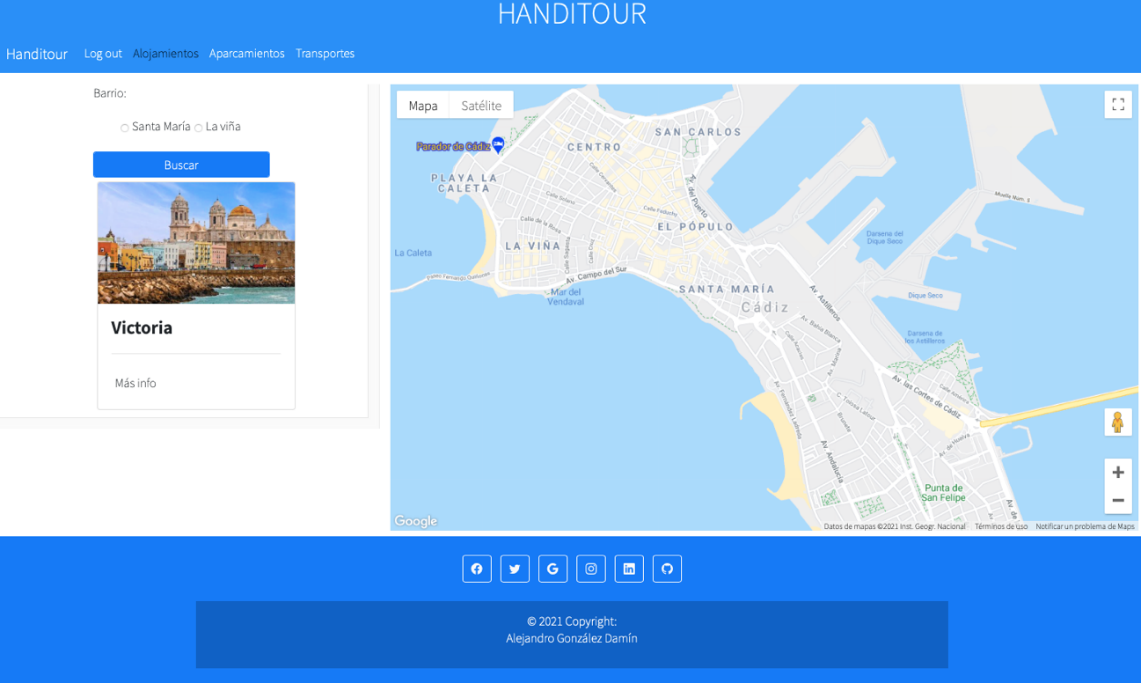
Para empezar a comentar esta ilustración, cabe destacar que aunque no se pueda apreciar en la misma tiene header y footer, pero por las dimensiones de la página no se han podido ilustrar correctamente. La composición de los mismos ha sido desarrollada exactamente igual que en la ilustración 2. Dicho esto, pasaremos a analizar la parte central de nuestra página de detalle.

En la parte justo debajo del header tenemos la información relativa al monumento seleccionado en la pantalla anterior. Con tres partes muy diferenciadas:

1. En la parte más a la izquierda de la pantalla, nos encontramos una tabla en la que se describe toda la información relativa al monumento seleccionado.
2. En la parte central de la pantalla, tenemos la imagen característica del monumento.
3. En la parte derecha de la pantalla, nos encontramos con un mapa, el cual está programado igual que los anteriores salvo con una excepción, no le hemos querido meter ningún filtro para que el usuario tenga referencias sobre lo que hay alrededor del monumento, por ejemplo restaurantes, souvenirs, transportes, etc.

En esta pantalla si el usuario está logueado le aparecerá un formulario donde podrá opinar sobre el monumento y sus alrededores. Por el contrario si el usuario no está logueado le aparecerá un párrafo con una redirección hacia la pantalla de log in. Tanto como si ha hecho log in como si no le aparecerán los comentarios hechos por otros usuarios.

## ALOJAMIENTOS.



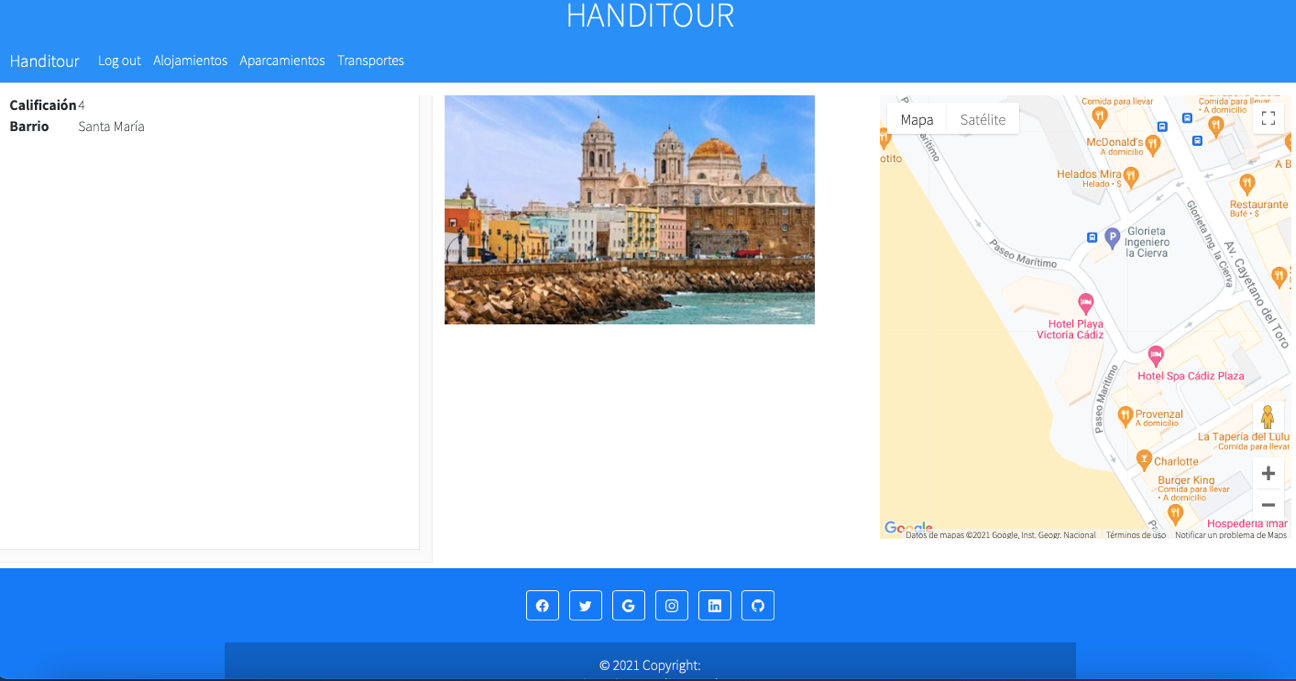
#### Ilustración 4. Alojamientos.

En este apartado tenemos los alojamientos de una ciudad. Al igual que en los anteriores apartados footer y header son iguales.

En esta pantalla podemos diferenciar dos grandes partes:

1. En la izquierda tenemos un buscador y todos los alojamientos de una ciudad. Si seleccionamos un barrio nos muestra todos los alojamientos de ese barrio.
2. En la parte derecha se encuentra el mapa programado al igual que los anteriores, con una diferencia, si el usuario desea hacer uso del buscador, aparecerá en el mapa el barrio deseado con sus alojamientos correspondientes. También el mapa tiene la opción de filtrado activada para que nada más se muestren alojamientos en esta pantalla.

## DETALLES DE UN ALOJAMIENTO.

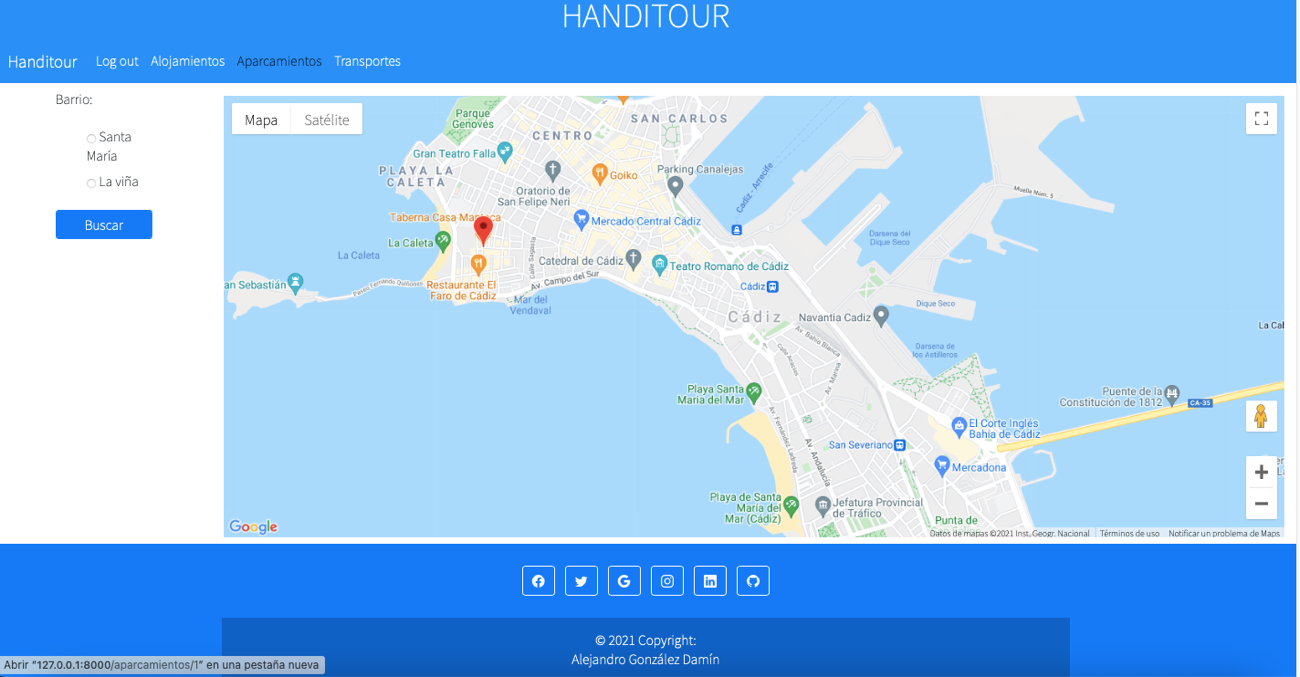


#### Ilustración 5. Detalles de un alojamiento.

En esta pantalla podemos visitar las características de un alojamiento en concreto. En cuanto a la estructura de la misma header y footer son comunes a todas las páginas, por lo tanto no tenemos mucho más que decir ya que lo hemos mostrado en anteriores punto de este apartado.

En el centro de la pantalla como es una vista de detalle, lo hemos hecho coincidir con la vista de detalle de un monumento. Para repasar dicha estructura el centro de la pantalla lo hemos dividido en tres columnas, en la primera de ellas tenemos los datos relativos al alojamiento, seguidamente se muestra la imagen del mismo y en la última su localización exacta.

## APARCAMIENTOS



#### Ilustración 6. Aparcamientos.

En esta pantalla header y footer son exactamente iguales que en las anteriores. El objetivo de esta pantalla es que el usuario tenga a mano los aparcamientos de minusválidos que hay en la ciudad. Para hacer más fácil al usuario encontrar aparcamiento le hemos programado al igual que en puntos anteriores un filtro de barrios que centrará el mismo en el mapa.

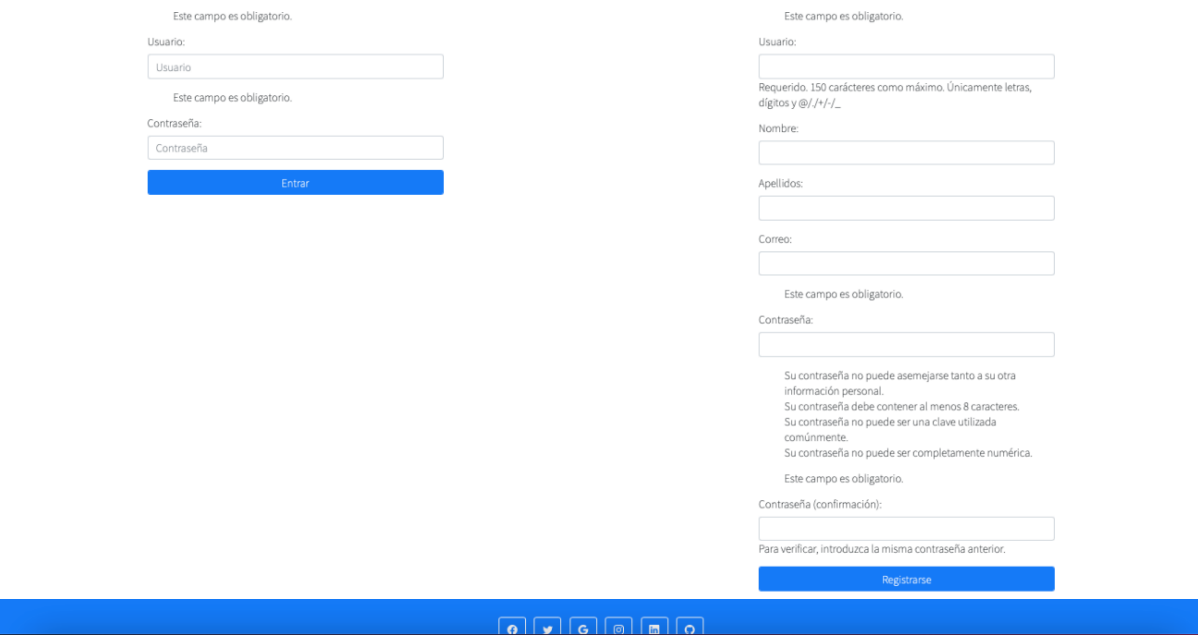
## TRANSPORTE.

#### 

#### Ilustración 7. Transporte.

En la siguiente pantalla al igual que en las anteriores header y footer son iguales que en los apartados anteriores. En esta pantalla el usuario puede consultar las líneas de autobuses de la ciudad y su localización mediante un mapa. También podrá filtrar las paradas por lo barrios y consultar que líneas que pasan por cada parada.

## LOG IN Y REGISTRO.



#### Ilustración 8. Log in y registro.

Con esta ilustración llegamos al final de esta sección, pero no menos importante. Para que el usuario pueda hacer algunas interacciones con nuestro portal, primero deberá haberse registrado en el mismo y haber iniciado sesión. En cuanto al desarrollo de esta pantalla, django nos proporciona un usuario predeterminado y los formularios propios para que esta tarea sea más llevadera. A parte, hace el control de inicio de sección y final de sección mediante sus respectivas vistas.

# INSTALACIÓN Y PREPARACIÓN.

Para empezar a trabajar en este proyecto primero hemos tenido que verificar que teníamos Python instalado en nuestro ordenador. Una vez verificado esto mediante un pip instalamos el framework django. Esto nos dejará un archivo muy importante que usaremos a lo largo de todo nuestro desarrollo que se llama manage.py. con el uso del comando Python manage.py startapp junto con el nombre de nuestra aplicación. Esto último creará la carpeta con todos los ficheros necesarios para ejecutar la aplicación.

Seguidamente haciendo uso de nuevo del manage.py “fabricamos las migraciones”, con el comando Python manage.py makemigrations. Una vez que tenemos las migraciones preparadas usamos el comando Python manage.py migrate, el conjunto de estos comandos inicializará la base de datos de nuestro proyecto. Cuando esté creada la app habrá que instalarla en el proyecto. Para hacer esto tenemos que abrir el módulo settings.py, buscar la línea dónde tenemos la lista de apps instalada y añadimos un nuevo ítem compuesto de comilla simple, el nombre de la app en minúscula, seguido de punto app, seguido de otro punto con el nombre de nuestra aplicación con la primera letra en mayúscula junto con Config.

En cuanto a la parte sobre control de versiones, nos hemos creado una cuenta de github con el dominio de correo electrónico proporcionado por el instituto. Seguidamente en la misma cuenta nos hemos creado un repositorio, que lo hemos puesto en común con el resto de alumnos que presentan el proyecto en esta convocatoria. Después de la puesta en común de nuestro proyecto, en la carpeta local del mismo hemos usado el comando git init, con el que le decimos a nuestra carpeta local que esa carpeta va a ser una carpeta remota. Una vez hecho esto ya sólo queda clonar el repositorio que hemos creado en nuestra carpeta local y usamos los comandos git commit y git push, que sería la subida inicial de nuestro código creado en los pasos anteriores. En cuanto a la frecuencia que hemos ido actualizando el repositorio ha sido un commit push cada vez que acabábamos una sesión de desarrollo.

# GUÍAS DE ESTILOS Y PROTOTIPADO.

En esta sección vamos a comentar las guías de estilos de nuestra aplicación y el desarrollo de los prototipos.

En cuanto a las guías de estilos, hemos modificado la guía que nos proporciona boostrap 4. Esta herramienta nos proporciona una guía de estilos híbridas juntando dos técnicas avanzadas para la colocación y capacidad de ajustar nuestro portal a diferentes pantallas, como son grid y flexbox. El grid nos divide los contenedores como máximo en 12 columnas, así nos permite colocar los elementos de nuestra página de forma fácil y directa. También a la hora de cambiar la resolución de la pantalla, nada más que tenemos que decirle cuantas columnas queremos dependiendo del tamaño de la pantalla. Junto a grid boostrap 4 hace uso de flexbox, esta herramienta les da flexibilidad a los contenedores para hacer más fácil la disposición de columnas que nos proporciona el grid.

A continuación, les mostramos los mockups que diseñamos antes de empezar a desarrollar en código nuestro proyecto. Debido a la repetición de las pantallas expuestas en el punto de descripción de este documento, hemos diseñado sólo y exclusivamente aquellas que no se repiten.

### Prototipo index

#### 

#### Ilustración 9. Prototipo index

### Prototipo ciudad.

#### 

#### Ilustración 10. Prototipo ciudad.

### Prototipo monumento



#### Ilustración 11. Prototipo monumento.

### Prototipo alojamientos.

## 

#### Ilustración . Prototipo alojamientos.

### Prototipo aparcamientos.



#### Ilustración . Prototipo apsrcamientos.

### Prototipo transportes.

## 

#### Ilustración . Prototipo transportes.

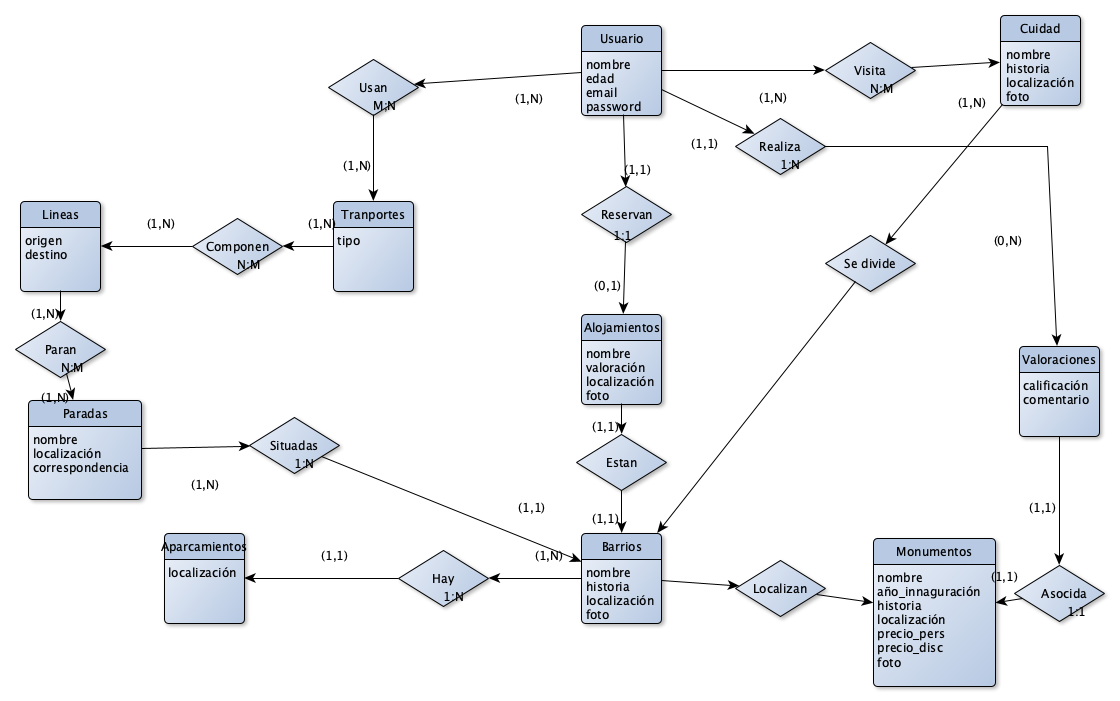
# DISEÑO.

En esta sección describiremos los casos de uso y el diagrama entidad relación de nuestro proyecto. El portal tiene principalmente 12 casos de uso, los cuales son los siguientes:

1. **Registro:** en este caso de uso el usuario podrá registrarse en nuestro portal.
2. **Log in:** el usuario una vez que haya hecho el registro podrá entrar como usuario en nuestro portal.
3. **Index**: este caso de uso muestra todas las ciudades insertadas en la base de datos.
4. **Ciudad:** este caso de uso lista todos los monumentos de una ciudad insertados en la base de datos.
5. **Mapa:** señala la situación geográfica en las distintas pantallas de nuestro portal.
6. **Detalle de monumento:** se muestra la información completa de un monumento.
7. **Comentario a un** monumento: el usuario puede hacer comentarios sobre un monumento.
8. **Alojamientos:** lista los alojamientos de una ciudad.
9. **Filtro de barrio:** filtra el mapa por barrios.
10. **Aparcamientos:** muestra la situación geográfica de los aparcamientos de una ciudad.
11. **Transportes:** lista las líneas de autobuses de una ciudad.
12. **Log out**: destruye la sección de un usuario.

A continuación, mostraremos el diagrama entidad relación de la base de datos de nuestro proyecto.

### Diagrama entidad relación.



#### Ilustración . Diagrama entidad relación.

En la ilustración podemos ver el diagrama entidad relación de nuestro proyecto, este se compone de 10 entidades las cuales describiremos junto con sus relaciones a continuación:

1. **Usuario**: sus atributos son nombre, edad, email y contraseña.
2. **Ciudad:** compuesta por nombre, historia, localización y foto. Junto a usuario forma una relación de muchos a muchos.
3. **Valoraciones:** compuesta por calificación y comentarios. Que junto a usuario forma una relación de muchos a muchos.
4. **Monumentos:** se compone de nombre, año de inauguración, historia, localización, precio persona, precio discapacitado y foto. Junto a valoraciones forma una relación de uno a muchos.
5. **Barrios:** se compone de nombre, historia, localización y foto. Junto a monumentos forman una relación de uno a muchos.
6. **Aparcamientos:** compuesto de localización. Junto a barrios forma una relación de uno a muchos.
7. **Alojamientos:** compuesto por nombre, valoración, localización y foto. Junto a barrio forma una relación de uno a muchos.
8. **Paradas:** que consta de nombre, localización y correspondencia. Junto a barrios forma una relación de uno a muchos.
9. **Líneas:** sus atributos son origen y destino. Junto a paradas forman una relación de muchos a muchos.
10. **Transportes:** su único atributo es tipo. Junto a líneas forman una relación de muchos a muchos.

# DESARROLLO

En cuanto a la secuencia de desarrollo que hemos usado en este proyecto, hemos ido encadenando una página con otra, es decir, si hemos desarrollado una lista a continuación desarrollamos los detalles de cada ítem de la lista. De este modo hemos obtenido el resultado final.

En cuanto a los conflictos que nos han surgido durante el desarrollo del proyecto, ha habido varios puntos a comentar en este apartado. En primer lugar, surgieron conflictos a la hora de desarrollar el mapa, ya que en el javascript no podía meter las variables de django si poníamos el script independiente del html, dado que estas nada más que funcionan en el html. Probamos una aplicación del propio django para imprimir los mapas llamada django-easy-map, esta aplicación funcionaba pero a la hora de tener un mapa con dos ubicaciones cuando recorría la consulta imprimía una mapa por cada iteración de la consulta. La solución parcial a este problema fue meter el código javascript en nuestro html, pero claro desde el punto de vista como programador no era una buena práctica. Finalmente, se nos ocurrió crear funciones javascript en archivos independientes con los argumentos necesarios y desde el html llamábamos a dichas funciones pasándole la variable django como argumento.

Otro conflicto que nos ha parado el desarrollo unos cuantos días fue crear el filtro de barrios. Lo hicimos con un formulario django al cuál le teníamos que pasar la id de la ciudad para que nos devolviera los barrios de la misma. La complicación estuvo en que no encontrábamos la forma de pasar de una ciudad a la lista de barrios, pero tuvo fácil solución django proporciona un atributo llamado set para los casos de relaciones de uno a muchos. Mediante este atributo se solucionó nuestro problema.

# DESPLIEGUE.

En esta sección vamos a explicar como hemos desarrollado el despliegue de nuestra aplicación. Esto lo hemos hecho mediante docker. El despliegue se produce en cuatro sencillos pasos:

1. Clonamos nuestro repositorio del proyecto en una máquina virtual creada por docker.
2. El paso anterior nos creará un directorio con los archivos clonados desde git.
3. Ponemos en marcha el contenedor de docker, con el comando docker-compose up, lo que nos dará el archivo compose al cual le meteremos la url principal de nuestro proyecto.
4. Finalmente entramos en la url indicada en el paso anterior y accederá a nuestro portal.

# MANUAL.

Llegados a este punto que ya hemos expuesto el diseño y desarrollo de nuestro proyecto, ahora expondremos como se debería de usar nuestra aplicación en su totalidad a nivel de usuario.

Cuando el usuario entra en el index de nuestro portal, se encuentra con dos acciones a elegir:

1. En la barra de navegación puede hacer clic en log in, lo que le redirigirá a la pantalla de autenticación que a su vez tiene dos acciones posibles:
   1. Si es la primera vez que accede a nuestro portal podrá rellenar el formulario de registro.
   2. Si ya tiene cuenta simplemente tiene que autenticarse con su usuario y contraseña.
2. Puede seleccionar una de las ciudades para obtener más información sobre ella, además si no conoce las ciudades al pasar el ratón por cada una de ellas podrá ver sus respectivos nombres.

Cuando el usuario pincha en una ciudad es redirigido al listado de monumentos de la misma, junto a éste aparecerá un mapa con la localización de cada monumento. En cuanto a las interacciones del usuario tiene diversas opciones.

En la barra de navegación tiene cuatro opciones:

1. Ir a la pantalla de autenticación explicado anteriormente.
2. Ir a la pantalla de alojamientos de esta ciudad.
3. Ir a la pantalla de aparcamientos de esta ciudad.
4. Ir a la pantalla de transportes de esta ciudad.

Por último, el usuario puede hacer clic en más info del monumento para encontrar todas las características de ese monumento.

En la pantalla de alojamientos, en cuanto a la barra de navegación el usuario opta a las cuatro interacciones expuestas anteriormente. En esta pantalla el usuario en una primera instancia puede ver todos los alojamientos de una ciudad. Haciendo uso del buscador compuesto por el listado de barrio podrá filtrar tanto el mapa como el listado según el barrio indicado, al hacer clic en buscar se aplicará el filtro.

En la pantalla posterior, además de las cuatro interacciones de la barra de navegación, el usuario podrá ver los aparcamientos de una ciudad agrupados en un mapa. Como en la pantalla anterior el usuario podrá filtrar dicho mapa por los barrios de una ciudad y ver los aparcamientos de dicho barrio.

En la última pantalla de nuestro portal la interacción del usuario es exactamente igual que en la pantalla de alojamientos, con la particularidad de que se mostrarán las paradas de autobuses de la ciudad.

# CONCLUSIONES

Después de que hayamos expuesto la elaboración y resultado final de nuestro proyecto, cabe destacar que es muy semejante a la idea inicial. Se ha elaborado en diez meses. En principio era cosa de fin de semanas, al pasar un par de meses viendo lo desarrollado hasta ese momento decidimos emplear más tiempo compaginándolo con el trabajo. En cuanto a mi experiencia personal, acorde con mi situación me ha sido más complicado y me ha supuesto mucho más esfuerzo en el desarrollo del mismo. Finalmente he quedado muy satisfecho con el resultado final.

Espero que el proyecto haya sido de vuestro agrado, he intentado ser original y creo haberlo conseguido en mi modesta opinión. Espero que os haya gustado la idea y el desarrollo del mismo.

En cuanto a las mejoras que se le pueden incluir a este proyecto, sería retroalimentar la base de datos desde la api de gougle maps.

Os agradezco vuestra atención y en especial a Don Javier Ortega por su paciencia y por siempre estar cuando lo he necesitado, porque gracias a su ayuda el proyecto ha salido adelante.

# ÍNDICE DE IMÁGENES.

[Ilustración 1. Index 4](#_Toc74003738)

[Ilustración 2. Ciudad. 5](#_Toc74003739)

[Ilustración 3. Detalle monumento. 6](#_Toc74003740)

[Ilustración 4. Alojamientos. 7](#_Toc74003741)

[Ilustración 5. Detalles de un alojamiento. 8](#_Toc74003742)

[Ilustración 6. Aparcamientos. 9](#_Toc74003743)

[Ilustración 7. Transporte. 10](#_Toc74003744)

[Ilustración 8. Log in y registro. 10](#_Toc74003745)

[Ilustración 9. Prototipo index 13](#_Toc74003746)

[Ilustración 10. Prototipo ciudad. 14](#_Toc74003747)

[Ilustración 11. Prototipo monumento. 14](#_Toc74003748)

[Ilustración 12. Prototipo alojamientos. 15](#_Toc74003749)

[Ilustración 13. Prototipo apsrcamientos. 15](#_Toc74003750)

[Ilustración 14. Prototipo transportes. 16](#_Toc74003751)

[Ilustración 15. Diagrama entidad relación. 18](#_Toc74003752)

# BIBLIOGRAFÍA

* <https://www.booking.com/searchresults.es.html?label=gen173nr-1FCAEoggI46AdIM1gEaEaIAQGYAQq4ARjIAQ3YAQHoAQH4AQuIAgGoAgS4AvyogoYGwAIB0gIkNmZkYTMzN2MtOGYwMC00NzQzLTgzNjItYmEwYzcxYWI0YzUz2AIG4AIB&sid=fa73a58635fc6573bd043057fb1e2476&sb=1&sb_lp=1&src=index&src_elem=sb&error_url=https%3A%2F%2Fwww.booking.com%2Findex.es.html%3Flabel%3Dgen173nr-1FCAEoggI46AdIM1gEaEaIAQGYAQq4ARjIAQ3YAQHoAQH4AQuIAgGoAgS4AvyogoYGwAIB0gIkNmZkYTMzN2MtOGYwMC00NzQzLTgzNjItYmEwYzcxYWI0YzUz2AIG4AIB%3Bsid%3Dfa73a58635fc6573bd043057fb1e2476%3Bsb_price_type%3Dtotal%3Bsig%3Dv1c46jFc67%3B&ss=Cádiz%2C+Andaluc%C3%ADa%2C+España&is_ski_area=&checkin_year=&checkin_month=&checkout_year=&checkout_month=&group_adults=2&group_children=0&no_rooms=1&b_h4u_keep_filters=&from_sf=1&ss_raw=cad&ac_position=1&ac_langcode=es&ac_click_type=b&dest_id=-374884&dest_type=city&place_id_lat=36.529514&place_id_lon=-6.295139&search_pageview_id=887747fee3aa01fa&search_selected=true&search_pageview_id=887747fee3aa01fa&ac_suggestion_list_length=5&ac_suggestion_theme_list_length=0#map_opened>
* <https://docs.djangoproject.com/en/3.2/>
* <https://stackoverflow.com>
* <https://github.com>
* <https://getbootstrap.com>
* <https://developers.google.com/maps/documentation/javascript/overview>
* <https://jquery.com>