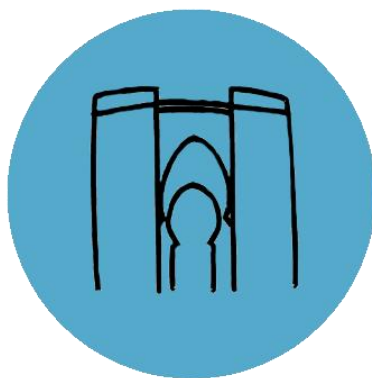


IES JUAN BOSCO

TURISMO EN CIUDAD REAL



SANDRA BLÁZQUEZ NOTARIO

2º DESARROLLO DE APLICACIONES WEB

CURSO 2021/2022

Introducción

La idea del proyecto surge, por un lado, de mis intereses profesionales al haber estudiado previamente Historia del Arte, y por otro, de la inconformidad personal con la página actual de turismo, debido a que presenta un diseño algo desactualizado.

Por otro lado, me pareció buena idea crear dos perfiles más a parte del usuario anónimo que realice las consultas de información: el usuario registrado, que en un diseño final sería capaz de guardar sus favoritos y comentar en visitas guiadas para ayudar a los usuarios a decidir si acudir. Otra función más adelante sería la de poder sugerir al usuario rutas según su tiempo, preferencias y otros parámetros.

Por otro lado, el usuario administrador (Oficina de Turismo) será capaz de manejar desde esta, a modo de backoffice, la información de interés que comparta. En definitiva, lo que se busca, como objetivo general, es simplificar la consulta y gestión del turismo.

Este proyecto formaba parte inicialmente de un proyecto mayor encaminado a ayudar a los Ayuntamientos y sus ciudadanos a comunicarse de manera simple.

Guía de estilo

Como fuente para la página se ha escogido Montserrat, una fuente sans-serif y redondeada, que transmite una sensación de cercanía y modernidad.

Por otro lado, para escoger los colores se ha pensado desde el punto de vista de que es parte de una web mayor. Así, los colores escogidos son de las gamas azul y amarillo. El azul transmite tranquilidad, confianza, seriedad, mientras que el amarillo es su complementario y refleja actividad, energía, usado de manera puntual y sin excesos.

Los tonos principales en la aplicación son:

```
$primaryColor: #55aacc;  
$secondaryColor: #F2BA52;  
$primaryColorDark: #192340;  
$primaryColorLight: #6ad3fd;
```

Del color principal derivan otros secundarios, en tono más oscuro y mas claro, que serán empleados en diferentes elementos:

- El color principal estará presente, en esta o su variante más clara, en todos los fondos a pantalla completa, botones principales y el header.
- El color primario oscuro se empleará principalmente en la parte de administrador, ya que, por acercarse al negro, transmite mayor seriedad. También aparecen en la parte de usuario, pero en menor medida, siendo en botones que necesiten destacar menos. Finalmente es el color principal del footer.

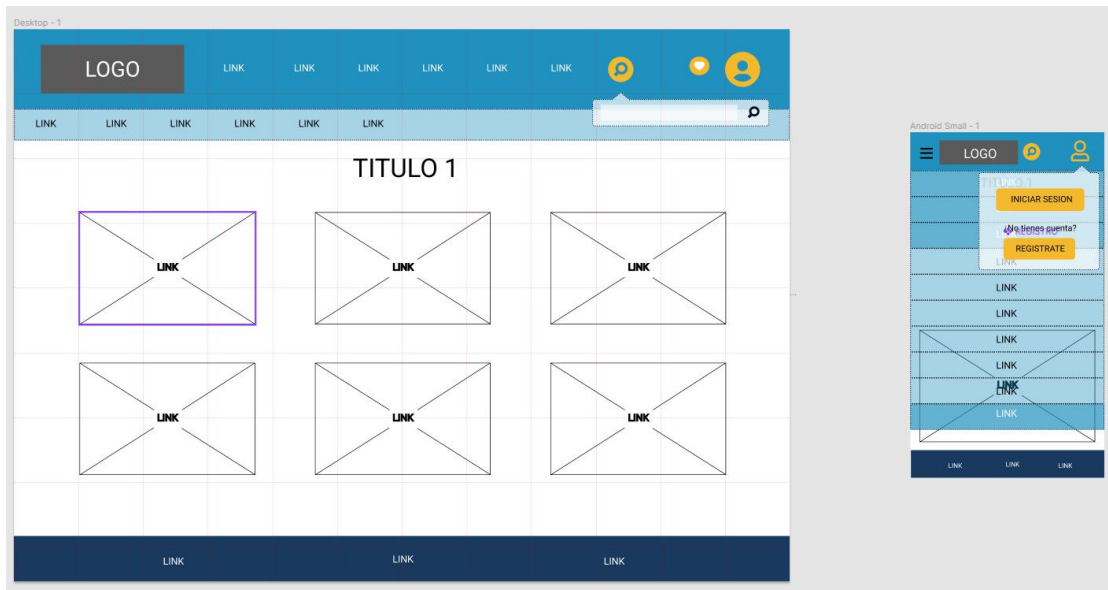
El color secundario será empleado, principalmente en botones secundarios así como en las pantallas de información de los sitios, buscando llamar la atención.

En cuanto al logotipo escogido, se ha buscado entre los elementos patrimoniales de Ciudad Real cuál podría ser el más representativo y sin duda se trata de la Puerta de Toledo. De manera que el logo, provisionalmente, es un diseño simplificado de esta:

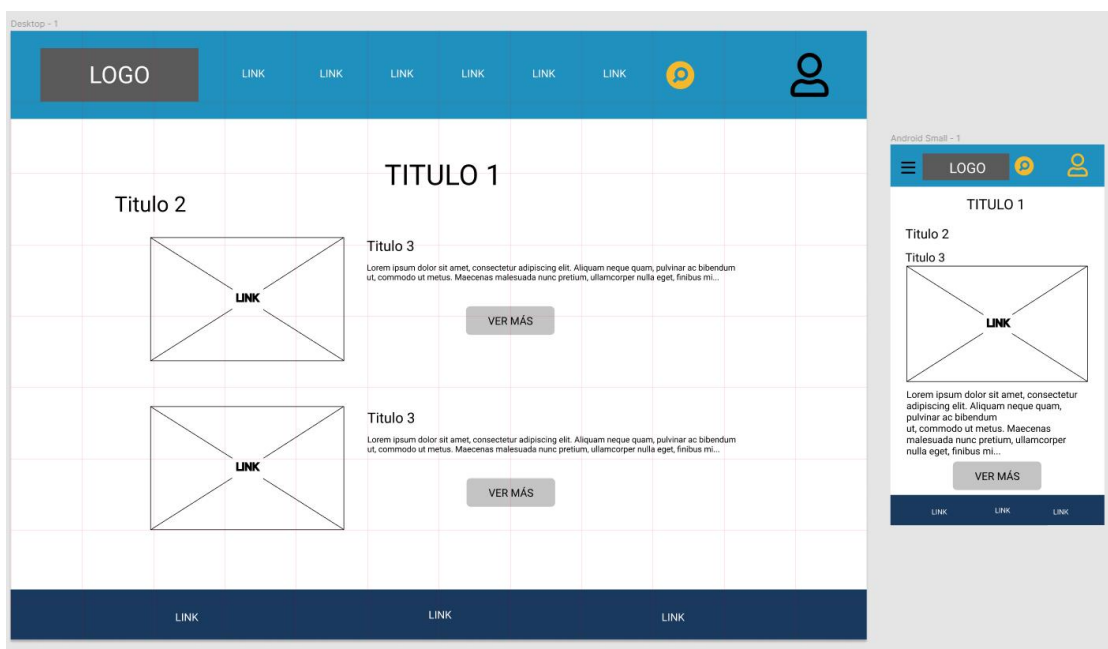


Mockups

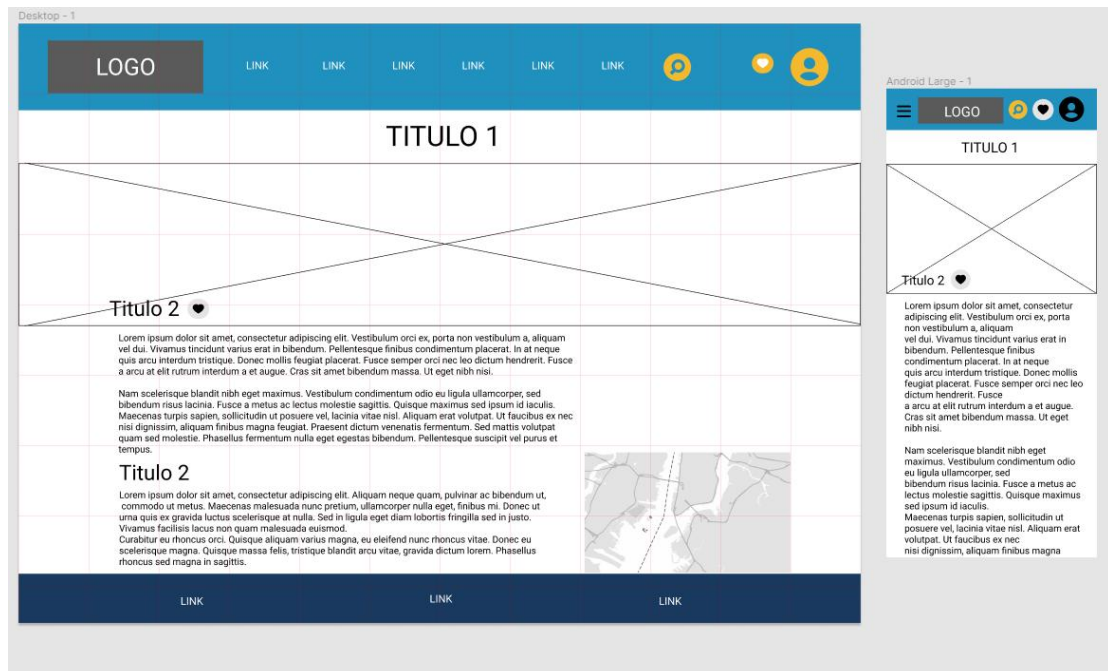
Menú principal. Como podemos ver, por lo general, el menú principal se ha mantenido, a excepción de la barra de navegación. Por un lado, debido a que finalmente se agregaron buscadores en las principales páginas, no se ha incluido el buscado. Por otro lado, para ser un diseño más amigable, los submenús pasaron de aparecer *on hover* a ser fijos.



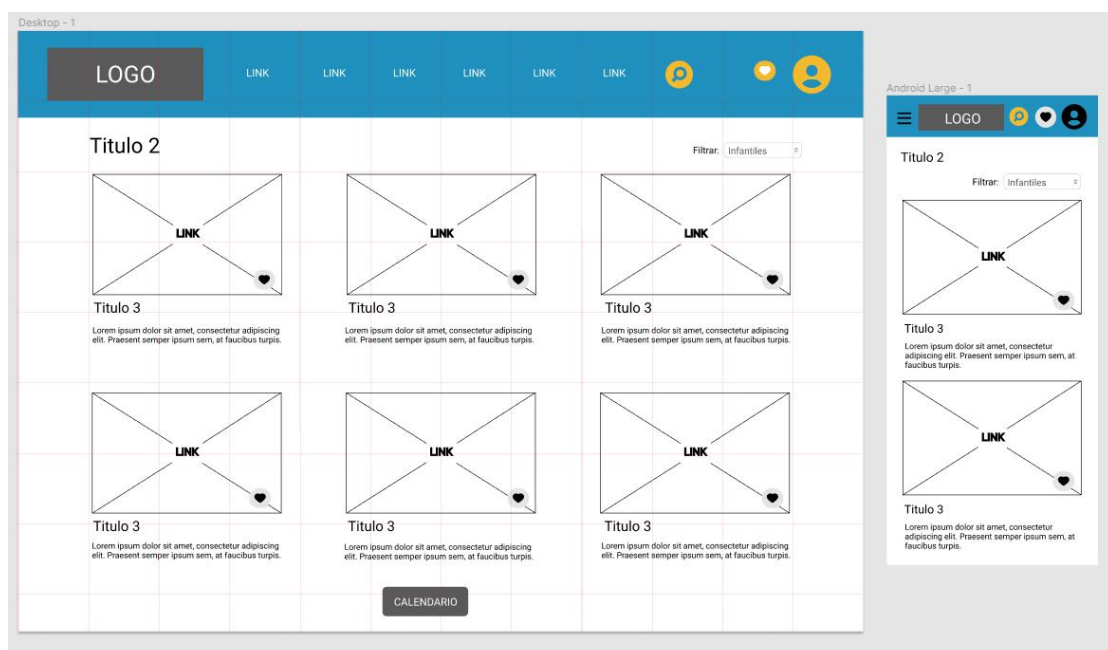
Listados de búsqueda. En la página de búsqueda he mantenido el diseño, a excepción de un contenedor algo más elaborado y que diferencia claramente donde acaba el elemento.



Patrimonio. La página de patrimonio ha sufrido algunos cambios, principalmente se ha simplificado el texto y la imagen de gran tamaño ha sido sustituida por un fondo que contenga la información y la imagen principal.



Ocio. La página de ocio tan solo ha sufrido una pequeña modificación al adaptar el diseño a cards de Bootstrap, ya que estas presentan el contenido de manera ordenada.

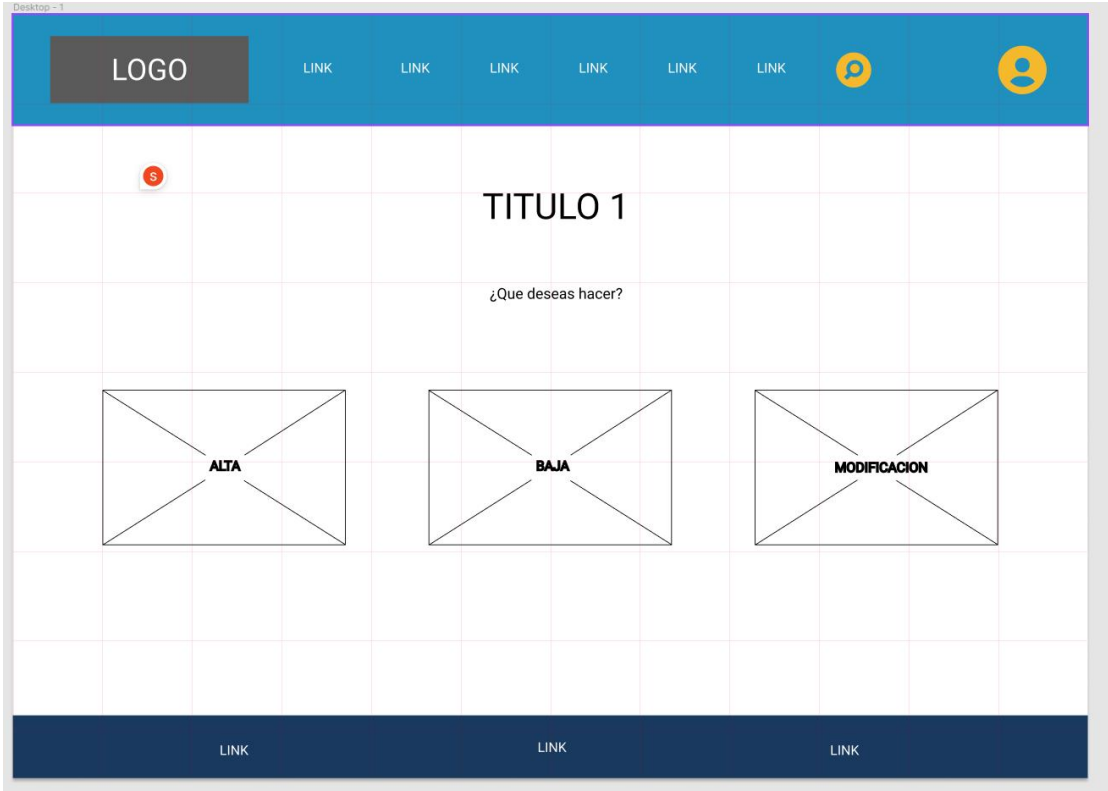


Historia. La página de historia no ha sufrido apenas cambios, a excepcion de no incluir un mapa con la localización.

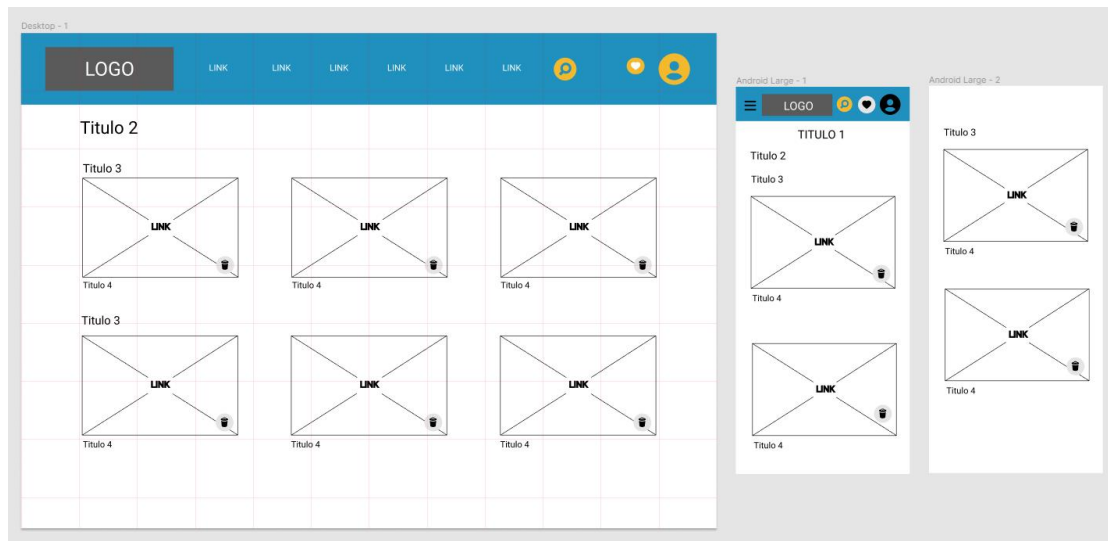


Páginas privadas

Index del administrador. El index del administrador ha sufrido bastantes cambios debido a que la organización de las páginas de altas, bajas y modificaciones se ha planteado de otra manera más intuitiva y amigable.

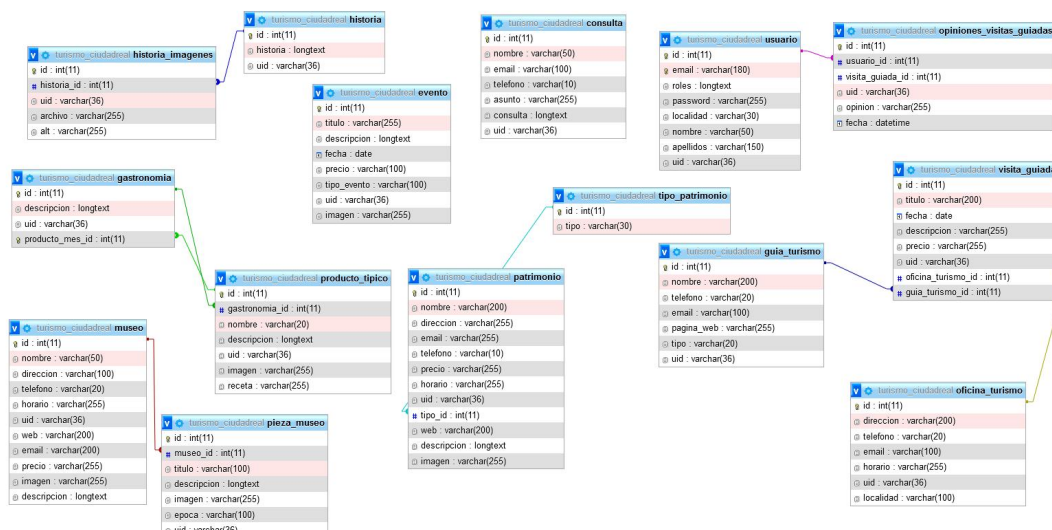


Index del usuario. El perfil del usuario es, quizá, la que mas modificaciones ha sufrido debido a que se ha dejado de lado la función de favoritos en favor de otras.



Base de datos

La base de datos de este proyecto es bastante grande debido a la gran cantidad de información que se muestra y se permite modificar. En un futuro, si se siguiera con la aplicación conjunta para los ayuntamientos, todas estas tablas tendrían un nexo de unión que sería una tabla con la localidad, algo que ha ido ya incluyendo en algunas tablas como campo.



Implementación

Para el desarrollo de este trabajo se ha escogido Symfony, debido a un mayor conocimiento en el tratamiento de los formularios y la gran cantidad de entidades que suponía su desarrollo. Junto con Symfony, toda la parte de front se ha desarrollado con Twig, mientras que para la parte de estilos se ha empleado Sass, Bootstrap y JQuery para hacer la página más amigable al usuario y evitar ciertas recargas innecesarias. Finalmente para el despliegue se ha escogido Heroku debido a que está muy bien documentado el despliegue de una aplicación de Symfony y posee además un servicio gratuito de base de datos MySQL online como es ClearDB MySQL.

Como entorno de desarrollo he usado tanto Netbeans como Visual Studio Code. Esto debido a que la parte visual la trabajo mejor desde Visual Studio Code, pero mientras desarrollaba los controladores y los templates, este me marcaba algunos errores, no autocompletaba...seguramente debido a que me falta algun plugin o realicé mal alguna configuración. A esto se sumaba la mayor familiaridad con Netbeans usando Symfony.

Entre algunos de los problemas encontrados cabe destacar que se ha quedado fuera del desarrollo la implementación de favoritos debido al desconocimiento de herencia con Doctrine (el método escogido fue que se generara una tabla llamada Elemento con los atributos compartidos y otra para cada hijo). Para evitar columnas innecesarias en la base de datos, planteé crear una clase padre conjunta para patrimonio y museos, también debido a la similitud de estas entidades. Sin embargo, a pesar de seguir las instrucciones de la documentación de Doctrine las tablas no se generaban como era esperado (generaba solo las tablas, Museo y Patrimonio, con todos los atributos). Además, como fue una decisión tardía, conforme me informaba de cómo se trabajaba vi que supondría grandes cambios.

Por ello, y para no frenar el proceso de desarrollo decidí optar por otra de las funcionalidades del usuario, muy parecida, pero donde solo se relacionaban dos entidades: el comentar una visita guiada. Esta decisión se toma también desde el punto de vista de utilidad, ya que esta funcionalidad ayuda al empresario local.

Otro problema encontrado fue que, mientras incluía llamadas a métodos de los controladores con Ajax, en algunas ocasiones se quedaba la petición bloqueada en











los métodos propios de Doctrine, (como `findAll()`), de forma que generaba un timeout. No encontré por qué pasaba y simplemente con el tiempo volvía a funcionar correctamente.

Finalmente otro de los problemas más importantes fue durante el despliegue, ya que Heroku devolvía errores relacionados con Dotenv. Finalmente opté por desplegar desde GitHub y saltó otro error relacionado con el `Composer-lock.json`, quizá debido a las pruebas que se habían ido haciendo antes por el error de Dotenv. Este error si pude solucionarlo actualizando todo lo relacionado con Composer. El último error fue que no detectaba bien el Procfile al desplegar, con lo cual no detectaba que la carpeta raíz era `‘/public’`, así que se regeneró varias veces hasta que lo pudo leer correctamente.

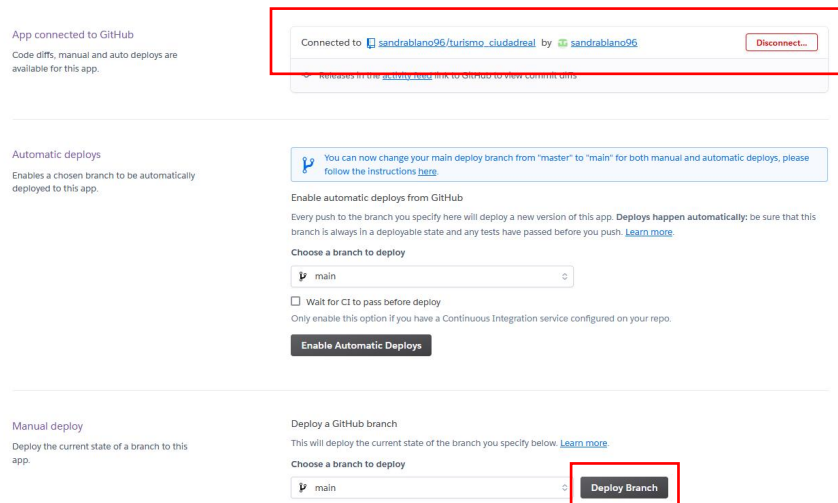
Despliegue

Para el despliegue se han seguido los siguientes pasos:

Primeramente, se han ejecutado diferentes comandos para preparar el proyecto, como indica la [documentacion de Symfony](#), relacionados principalmente con comprobar que la aplicación cumple los requisitos, con la creación de un nuevo `.env` o la actualización de Composer a producción. Posteriormente, siguiendo este [tutorial de Symfony](#), se ha accedido a la carpeta `‘/public/build’` y se ha ejecutado el comando `npm run build` para que genere los ficheros necesarios. También quitaremos del `.gitignore` la carpeta `‘/public/build’` para que se suba y Heroku encuentre el `index.php`.

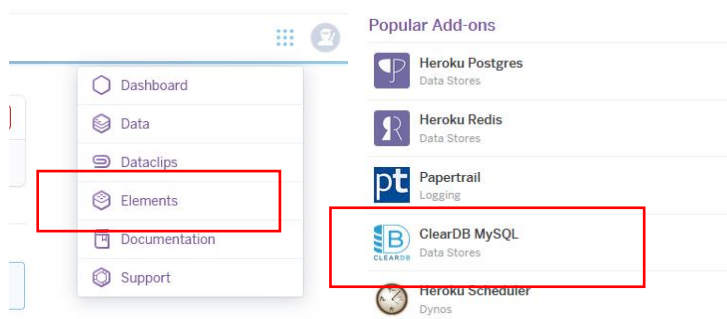
Nombre	Fecha de modificación	Tipo	Tamaño
 758.4c96db9f.js	13/06/2022 20:26	Archivo de origen ...	59 KB
 app.b75294ae.css	13/06/2022 20:26	Archivo de origen ...	1 KB
 app.beff9090.js	13/06/2022 20:26	Archivo de origen ...	3 KB
 app.css	22/05/2022 17:20	Archivo de origen ...	1 KB
 app.js	22/05/2022 17:20	Archivo de origen ...	19 KB
 entrypoints.json	13/06/2022 20:26	Archivo de origen ...	1 KB
 manifest.json	13/06/2022 20:26	Archivo de origen ...	1 KB
 runtime.61b1725c.js	13/06/2022 20:26	Archivo de origen ...	2 KB
 runtime.js	22/05/2022 17:20	Archivo de origen ...	15 KB
 vendors-node_modules_symfony_stimul...	22/05/2022 17:20	Archivo de origen ...	481 KB

Pasamos ya a Heroku, y siguiendo la [documentación oficial para Symfony](#), creamos un nuevo proyecto que vincularé en mi caso al repositorio de GitHub, añadimos a nuestro proyecto el archivo Procfile que indica que la carpeta principal es '/public' y configuramos la variable de APP_ENV a prod, lo cual se almacenará en Heroku con `heroku config:set APP_ENV=prod`. Añadimos todos los cambios y desde la web, seleccionamos Deploy Branch.



Cuando termine, tendremos la aplicación funcionando.

Lo siguiente es añadir una base de datos. Para esto nos vamos a Elementos y seleccionamos el addons ClearDB MySQL.



Seleccionamos la app que corresponda y nos habrá añadido una variable de entorno con el usuario, contraseña y el hostname. En el .env de nuestro proyecto cambiamos el enlace a local por este nuevo enlace. Y para acceder a esta base de

datos debemos usar un gestor de bases de datos, en mi caso lo haré con PHPMyAdmin. Para esto debemos añadir una nueva conexión con estas credenciales. Esto se hace editando el archivo config.inc en Xampp/PHPMyAdmin. Añadimos al final las siguientes líneas:

```
/* Heroku remote server */
$i++;
$cfg["Servers"][$i]["host"] = ""; //provide hostname
$cfg["Servers"][$i]["user"] = ""; //user name for your remote server
$cfg["Servers"][$i]["password"] = ""; //password
$cfg["Servers"][$i]["auth_type"] = "config"; // keep it as config
/*
 * End of servers configuration
*/
```

Y ya podremos acceder desde nuestro gestor e importar la base de datos para que esté disponible online.

Trabajo futuro

Queda por añadir varias cosas para un funcionamiento ideal como:

- Verificación del correo por email y recuperación de contraseña.
- Replantear las entidades relacionadas con Favoritos para una correcta

implementación de estos.

- Rediseño de algunas partes para un aspecto más profesional.
- Añadir otras funcionalidades como el “Planifica tu viaje”, que generaría

propuestas de rutas según las preferencias del usuario.

- Añadir galería de imágenes en otros elementos, como Patrimonio.
- Llevar el login a un modal y permitir hacer login con formulario con Ajax.
- Si finalmente se decide plantearlo para varios ayuntamientos, se debería

crear otra Entidad con la localidad y relacionarla con el resto de entidades

Conclusiones

Realizar el trabajo con Symfony ha ayudado a asentar los conocimientos básicos adquiridos sobre este Framework durante el curso, sin embargo, conforme me planteaba como realizar ciertas cosas de manera eficaz también me ha demostrado lo poco que realmente lo conozco. Por otro lado, con Twig siento que he avanzado algo más, ya que al principio no sabía como plantear ciertos elementos que requerían actualizarse sin recargar la página.

En definitiva, ha sido un trabajo que me ha permitido abrir horizontes hacia nuevas tecnologías y que quisiera seguir modelando conforme adquiera conocimientos en otros ámbitos, por ejemplo, me gustaría integrar algún framework para la parte de front, ya que me he dado cuenta de la utilidad de tener componentes y evitar recargas continuas en las web.