

21 DE ENERO DE 2023



# ASEREGE FRAMEWORK

## MEMORIA SOBRE LA WEB DE BARCELONA

EDGAR MUÑOZ MANJÓN

CENTRO DE FORMACION PROFESIONAL FPLLEFIÀ

TUTOR: JORDI EGEA GUIRADO

EVALUADORES: JORDI EGEA, CARLOS ARREBOLA, ANNA ROVIRA

## Contenido

1. Definición del proyecto y casos de uso .....	3
DIAGRAMAS DE CASOS DE USO .....	4
DIAGRAMA DE BASE DE DATOS .....	6
DIAGRAMA DE CLASES .....	7
2. BOCETOS .....	7
3. PROTOTIPOS .....	7
4. GUIA DE ESTILOS .....	11
5. Análisis de accesibilidad y usabilidad .....	14
6. Herramientas para implementar la Aplicación Web .....	14
Metodología a utilizar .....	14
Funcionamiento de la web .....	15
Clase UserAPI .....	15
Clase LoginAPI .....	16
Clase ImageAPI .....	17
Clase PictureAPI .....	17
Clase RolesAPI .....	18
7. Motivación .....	19
8. Planificación .....	20
9. Pruebas unitarias .....	22
Consideracions Front End .....	23
HTML .....	23
CSS .....	23
JavaScript .....	23
10. Managers .....	25
Recursos para realizar el FrontEnd .....	26
Consideraciones Back End .....	26
Modelo .....	26
Vista .....	27
Controlador .....	27
11. AES .....	29
Bibliotecas Externas .....	31
12. OpenAPI .....	31

APIS .....	31
13. Manual de desarrollo del programa .....	32
Instalacion Windows .....	32
Instalacion Linux .....	34
14. Publicación .....	36
Publicar en servidor Windows .....	36
Publicar en servidor Linux .....	43
15. Configuración .....	44
Json de Configuracion .....	44
16. Anexo:Codigo fuente .....	46
17. Licéncia y presupuesto .....	44
18. Referencias .....	46

## 1. Definición del proyecto y casos de uso

Para la creación de este sitio web, se ha utilizado el servidor AseregeHTMLFramework, el cual ha sido desarrollado en el lenguaje de programación C#.

Este servidor se encarga de gestionar los permisos de los usuarios, de la manipulación de la Base de Datos de MySQL y del sistema de validación del BackEnd propio.

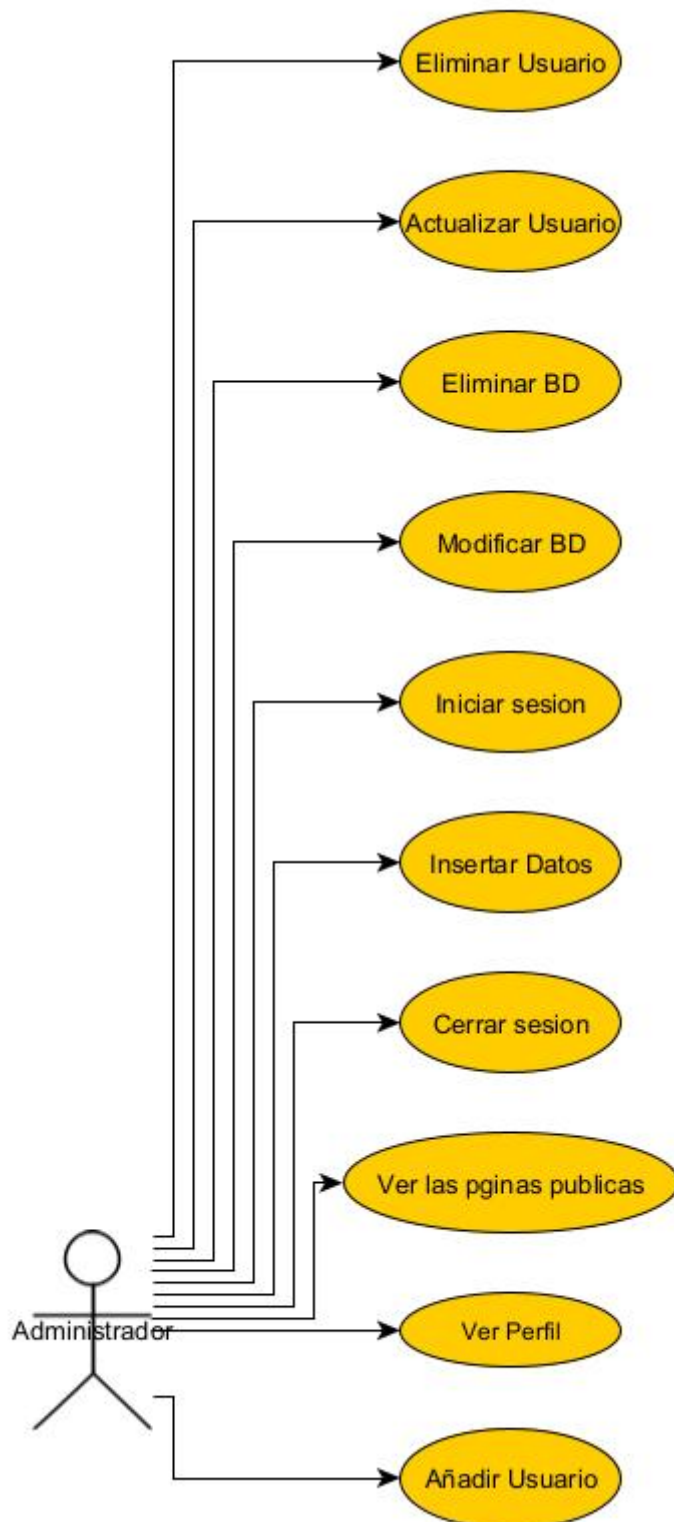
Además, el servidor está vinculado con el FrontEnd de la página web, el cual ha sido creado mediante la utilización de componentes de la biblioteca de MudBlazor y componentes personalizados que he creado.

Para poder conectar el FrontEnd con el BackEnd, se han utilizado diversas APIs que permiten la comunicación entre ambas partes, lo que también incluye la conexión con la Base de Datos de MySQL. Esto ha sido esencial para garantizar el correcto funcionamiento de la página web y para asegurarse de que los usuarios puedan disfrutar de una experiencia fluida y sin interrupciones.

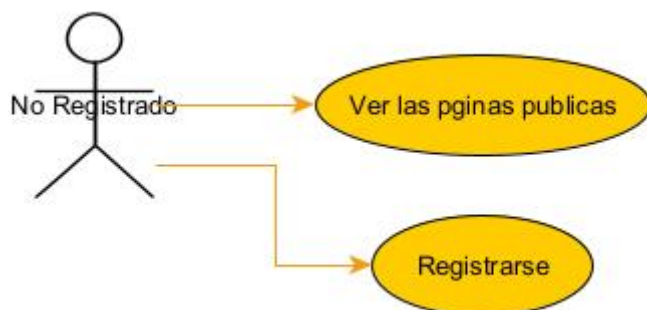
En definitiva, se ha puesto especial atención en la creación de una página web que ofrezca una amplia gama de información útil y relevante sobre la ciudad de Barcelona, todo ello implementando tecnologías de última generación para garantizar una experiencia de usuario excepcional.

## DIAGRAMAS DE CASOS DE USO

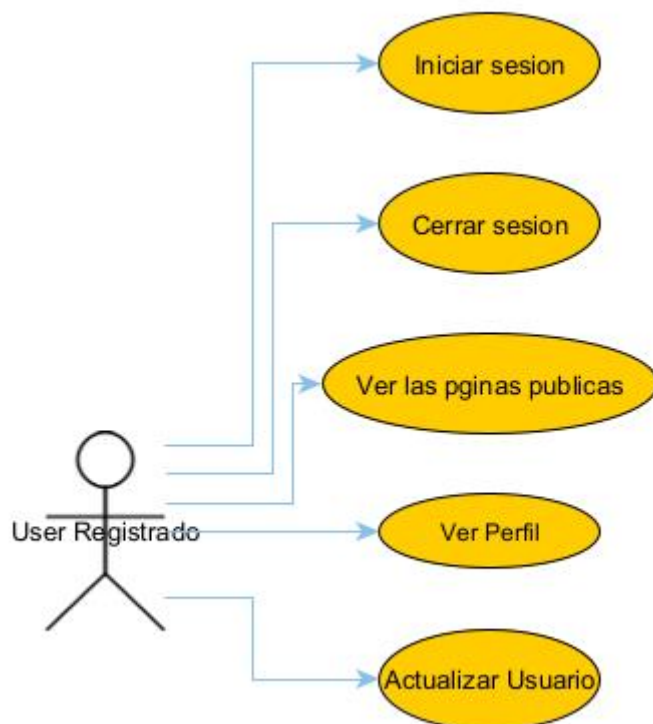
En este diagrama se puede ver los diferentes roles que tiene cada usuario:



*Ilustración 1. Representación de las funciones que puede tener el Administrador. Las páginas de transportes y visitas, es obligatorio registrarse o iniciar sesión previamente.*



*Ilustración 2. Representación de las funciones que puede tener un usuario anónimo o NO registrado.*



*Ilustración 3. Representación de las funciones que puede tener el usuario registrado en la pagina web.*

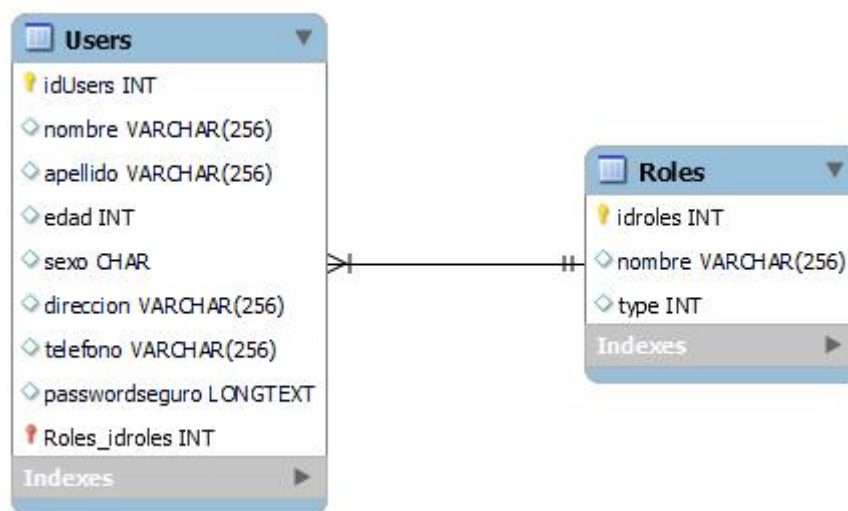


Existen diferentes tipos de roles:

- El administrador realiza **todas** las tareas de la gestión de información de la web. Este se encarga de **administrar permisos** a los usuarios, ya sean registrados o no.
- El usuario **NO** registrado puede ver las páginas web, pero no podrá acceder a sus diferentes secciones: mapa, contenido de la imagen... Hay que tener en cuenta que hay páginas que no puede acceder debido a que a la restricción que se ha mencionado anteriormente.
- El usuario registrado puede acceder las páginas web, cerrar la sesión, modificar su perfil. El usuario puede navegar por las páginas web, pero no puede modificar ni eliminar nada de información, incluyendo que no puede añadir datos a la base de datos.

## DIAGRAMA DE BASE DE DATOS

Esta es la representación del diagrama de clases de la base de datos:



*Ilustración 4. Representación de las tablas de la base de datos. La tabla Users, contiene los datos personales como el nombre, apellido y edad . La tabla Roles, contiene la información sobre el tipo de usuario y se diferencian mediante su ID.*

## DIAGRAMA DE CLASES

Esta es la representación del diagrama de clases (estructura global del proyecto):

## 2. BOCETOS

- Tarea 1: Bocetos (lápiz y papel) de las siguientes páginas:

No registrado:

- Registro

Registrado

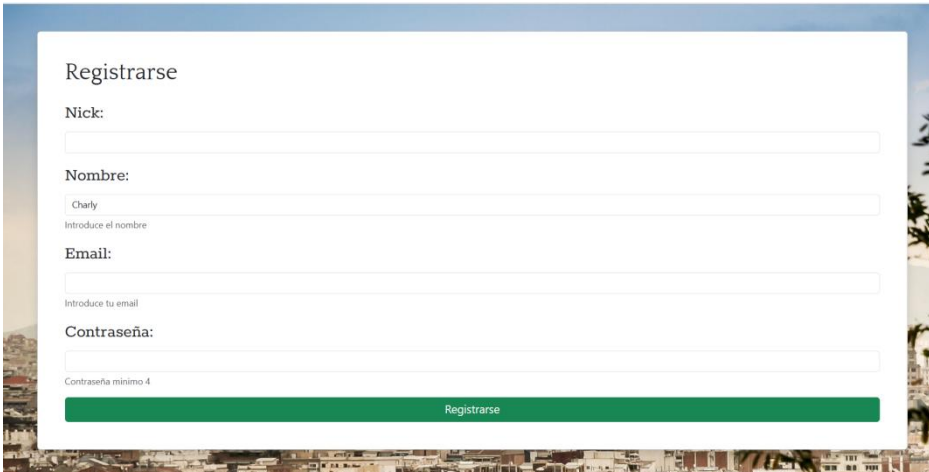
- Login
- Ver perfil
- Editar perfil

Administrador:

- a) Panel de usuarios con sistema de filtrado/ordenación por categorías y gestión crud

## 3. PROTOTIPOS

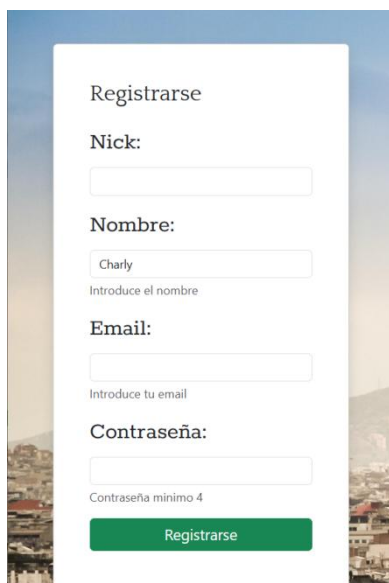
### Mockup



The mockup shows a registration form with the following fields and labels:

- Registrarse** (Title)
- Nick:** (Label) with a text input field.
- Nombre:** (Label) with a text input field containing the placeholder text "Charly". Below the field is the hint "Introduce el nombre".
- Email:** (Label) with a text input field. Below the field is the hint "Introduce tu email".
- Contraseña:** (Label) with a text input field. Below the field is the hint "Contraseña mínimo 4".
- Registrarse** (Green button at the bottom).





Registrarse

Nick:

Nombre:

Introduce el nombre

Email:

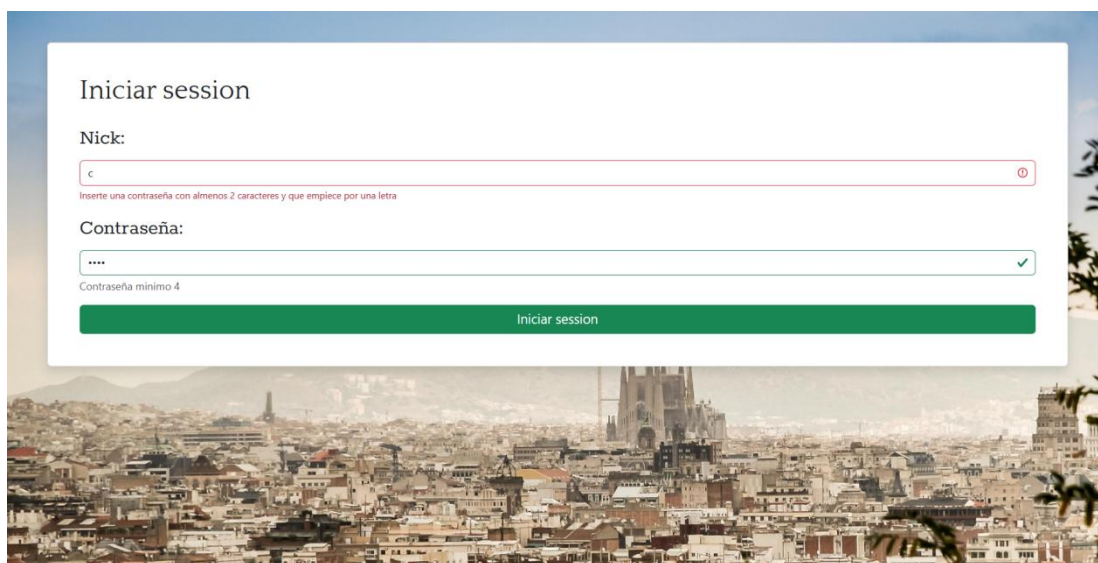
Introduce tu email

Contraseña:

Contraseña mínimo 4

Registrarse

*Ilustración 5: Página de registro que permite crear una cuenta nueva. El formulario está disponible para tabletas, ordenadores y móviles.*



Iniciar session

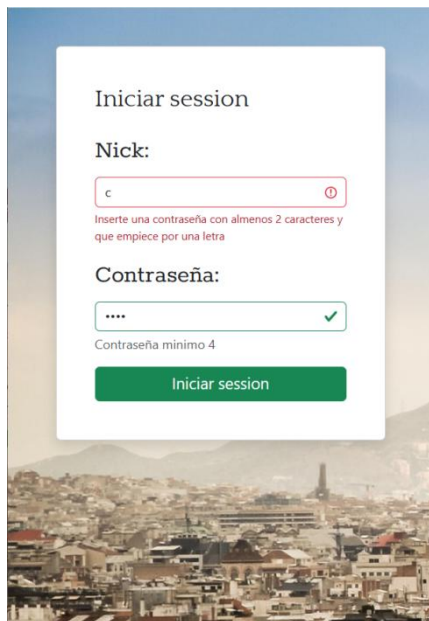
Nick:

Inserte una contraseña con al menos 2 caracteres y que empiece por una letra

Contraseña:

Contraseña mínimo 4

Iniciar session



Iniciar session

Nick:

c

Inserte una contraseña con al menos 2 caracteres y que empiece por una letra

Contraseña:

\*\*\*\*

Contraseña mínimo 4

Iniciar session

*Ilustración 6: La página de inicio de sesión es una página para que el usuario registrado, pueda volver a acceder a la web. En caso contrario, deberás ir al apartado ilustración 1.*

#### Editar usuario

Nombre

Edgar

Apellido

Muñoz

Edad

21

Sexo

M

Email

edgarmunozmanjon@gmail.com

Telefono

+34648401735

Password

Repetir Contraseña

FFF

CANCEL

*Ilustración 7: Página Ver perfil/Editar perfil*

#### PanelAdministracion

Id	Nombre	Apellido	Edad	Sexo	Email	Telefono	Contraseña	Rol	Editar	Borrar
1	Edgar	Muñoz	21	M	edgarmunozmanjon@gmail.com	+34648401735	SECRETO ***	1		
2	Jose	Pico	45	M	elpicasocas@gmail.com	+666666666	SECRETO ***	3		

Rows per page: 10 1-2 of 2

*Ilustración 8: Panel de usuarios con sistema de filtrado/ordenación por categorías y gestión crud*

## 4. GUIA DE ESTILOS

### *Css de la web general*

```
html, body {  
    font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;  
    background-color: #b5f3f3;  
}
```

```
a, .btn-link {  
    color: #0366d6;  
}
```

```
.btn-primary {  
    color: #fff000;  
    background-color: #1b6ec2;  
    border-color: #1861ac;  
}
```

```
.content {  
    padding-top: 1.1rem;  
}
```

```
.valid.modified:not([type=checkbox]) {  
    outline: 1px solid #26b050;  
}
```

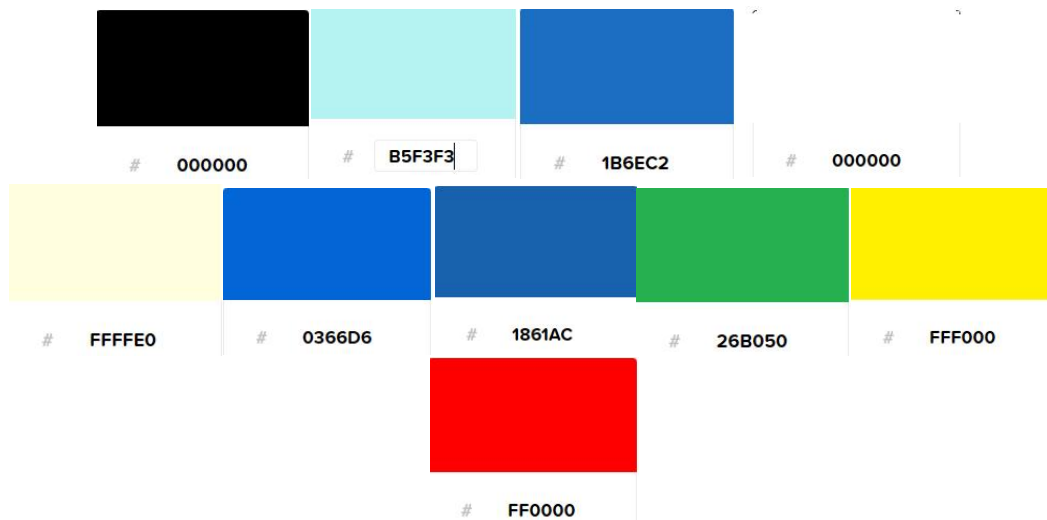
```
.invalid {  
    outline: 1px solid red;  
}
```

```
.validation-message {  
    color: red;  
}
```

```
#blazor-error-ui {  
    background: lightyellow;  
    bottom: 0;  
    box-shadow: 0 -1px 2px rgba(0, 0, 0, 0.2);  
    display: none;  
    left: 0;  
    padding: 0.6rem 1.25rem 0.7rem 1.25rem;  
    position: fixed;  
    width: 100%;  
    z-index: 1000;  
}
```

```
#blazor-error-ui .dismiss {  
    cursor: pointer;  
    position: absolute;  
    right: 0.75rem;  
    top: 0.5rem;  
}
```

### Paleta de colores



### Wireframe

El proyecto ha pasado por varias fases de planificación. Es importante tener en cuenta que se deben cumplir ciertos requisitos para satisfacer al usuario y asegurarse de que la página web se haya construido correctamente. Para lograr esto, se han creado bocetos que son diseños preliminares presentados al usuario para obtener su aprobación y continuar con el diseño y maquetación. También se han realizado los prototipos, que es la representación gráfica del proyecto con los datos que el usuario ha solicitado. Si todo ha ido bien, se prepara para maquetar, dando así el resultado final de la aplicación que el solicitante ha pedido.

//EJEMPLO DE WIREFRAME

## 5. Análisis de accesibilidad y usabilidad

Este enlace puede servirte de referencia para el siguiente tema: [Enlace](#)

El tema de la accesibilidad y usabilidad es **MUY IMPORTANTE** ya que podrás lograr captar la atención de los usuarios, ya sean gente con discapacidad o no. También, **es fundamental** tener en cuenta que estos análisis **valoran** tu capacidad de desarrollar la web / aplicación y poder **mejorar** cada proyecto que vayas a realizar.

Hay que tener en cuenta varios factores, que **determinan** el **orden y pulcritud** del diseño realizado. Por ejemplo: los menús deben de ir al **margen izquierdo** debido a que la utilizad, el formato de búsqueda y tener más facilidad a la hora de leer el texto que hayas utilizado. Otro punto a favor es tener un control para las imágenes, es decir, que puedas acceder a cada imagen de manera sencilla y la resolución es mucho más perceptible o visual.

Por contra, hay que **evitar la sobrecarga** de texto, dado que el usuario **IGNORARA** el plan de trabajo mencionado anteriormente. El tipo de tipografías o la variedad de colores que necesites es bastante necesario debido a que el estímulo del usuario se enfoque a la propuesta que realices. Por ejemplo, el tipo de botones que utilices, ya que el aspecto de la web /aplicación se podrá manipular más elemental la atención del lector. Un claro ejemplo es el siguiente: Verdadero (verde) y FALSO (rojo).



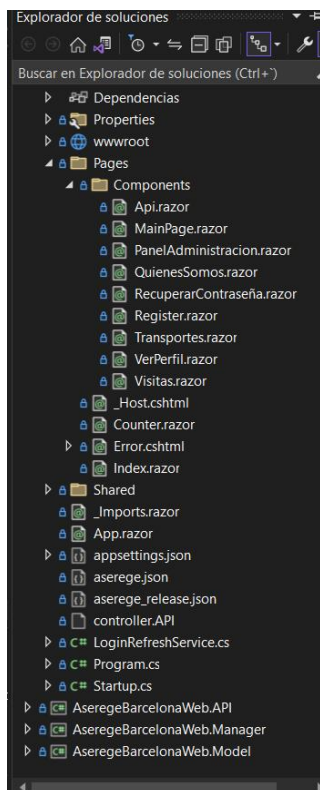
## 6. Herramientas para implementar la Aplicación Web

### Metodología a utilizar

Se ha escogido una metodología ágil y personalizada que se adapta a todos los recursos que el usuario ha solicitado. Se ha tenido en cuenta las tecnologías implementadas mencionadas anteriormente, comprobando el funcionamiento de la web y solucionando los errores que iban apareciendo. También se ha tenido en cuenta de la organización de los ficheros y procedimientos, puesto que es más cómodo y ágil a la hora de diseñar los elementos y funciones que la web va a necesitar.

Por ejemplo, los componentes se distribuyen en una carpeta diferente respecto a las APIs. En la siguiente imagen, se puede ver cómo está estructurado las carpetas con sus respectivos ficheros:





## Funcionamiento de la web

Las APIs que se han utilizado para desarrollar la web son las que se va a explicar a continuación.

### Clase UserAPI

Esta clase es una API de usuario que proporciona endpoints HTTP para realizar operaciones CRUD (Create, Read, Update, Delete) en la tabla de usuarios de una base de datos MySQL. La clase pertenece al espacio de nombres "AseregeBarcelonaWeb.API".

La clase está decorada con el atributo "ApiController" que indica que es una API controller, y con el atributo "Route" que especifica la ruta base para todos los endpoints de la API, que es "/api/users".

El endpoint HTTP GET es responsable de devolver todos los registros de usuario almacenados en la tabla de la base de datos. La acción primero obtiene los usuarios de la base de datos utilizando la clase "MySQLManager", que se encarga de realizar la operación de consulta y devuelve los registros en forma de arreglo de objetos "User". Después de eso, se elimina el campo "Passwordseguro" de cada objeto "User" para ocultar la contraseña en la respuesta JSON que se devuelve. Finalmente, la respuesta se devuelve como un objeto JSON con formato utilizando la clase JsonConvert de la biblioteca Newtonsoft.Json.

El endpoint HTTP POST se utiliza para agregar un nuevo registro de usuario a la tabla de la base de datos. La acción recibe un objeto "User" en el cuerpo de la solicitud HTTP (representado por el atributo [FromBody]), que contiene los datos del nuevo usuario a agregar. La acción utiliza la clase "MySQLManager" para insertar el registro en la tabla de la base de datos y devuelve el resultado de la operación.

El endpoint HTTP PUT se utiliza para actualizar un registro de usuario existente en la tabla de la base de datos. La acción recibe el parámetro "id" en la ruta de la solicitud HTTP, que indica el identificador del usuario a actualizar, y un objeto "User" en el cuerpo de la solicitud HTTP, que contiene los nuevos datos del usuario. Este endpoint se utiliza para el formulario de registro o en el botón de Editar Perfil, en el que, una vez el usuario haya introducido los datos y haya hecho click en el botón Registrarse, la API captura los valores y hace una nueva instancia en la Base de Datos, donde actualizará e insertará los datos modificados. Si el usuario no existe, y ha hecho el procedimiento para registrarse, la base de datos registrará a ese usuario nuevo. Si ya está registrado, irás al apartado del Login donde actuará dicha API.

El endpoint HTTP DELETE se utiliza para eliminar un registro de usuario existente en la tabla de la base de datos. La acción recibe el parámetro "id" en la ruta de la solicitud HTTP, que indica el identificador del usuario a eliminar. La acción utiliza la clase "MySQLManager" para eliminar el registro de la tabla de la base de datos y devuelve el resultado de la operación.

Se debe tener en cuenta que la API users también es el conector de la base de datos con el formulario de login y registro. Esta está vinculada a la tabla Roles, que hace referencia al rol del usuario (Administrador o Usuario Anónimo). También se encuentra el Usuario Registrado, que es el usuario que NO tiene los mismos permisos que el Administrador, pero sí que tiene unos permisos que debe cumplir para poder visitar la página. Las APIs heredan de la clase ControllerBase. La clase de la cual hereda tiene como función proporcionar una funcionalidad común a todos los controladores de la aplicación. La clase ControllerBase proporciona algunas características comunes para todos los controladores, como la capacidad de procesar solicitudes HTTP y la capacidad de acceder a los servicios de la aplicación.

Al heredar de la clase ControllerBase, los controladores pueden aprovechar la funcionalidad proporcionada por la clase base, lo que facilita la creación de controladores personalizados para su aplicación. Algunas de las características proporcionadas por la clase ControllerBase incluyen:

- Acceso a los servicios de la aplicación mediante la propiedad HttpContext.RequestServices.
- Métodos para devolver respuestas HTTP, como Ok() y NotFound().
- Propiedades y métodos para trabajar con el contexto de la solicitud HTTP actual.

### Clase LoginAPI

La clase LoginAPI es una API de autenticación que permite a los usuarios autenticarse en la aplicación web. La clase contiene cuatro métodos HTTP (GET, POST, PUT y DELETE), pero solo se utiliza el método POST para realizar la autenticación del usuario. El método GET se utiliza para obtener información sobre el usuario, mientras que los métodos PUT y DELETE se utilizan para actualizar y eliminar información del usuario.

El método POST recibe una solicitud HTTP con un objeto JSON que contiene las credenciales del usuario (nombre de usuario y contraseña) y, después de validar la información con la base de datos, devuelve un mensaje indicando si la autenticación ha sido exitosa o no. La clase

utiliza la clase `MySQLManager` para acceder a la base de datos y realizar las operaciones de autenticación y validación.

### Clase `ImageAPI`

La clase `ImageAPI` define una ruta de acceso [`Route("api/images")`] que coincide con la URL `https://<nombre del servidor>/api/images`. El atributo [`ApiController`] indica que la clase debe usar el comportamiento predeterminado de Web API, como la validación automática de modelos.

El **método `Get()`** es un método de acción que se invoca cuando se hace una solicitud HTTP GET a la ruta de acceso definida. En este método, se crea una instancia de la clase `MySQLManager` que maneja la interacción con la base de datos MySQL. Luego, se llama al método `SelectCountImages()` de la clase `MySQLManager` para obtener el número de imágenes en la base de datos. Finalmente, se llama al método **`DisposeAsync()`** de la **clase `MySQLManager`** para liberar los recursos utilizados por la instancia.

El resultado se devuelve como una respuesta HTTP 200 (OK) que contiene el número de imágenes como un objeto JSON. El número de imágenes se pasa como parámetro al **método `Ok()`** que devuelve una instancia de la clase `OkObjectResult` que envuelve el objeto JSON.

En resumen, esta clase define un controlador que maneja solicitudes HTTP GET a la ruta de acceso "api/images". El controlador utiliza la clase `MySQLManager` para interactuar con una base de datos MySQL y devuelve el número de imágenes como una respuesta HTTP 200.

### Clase `PictureAPI`

Esta clase define un controlador que maneja las solicitudes HTTP GET, POST, PUT y DELETE a la ruta de acceso `/api/picture/{key:int}`. El parámetro `key` es una variable de ruta que especifica el identificador de la imagen en la base de datos.

El atributo [`ApiController`] indica que la clase debe utilizar el comportamiento predeterminado de Web API, como la validación automática de modelos. El atributo [`Route`] especifica la ruta base para todas las acciones de este controlador.

El método `Get()` es una solicitud HTTP GET que recibe un identificador `key` y devuelve la imagen correspondiente en formato JSON. En este método, se crea una instancia de la clase `MySQLManager` que maneja la interacción con la base de datos MySQL. Luego, se llama al método `SelectPicturesByIdAsync()` de la clase `MySQLManager` para obtener la imagen correspondiente al identificador `key`. Finalmente, se llama al método `DisposeAsync()` de la clase `MySQLManager` para liberar los recursos utilizados por la instancia.

El método `Post()` es una solicitud HTTP POST que recibe un objeto `Picture` como cuerpo de la solicitud. Este objeto representa la información de la imagen que se va a insertar en la base de datos. En este método, se crea una instancia de la clase `MySQLManager` que maneja la interacción con la base de datos MySQL. Luego, se verifica si el usuario que realiza la solicitud está autorizado y si la contraseña es correcta utilizando la clase `CryptographyManager`. Si el usuario está autorizado, se llama al método `InsertFile()` de la clase `MySQLManager` para

insertar la imagen en la base de datos. Si la inserción es exitosa, se devuelve una respuesta HTTP 200 OK que contiene la URI de la imagen insertada.

El método Put() es una solicitud HTTP PUT que recibe un identificador id y un objeto Role como cuerpo de la solicitud. En este método, se debe definir el código necesario para actualizar una imagen en la base de datos, pero en este caso el cuerpo del método simplemente devuelve una respuesta HTTP 200 OK sin realizar ninguna acción.

El método Delete() es una solicitud HTTP DELETE que recibe un identificador id. En este método, se debe definir el código necesario para eliminar una imagen de la base de datos, pero en este caso el cuerpo del método simplemente devuelve una respuesta HTTP 200 OK sin realizar ninguna acción.

En resumen, esta clase define un controlador API que maneja las solicitudes HTTP GET, POST, PUT y DELETE a la ruta de acceso /api/picture/{key:int}. Los métodos Get() y Post() interactúan con la base de datos MySQL para recuperar y insertar imágenes, respectivamente, mientras que los métodos Put() y Delete() no realizan ninguna acción real en este caso.

### Clase RolesAPI

Esta clase es una API de roles que proporciona endpoints HTTP para realizar operaciones CRUD (Create, Read, Update, Delete) en la tabla de roles de una base de datos MySQL. La clase pertenece al espacio de nombres "AseregeBarcelonaWeb.API".

La clase está decorada con el atributo "ApiController" que indica que es una API controller, y con el atributo "Route" que especifica la ruta base para todos los endpoints de la API, que es "/api/roles".

El endpoint HTTP GET es responsable de devolver todos los registros de roles almacenados en la tabla de la base de datos. Sin embargo, en la implementación actual de la acción, simplemente devuelve una respuesta HTTP 200 OK vacía sin realizar ninguna operación de consulta en la base de datos. Es necesario agregar la lógica para obtener los registros de la tabla de roles de la base de datos y devolverlos como una respuesta JSON formateada.

El endpoint HTTP POST se utiliza para agregar un nuevo registro de rol a la tabla de la base de datos. La acción recibe un objeto "Role" en el cuerpo de la solicitud HTTP (representado por el atributo [FromBody]), que contiene los datos del nuevo rol a agregar. La acción utiliza la clase "MySQLManager" para insertar el registro en la tabla de la base de datos y devuelve el resultado de la operación.

El endpoint HTTP PUT se utiliza para actualizar un registro de rol existente en la tabla de la base de datos. La acción recibe el parámetro "id" en la ruta de la solicitud HTTP, que indica el identificador del rol a actualizar, y un objeto "Role" en el cuerpo de la solicitud HTTP, que contiene los nuevos datos del rol. Sin embargo, la acción aún no se ha implementado completamente y se necesita agregar lógica para actualizar el registro en la base de datos.

El endpoint HTTP DELETE se utiliza para eliminar un registro de rol existente en la tabla de la base de datos. La acción recibe el parámetro "id" en la ruta de la solicitud HTTP, que indica el

identificador del rol a eliminar. La acción utiliza la clase "MySQLManager" para eliminar el registro de la tabla de la base de datos y devuelve el resultado de la operación.

También se ha utilizado Web Components, que permite desarrollar la aplicación de manera modular, siendo eficaz la organización y la estructuración de la web. Los Web Components son una tecnología de desarrollo web que permite la creación de elementos personalizados reutilizables y encapsulados, que pueden ser utilizados en diferentes páginas web. Estos componentes son construidos utilizando HTML, CSS y JavaScript, y se pueden importar y utilizar en cualquier página web, sin importar el framework o librería de front-end utilizado.

Por otro lado, Blazor es una tecnología de Microsoft que permite desarrollar aplicaciones web interactivas utilizando C# en lugar de JavaScript aunque también se puede implementar estructuras de JavaScript. Por ejemplo:

```
@inject IJSRuntime jsRuntime  
await jsRuntime.InvokeAsync<string>("localStorage.getItem", "Name");
```

Blazor Server-Side es un modelo de implementación de Blazor que utiliza una arquitectura de servidor-cliente, en la que la mayor parte del código de la aplicación se ejecuta en el servidor y solo se envía la interfaz de usuario al cliente.

En Blazor Server-Side se pueden crear Web Components utilizando C# y Razor syntax para construir componentes personalizados reutilizables. Los Web Components creados en Blazor Server-Side pueden ser utilizados en cualquier página web que utilice Blazor Server-Side como modelo de implementación, lo que permite una mayor modularidad y reutilización de código.

## 7. Motivación

La motivación ha sido aprender el lenguaje de programación que se ha utilizado, además de mejorar mis habilidades de aprendizaje. Otra de las razones que me han llevado a este punto ha sido ofrecer a los usuarios una página web que permita poder utilizarse en cualquier dispositivo, tener más recursos para poder navegar e informarse sobre zonas de interés (visitas), recomendaciones y consejos, además de tener un enlace que permita ver más información sobre los transportes públicos y más consejos sobre Barcelona.

Hoy en día, hay páginas que no son accesibles ni usables, por lo tanto, se ha optado por realizar la web, teniendo en cuenta la fiabilidad y la disposición de todos los usuarios, ya sean turistas o no. También, otra de las razones fue lo mencionado anteriormente.

## 8. Planificación

Para planificar el proyecto, se ha utilizado la herramienta de Notion.so, que consiste en una web en la cual, puedes organizarte en tareas para poder diseñar de manera cómoda y eficaz.

Nombre de la Tarea	Descripción	Horas para el diseño	Finalización
Diagrama Base de Datos	Realizar la estructura de las tablas de la base de datos	6 horas	90 %
Diagrama de Clases	Realizar la estructura de organización de las clases utilizadas	Falta por realizarla	0%
Diagrama de Casos de Uso	Realizar la administración de permisos que puede tener cada usuario	2 horas	100 %
Escalar las imágenes y textos de la vista HOME para version movil	Hacer que las imágenes se puedan ver perfectamente en móvil, tablet y ordenador	25 horas	100 %
API Open API 3	Realizar las APIS, que conectarán el FrontEnd y el BackEnd	50 horas	90 %
Registro	Hacer el componente para que el usuario permita registrarse a la web	30 horas	90 %
Recuperación de Contraseña	Permite al usuario obtener una contraseña nueva	Falta realizarla	0%
Login	Permite al usuario logearse para acceder a la web	25 horas	100%

<b>Visitas</b>	Permite ver al usuario el apartado Visitas de la web	15 horas	100 %
<b>Transportes</b>	Permite ver al usuario el apartado de Transportes de la web	35 horas	89 %
<b>Panel de Administración</b>	Permite ver al usuario con rol de Administrador los usuarios que hay registrados. También puede editar y eliminar.	20 horas	100 %
<b>Componente de Picture Card</b>	Es un elemento interno, que su función es pintar en la web las fichas de los lugares con su información	20 horas	100 %
<b>Quiénes Somos?</b>	Permite ver al usuario información de la persona que ha realizado la web.	20 horas	75 %
<b>Página de Error</b>	Permite ver al usuario la página de Error 404 cuando la página no existe	10 horas	100 %
<b>Ver los JSON de los usuarios</b>	Permite ver al usuario los usuarios que hay almacenados en la base de datos cuando pone en la url /api/users	20 horas	100 %
<b>Subir imágenes / videos</b>	Permite al usuario subir fotos o videos (Videos de	2 horas	100%



	YouTube NO)		
<b>Diseño</b>	Realización de los bocetos y los prototipos	24 horas	100 %
<b>Realización de la web</b>	Realizar toda la web	2992 horas (4 meses)	90 %
<b>Redacción de la memoria y presentación</b>	Escribir la memoria y realizar/preparar la presentación oral	180 horas	70 %
<b>Evitar la política de CORS</b>	Evitar que la web de problemas por la política de CORS	5 horas	100%
<b>Satisfacción del aprendizaje</b>			100 %

*Tabla 1: Planificación de las tareas del proyecto*

## 9. Pruebas unitarias

Las pruebas unitarias son una práctica fundamental en el desarrollo de páginas web y aplicaciones. Estas pruebas son esenciales porque permiten evaluar la eficacia de los elementos implementados en el sistema, lo que a su vez ayuda a garantizar la calidad y el rendimiento del producto final.

Cuando se desarrolla una página web o una aplicación, se utilizan diferentes elementos, como funciones, módulos y componentes. Cada uno de estos elementos debe ser probado individualmente para asegurarse de que cumple con su función correctamente. Estas pruebas se conocen como pruebas unitarias, y se llevan a cabo a nivel de código, lo que significa que se prueban fragmentos de código específicos para asegurarse de que funcionan correctamente.

Las pruebas unitarias son importantes por varias razones. En primer lugar, ayudan a detectar errores y problemas en el código antes de que se conviertan en problemas más grandes en la página web o aplicación. Esto ahorra tiempo y dinero, ya que se evitan los costos asociados con la corrección de errores en etapas posteriores del desarrollo.

Además, las pruebas unitarias ayudan a garantizar la calidad del producto final. Al probar cada elemento individualmente, se asegura que cada uno cumpla con su función de manera adecuada, lo que se traduce en un producto final más sólido y fiable.

Otra ventaja de las pruebas unitarias es que permiten a los desarrolladores realizar cambios y actualizaciones con más confianza. Cuando se realizan cambios en un elemento específico, las pruebas unitarias pueden ser utilizadas para asegurarse de que los cambios no afecten negativamente a otras partes del sistema.

En resumen, las pruebas unitarias son esenciales para el desarrollo de páginas web y aplicaciones de calidad. Al probar cada elemento individualmente, se asegura que el producto final sea sólido y fiable, lo que a su vez se traduce en una mejor experiencia para los usuarios.

## Consideracions Front End

El frontend está realizado mediante el lenguaje de JavaScript. Hay que tener en cuenta que a la hora de realizar el frontend, tienes que utilizar diferentes bibliotecas, ya que es bastante necesario para poder desarrollar el proyecto de manera más sencilla y cómoda. **NO** hay que olvidar que el proyecto debe de ser **responsive**, es decir, tiene que verse de manera correcta en versión móvil y para tablet. Pero antes de nada, hay que saber los siguientes conceptos:

**HTML:** es un lenguaje de marcado, cuyas siglas en inglés significan **Hypertext Markup Language** y se utiliza para crear páginas web. Se compone de etiquetas que indican al navegador cómo mostrar el contenido de la página, como el texto, las imágenes, los enlaces y otros elementos multimedia. Es la base de la mayoría de los sitios web y se utiliza en conjunto con otros lenguajes de programación y tecnologías para crear sitios web interactivos y dinámicos.

**CSS:** sus siglas en inglés significan **Cascading Style Sheets**. Este lenguaje se utiliza para definir la presentación y el estilo de una página web creada con HTML. Con CSS, los desarrolladores pueden especificar cómo se ven los elementos de la página, incluyendo el tamaño, el color, la fuente y la disposición. Esto permite separar la estructura del contenido (HTML) de su estilo (CSS), lo que facilita la gestión y el mantenimiento de los sitios web y permite un mayor control sobre su aspecto y diseño.

**JavaScript:** es un lenguaje de programación utilizado para crear interactividad y dinamismo en páginas web. Es un lenguaje interpretado por el navegador que permite a los desarrolladores agregar funcionalidades avanzadas a las páginas web, como formularios interactivos, animaciones, efectos visuales y lógica de programación. JavaScript también se utiliza para crear aplicaciones web complejas, como juegos en línea y aplicaciones de comercio electrónico. Es uno de los lenguajes de programación más populares y se ejecuta en todos los principales navegadores web.

Es importante hablar de **Bootstrap**. Bootstrap es un **framework** o conjunto de **herramientas de diseño web**, basado en HTML, CSS y JavaScript, que se utiliza para desarrollar sitios web responsivos y con un diseño moderno y atractivo. Este framework incluye una gran cantidad de componentes predefinidos, como botones, formularios, menús de navegación, y otros elementos de interfaz gráfica de usuario, que se pueden personalizar para adaptarse a las necesidades de cada proyecto.

También es importante trabajar de manera modular, es decir, realizar web components. Los web components, son fundamentales a la hora de diseñar, debido a que es más sencillo distribuir los estilos CSS, la lógica de JavaScript y el código HTML por separado. Para ello, crea el **scaffolding** (sistema de carpetas) que veas conveniente para guardar los documentos con **extensión .css, .html o .js** que vas a utilizar.

Ahora, vamos a hablar de las **vistas**. Las vistas són las páginas web que se van a mostrar en el navegador, pero, con la lógica del JavaScript y la función de trabajar modulamente, se puede alternar las vistas sin cambiar de página principal.

Una vez se haya entendido los conceptos anteriores, seguimos con el procedimiento.

El programa utiliza la biblioteca MudBlazor. Crea una disposición con una barra de aplicaciones, un cajón y un área de contenido principal, con varios componentes MudBlazor utilizados para dar estilo y funcionalidad.

Los componentes MudThemeProvider, MudDialogProvider y MudSnackbarProvider envuelven toda la disposición y proporcionan funcionalidad de tematización, diálogo y snackbar, respectivamente.

Dentro de MudLayout, hay un componente MudAppBar que contiene un MudIconButton, MudSpacer, MudChipSet y MudMenu. MudIconButton se utiliza para alternar el cajón lateral, y MudChipSet contiene un conjunto de componentes MudChip que actúan como enlaces de navegación a diferentes páginas dentro de la aplicación. MudMenu contiene un avatar de usuario y elementos de menú desplegable para registrarse, iniciar sesión, acceder al perfil del usuario y cerrar sesión.

El componente MudDrawer contiene un MudDrawerHeader y un MudPaper con un MudNavMenu dentro, que proporciona enlaces de navegación adicionales a diferentes páginas dentro de la aplicación.

El componente MudMainContent contiene el contenido principal de la aplicación, que se determina por la variable SelectedPage en la instrucción switch. En este fragmento de código, se utiliza para representar condicionalmente diferentes páginas (MainPage, Register y un formulario de inicio de sesión) dependiendo de la selección del usuario en MudChipSet.

A continuación, vamos a hablar de clases. Una **clase** es un tipo de **objeto** que permite definir un conjunto de **propiedades** y **métodos** que pueden ser utilizados para crear objetos. Una **propiedad** es una variable que existe dentro de una clase. **Puede ser pública o privada.**

Las propiedades publicas se puede acceder desde fuera de la clase. Para ello, puedes utilizar la siguiente instrucción: **this.nombreVariable**.

Las **propiedades privadas** sólo se podrá acceder a ellas desde un método de clase. Para ello, se utiliza la siguiente instrucción: **this.#name**.

Un ejemplo de cómo se crea una clase:

```
public class User
{
    public int ID { get; set; }
    public string Nombre { get; set; }
    public string Apellido { get; set; }
    public int Edad { get; set; }
    public char Sexo { get; set; }
    public string Email { get; set; }
    public string Telefono { get; set; }
    public string Passwordseguro { get; set; }
```

```
public int Roles_idroles { get; set; }  
}
```

También se ha utilizado la biblioteca de **Multiavatar**. Multiavatar es una biblioteca de código abierto que proporciona un conjunto de funciones para generar avatares únicos y personalizados utilizando combinaciones de formas y colores. La biblioteca es "multi" porque permite generar un gran número de avatares diferentes a partir de una única fuente de datos.

Multiavatar se puede utilizar en una amplia variedad de aplicaciones, como redes sociales, juegos en línea, chats y foros, para permitir a los usuarios personalizar su avatar de una manera fácil y rápida. Los avatares generados por Multiavatar son ligeros, escalables y no contienen información personal identificable, lo que los hace ideales para ser utilizados en entornos en línea.

La biblioteca Multiavatar también es compatible con una amplia variedad de lenguajes de programación, como **JavaScript**, **Python**, **Ruby** y **PHP**, lo que la hace muy flexible y fácil de integrar en proyectos existentes.

Para instalarla, ejecutamos está orden: ***npm install multiavatar -D***.

O en nuestro caso : ***dotnet add package BlazorMultiavatar --version 0.9.0.2***.

## 10. Managers

El método Login, el usuario le pasa un objeto Authorize que se utiliza para iniciar y cerrar sesión del usuario. El método **login(Authorize user)** toma el nombre de usuario y la contraseña, cifra la contraseña y la guarda en el almacenamiento local del navegador. Una vez se haya finalizado el proceso, **es fundamental cerrar la base de datos**. Para ello, se utiliza el **método Dispose()**.

El método **generateHash(string seed, string data)** genera un hash encriptado utilizando el algoritmo SHA-256. El método **GeneratePasswordHash(string input)** cifra la contraseña del usuario utilizando el algoritmo AES y devuelve el valor cifrado en formato base64.

El programa interno funciona de la siguiente manera. En primer lugar, se define un socket de red en el cual se emitirá un buffer HTML. Seguido se abre un socket por medio de signalR que escuchara a todos los posibles clientes que puedan haber. Por cada cliente se le asigna una instancia de la clase Startup que contendrá los valores por defecto del servidor. En esa clase se inicializa un endpoint / para determinar la página principal del programa. Además se abren endpoints adicionales para Blazor y la API de Aserege Framework.

Aserege tiene un patron Observer que escucha todas las peticiones en los endpoints y realiza una tarea asincronica segun el caso de la peticion establecida. La peticion debe ser correcta para poder entrar en el endpoint.

```
//la clase hereda del código interno de ASP NET
[Route("api/login")] public class LoginAPI : ControllerBase
{
    //la función get, obtiene del body un JSON del tipo Authorize
    [HttpGet] public IActionResult Get([FromBody] Authorize model)
    {
        //devuelve una nueva instancia a la base de datos
        return Ok(new MySQLManager().GetUser(model));
    }
}
```

Para realizar la petición se envía de la siguiente forma dentro del Body por petición **POST**

```
{
    "Name": "Juan",
    "Password": "JuanChoas1234#@#"
}
```

Una vez enviada, el programa hará una excepción y ejecutará uno de los métodos de **CRUD (GET/POST/PUT/DELETE)** y devolverá el resultado de dicha petición establecida.

## Recursos para realizar el FrontEnd

Para más información sobre Javascript, puedes hacer click en el siguiente enlace: [Enlace](#).

Para utilizar bootstrap, puedes acceder al siguiente enlace: [Enlace](#).

Para probar la librería multiavatar, accede al siguiente enlace: [Enlace](#).

## Consideraciones Back End

El backend está realizado mediante el lenguaje de C#. Hay que tener en cuenta que a la hora de realizar el backend, tienes que utilizar diferentes bibliotecas y librerías, ya que hay clases internas del lenguaje que ya vienen predefinidas, es decir, que ya hay clases y métodos que puedes utilizar. Y si quieres utilizarlas, debes de realizar ese paso, ya que el programa dará **error** debido a que no encuentra la **librería** o **biblioteca** correspondiente. Por ejemplo, si quieres utilizar Multiavatar, debes de añadir en el fichero `_Imports.razor` la librería que utilizarás, de la siguiente manera: **@using BlazorMultiavatar**.

Es necesario que te organices de manera sencilla, ya que será más cómodo poder desarrollar este tipo de webs/aplicaciones. Por ejemplo, el **scaffolding** (sistema de carpetas) es muy necesario, debido a que puedes estructurar y almacenar en carpetas los ficheros del **MVC (Modelo - Vista - Controlador)**. Por si tienes alguna duda, se define brevemente que función hace cada uno.

**Modelo:** El Modelo actúa como una **interfaz** entre la Vista (que es la interfaz de usuario) y el Controlador (que maneja las interacciones del usuario con la aplicación), proporcionando los

datos necesarios para mostrarlos en la Vista y realizando las operaciones necesarias para actualizar la información en el sistema. En resumen, el Modelo es la **parte central** de la arquitectura MVC, que se encarga de la **manipulación y gestión de datos** en la aplicación.

**Vista:** la Vista es la capa de la aplicación que se encarga de mostrar la interfaz de usuario al usuario final. Es decir, la Vista es la parte de la aplicación que presenta los datos al usuario de una forma visualmente atractiva y fácil de entender.

La Vista **no** tiene conocimiento sobre cómo se manejan los datos en la aplicación, su único propósito es mostrarlos y permitir que el usuario interactúe con ellos. La Vista **recibe** los datos del Modelo y los **presenta** en una forma visualmente atractiva, ya sea en una página web, una aplicación móvil, una ventana de escritorio, etc.

La Vista también **puede contener** elementos interactivos, como botones, menús, formularios, etc., que permiten al usuario interactuar con los datos de la aplicación. Estos elementos suelen estar conectados al Controlador, que es el encargado de procesar las interacciones del usuario y actualizar el Modelo en consecuencia.

En resumen, la Vista es la capa de la aplicación que se encarga de presentar los datos al usuario de una forma visualmente atractiva y fácil de entender, sin conocer cómo se manejan los datos en la aplicación.

**Controlador:** el Controlador es la capa que se encarga de manejar las interacciones del usuario con la aplicación y actualizar el Modelo en consecuencia. Es decir, el Controlador actúa como un intermediario entre la Vista (interfaz de usuario) y el Modelo (datos y lógica de negocio).

Cuando el usuario interactúa con la Vista, los elementos interactivos (botones, formularios, etc.) envían una solicitud al Controlador. El Controlador procesa la solicitud y realiza las operaciones necesarias en el Modelo para actualizar los datos y mantener la consistencia del sistema.

El Controlador también es responsable de enviar datos de vuelta a la Vista para que se actualice la interfaz de usuario en consecuencia. Es decir, el Controlador coordina la comunicación entre la Vista y el Modelo.

Es importante destacar que el Controlador no tiene conocimiento sobre la presentación visual de los datos, ya que esa tarea es responsabilidad de la Vista. El Controlador sólo se enfoca en la lógica de negocio y en la manipulación de datos.

En resumen, el Controlador es la capa de la aplicación que maneja las interacciones del usuario con la aplicación y se encarga de actualizar el Modelo en consecuencia, manteniendo la consistencia del sistema. También coordina la comunicación entre la Vista y el Modelo, pero no tiene conocimiento sobre la presentación visual de los datos.

Una vez aclarado los apartados del MVC, podrás continuar con el procedimiento.

A continuación, vamos a desarrollar el **backend**. Hay que tener en cuenta los pasos mencionados anteriormente, ya que hay que tener presente que el proyecto que estas desarrollando es **seguro**, es decir, **evitar** a que gente externa pueda acceder a la base de datos y manipularlos o eliminar datos. Para ello, es fundamental hacer un **sistema de permisos** para gestionar cada rol de los usuarios que tengas. Puedes organizar el código de los ficheros de la manera que quieras, siempre y cuando sepas lo que estás haciendo en todo momento.

El **fichero AseregeHtmlFramework** es el fichero principal, que se encarga de realizar tantas instancias como usuarios haya en la base de datos, es decir, crea diferentes **Sockets** que la función principal expone a cada uno de los usuarios que se conectan al servidor. Eso **evita una sobrecarga** en la red o en el servidor. Este Socket utiliza el **puerto 5000**, debido a que es un puerto comúnmente utilizado para servidores web de Blazor y aplicaciones web en el **protocolo HTTP**. Se utiliza para permitir que los clientes (navegadores web u otras aplicaciones) accedan a los recursos web que ofrece el servidor. Además se deberá añadir un puerto adicional para el **protocolo HTTPS** (página segura) con **puerto 5001** ya que es obligatorio actualmente.

El **fichero MainLayout** además incluye el routing. El routing se compone de diversas funciones o tareas asincrónicas que permite poder gestionar la web de manera correcta y eficaz. Las tareas que se han implementado son las siguientes:

```
private void SelectedValuesChanged()
{
    if (selected != null)
    {
        SelectedPage = selected.Text;
    }
    else
    {
        StateHasChanged();
        SelectedPage = "blue";
    }
    StateHasChanged();
}
```

Esta tarea permite comprobar el estado del selector que hayamos hecho click. Esto se utiliza para la navegación del menú de la vista Home. En caso de que el recuso sea null (no exista), automáticamente, redirige a la página principal. Para tener la referencia, se ha creado un identificador (color del menú).

```
private async Task TravelTo(string page)
{
    SelectedPage = page;
    await Task.CompletedTask;
}
```

Esta tarea permite cambiar la vista mediante el identificador mencionado anteriormente.

```
protected override async Task OnAfterRenderAsync(bool firstRender)
```



```
{  
  
    await RefreshPermissions();  
  
    StateHasChanged();  
  
    await Task.CompletedTask;  
  
}
```

Esta función permite actualizar los permisos siempre y cuando, se haya cambiado de usuario. Por ejemplo, el usuario de Administrador tiene unos permisos diferentes al de un usuario registrado.

## 11. AES

Este backend contiene además el sistema de cifrado **AES**. El cifrado **AES (Advanced Encryption Standard)** es un algoritmo de cifrado simétrico que se utiliza para proteger la confidencialidad de los datos. Es muy popular en la actualidad debido a su eficiencia y seguridad. Los usuarios que trabajan con AES encuentran que es atractivo por varias razones.

Tiene las siguientes características:

- **Expansión de clave:** AES utiliza una sola clave durante la primera etapa, pero esta clave se expande en múltiples subclaves para su uso en cada una de las rondas individuales. La expansión de la clave se realiza mediante una serie de operaciones de rotación y sustitución que generan un conjunto de subclaves que se utilizarán en cada ronda.
- **Datos de bytes:** AES opera en datos de bytes en lugar de datos de bits. Esto significa que los datos se dividen en bloques de 128 bits que a su vez se dividen en 16 bytes. Cada byte se trata como un número entero en el rango de 0 a 255.
- **Longitud de la clave:** El número de rondas que se realizan depende de la longitud de la clave utilizada para cifrar los datos. El AES tiene tres tamaños de clave: 128 bits, 192 bits y 256 bits. El tamaño de clave de 128 bits tiene diez rondas de cifrado, el tamaño de clave de 192 bits tiene doce rondas y el tamaño de clave de 256 bits tiene catorce rondas.

Tiene las siguientes ventajas:

En primer lugar, AES es relativamente fácil de entender y de implementar. Esto significa que los desarrolladores pueden crear aplicaciones y sistemas que utilizan AES sin tener que dedicar demasiado tiempo y recursos a la programación y la implementación. También significa que los usuarios finales pueden comprender cómo funciona el cifrado AES y cómo protege sus datos.

En segundo lugar, AES es extremadamente rápido. Los tiempos de cifrado y descifrado son muy rápidos en comparación con otros algoritmos de cifrado. Esto significa que los usuarios pueden proteger sus datos sin tener que esperar demasiado tiempo para acceder a ellos.

En tercer lugar, AES requiere menos memoria que muchos otros tipos de cifrado. Esto significa que puede funcionar en sistemas con recursos limitados y no afectar el rendimiento general.

del sistema. Además, puede ser una gran ventaja en dispositivos móviles, donde el espacio de almacenamiento y la energía son limitados.

Por último, como mencionaste, AES se puede combinar con otros protocolos de seguridad para crear una capa adicional de protección. Por ejemplo, se puede utilizar AES junto con el protocolo de seguridad inalámbrica WPA2 para proteger las redes Wi-Fi. También se puede utilizar junto con otros tipos de encriptación como **SSL (Secure Sockets Layer)** para proteger la comunicación en línea.

En resumen, el cifrado AES es atractivo para quienes trabajan con él debido a su facilidad de implementación, rapidez, eficiencia en el uso de memoria y la capacidad de combinarse con otros protocolos de seguridad para proporcionar una capa adicional de protección.

Hay que tener en cuenta que el backend, está vinculado a la Base de Datos de MySQL, por lo cual, hay que tener un fichero de configuración dónde puedas almacenar la conexión a dicha base. Un ejemplo del fichero de configuración puede ser el siguiente:

```
Public class AseregeConfiguration
{
    public string hostname;
    public int port;
    public string username;
    public string password;
    public string databaseName;
}
```

**Hay que tener en cuenta que la contraseña debe guardarse cifrada porque así tienes más seguridad.**

El lenguaje SQL es un lenguaje que te permite conectarte a una Base de Datos remota o local que permite crear y manipular los datos, además de poder actualizar y eliminar. Este “idioma” se utiliza en estas Bases de Datos: MySQL, MariaDB, PostgreSQL...

Las características principales son:

- Es un lenguaje **NO** relacional
- Es un **lenguaje estándar**, lo que significa que existen normas y convenciones establecidas para su uso en todo el mundo.
- Puede trabajar con **múltiples** tablas a la vez, permitiendo la realización de operaciones complejas que involucran la combinación de datos de diferentes tablas.
- Es un **lenguaje interactivo**, lo que significa que los usuarios **pueden realizar** consultas y recibir respuestas en tiempo real.
- Incluye medidas de seguridad para proteger los datos almacenados en la base de datos, como la autenticación y la autorización de usuarios y roles.

## Bibliotecas Externas

Los ficheros mencionados anteriormente, utilizan las siguientes bibliotecas:

- **mysql-connector-j** -> esta biblioteca es la que se encarga de poder manipular las clases internas de manera sencilla, ya que puedes utilizar las clases cada vez que quieras.
- **jackson-databind** -> esta biblioteca es la que se encarga de gestionar las herramientas utilizadas para desarrollar las funciones que hemos implementado en nuestro diseño de la página web.

Eso sí, **se debe de recordad en importar las librerías internas que implementan en dichas bibliotecas.**

## Recursos Extras para la información de los apartados anteriores

En este enlace, puedes obtener más información sobre todas las clases e interfaces que tiene JAVA : [Enlace](#).

Para más información sobre el sistema de encriptación AES, puedes consultar el siguiente enlace: [Enlace](#).

Para obtener la biblioteca de MySQL, debes de acceder al siguiente enlace: [Enlace](#).

Para obtener la biblioteca jackson, debes de acceder a este enlace: [Enlace](#).

## 12. OpenAPI

OpenAPI es una herramienta que ayuda a los desarrolladores a diseñar, documentar y utilizar APIs de manera más eficiente y efectiva, lo que puede mejorar la interoperabilidad y la integración de sistemas en la web.

### APIS

Las apis utilizadas son GET, POST, PUT y DELETE, puesto que dependiendo de la instrucción que ejecutes, hará una petición asíncrona a la api que corresponda y lo ejecutará. Es muy necesario este tipo de comunicación asíncrona ya que evitamos la sobrecarga de peticiones y el programa deje de funcionar.

#### ■ GET :

- ◆ **GET /api/role** : esta api se encarga de devolver el tipo de rol que administra el usuario. Práviamente, se debe haber definido que tipo de rol conlleva (Administrador, Registrado, Anónimo).
- ◆ **GET /api/user** : esta api se encarga de devolver los usuarios creados y almacenados en nuestra Base de Datos.
- ◆ **GET /api/user{id:int}** : esta api se encarga de devolver el usuario creado que cumpla con la id que le hemos especificado.

#### ■ POST:

- ◆ **POST /api/role**: esta api se encarga de insertar un rol a la Base de Datos, que se envía desde el body del documento.

```
{ "Idroles":1, "Nombre":"Jefe", "Type":1}
```

- ◆ **POST /api/user** : esta api se encarga de insertar un usuario a la Base de Datos.

```
{ "idUsuario":1, "nombre":"Edgar", "apellido":"Muñoz",  
"edad":21, "sexo":"M", "email":"edgarmunozmanjon@gmail.com",  
"telefono":"+34648401735", "passwordseguro":"administradorHACKER",  
"Roles_idroles": "1" }
```

#### ■ **PUT:**

- ◆ **PUT /api/roles {nombre: string, int type}** : esta api se encarga de actualizar el rol almacenado en la Base de Datos.
- ◆ **PUT /api/usuario {nombre: String, apellido: String, edad: int, sexo: char, email: String, telefono: String, passwordseguro: String}** : esta api se encarga de actualizar el usuario almacenado en la Base de Datos.

#### ■ **DELETE:**

- ◆ **DELETE /api/roles {nombreRol: string}** : esta api se encarga de eliminar el rol del usuario que cumpla con el nombre del rol que le hemos especificado.
- ◆ **DELETE /api/roles {idrole: int}** : esta api se encarga de eliminar el rol registrado en la Base de Datos cuyo id cumpla con el identificador que le hemos especificado.
- ◆ **DELETE /api/usuario{idusuario, nombreRol: string}** : esta api se encarga de eliminar el usuario cuyo id o nombre cumplan con las especificaciones que le hayamos pasado.

## 13. Manual de desarrollo del programa

Para realizar la instalación del software AseregeHtmlFramework en Windows, se deben seguir los siguientes pasos:

### Instalacion Windows

1. En primer lugar, se debe instalar Node.js en el sistema. Para ello, se puede utilizar el comando "**winget install -e --id OpenJS.NodeJS**" en la línea de comandos, que instalará automáticamente la última versión de **Node.js**.
2. Si se prefiere, se puede descargar la versión específica de Node.js para la arquitectura del PC, ya sea x86 (32 bits) o x64 (64 bits), desde la página oficial de Node.js en los enlaces proporcionados. Se debe seguir los pasos del asistente de instalación para completar la instalación.

- ◆ x64 -> <https://nodejs.org/dist/v18.14.2/node-v18.14.2-x64.msi>  
x86 -> <https://nodejs.org/dist/v18.14.2/node-v18.14.2-x86.msi>

3. Una vez instalado Node.js, se debe instalar Git utilizando el comando "**winget install -e --id Git.Git**" en la línea de comandos.

4. A continuación, se debe clonar el repositorio de **AseregeHtmlFramework** desde GitHub utilizando el comando siguiente en la línea de comandos: `"git clone --recursive https://github.com/daw2-munoz22/AseregeHtmlFramework.git"`.
5. Después de clonar el repositorio, se debe acceder a la carpeta "**AseregeHtmlFramework**" y luego a la carpeta "**layout**" utilizando los comandos `"cd AseregeHtmlFramework"` y `"cd layout"`, respectivamente.
6. Una vez dentro de la carpeta "**layout**", se debe instalar las dependencias necesarias del proyecto utilizando el comando `"npm install"` en la línea de comandos.
7. Finalmente, se debe instalar JDK utilizando el comando `"winget install EclipseAdoptium.Temurin.17.JDK"` en la línea de comandos.
8. Una vez realizados estos pasos, se puede compilar la solución .pom con el editor de preferencia para empezar a utilizar el software **AseregeHtmlFramework** en Windows.
9. Siguiendo estos pasos, se podrá realizar la instalación del software **AseregeHtmlFramework** en Windows sin problemas. Es **importante** destacar que se deben seguir los pasos en el orden indicado para asegurarse de que todas las dependencias estén correctamente instaladas.

## Instalacion Linux

Para instalar el software **AseregeHtmlFramework** en Linux, debes seguir los siguientes pasos:

1. Actualiza la lista de paquetes disponibles en tu sistema con el siguiente comando:

```
sudo apt update
```

2. Instala el cliente de **Git** para descargar el repositorio con el siguiente comando:

```
sudo apt-get install git
```

3. Instala **Node.js** para poder ejecutar el servidor con el siguiente comando:

```
sudo apt-get install nodejs
```

4. Instala **MySQL** Server para tener una base de datos con el siguiente comando:

```
sudo apt-get install mysql-server
```

5. Inicia el servicio de **MySQL** con el siguiente comando:

```
sudo service mysql start
```

6. Verifica el estado del servicio de **MySQL** con el siguiente comando:

```
sudo service mysql status
```

7. Instala los paquetes necesarios para utilizar HTTPS con el siguiente comando:

```
sudo apt install apt-transport-https ca-certificates
```

8. Añade la clave pública de **AdoptOpenJDK** para poder instalar Java con el siguiente comando:

```
wget -qO - https://adoptopenjdk.jfrog.io/adoptopenjdk/api/gpg/key/public | sudo apt-key add -
```

9. Añade el repositorio de **AdoptOpenJDK** con el siguiente comando:

```
sudo add-apt-repository --yes https://adoptopenjdk.jfrog.io/adoptopenjdk/deb/
```

10. Para instalar la última versión de **Java**, escribe el siguiente comando y presiona **ENTER**:

Este comando instalará la última versión de Java disponible en la actualidad.

```
sudo apt install adoptopenjdk-latest-hotspot
```

11. Para instalar **Java 16** en su lugar, escribe el siguiente comando y presiona **ENTER**:

Este comando instalará la versión 16 de Java.

```
sudo apt install adoptopenjdk-16-hotspot
```

12. Escribe el siguiente comando en la terminal y presiona **ENTER** para comprobar la version de java instalada:

La terminal debería mostrar la versión de Java instalada en tu sistema operativo.

```
java -version
```

13. Escribe el siguiente comando en la terminal y presiona **ENTER**:

```
git clone --recursive https://github.com/daw2-munoz22/AseregeHtmlFramework.git
```

Este comando clonará el repositorio del proyecto en tu sistema operativo.

14. Escribe el siguiente comando en la terminal y presiona **ENTER**, este comando te llevará al directorio del proyecto.

```
cd AseregeHtmlFramework
```

15. Este comando te llevará al directorio "layout" del proyecto.

```
cd layout
```

16. Escribe el siguiente comando en la terminal y presiona Enter, este comando instalará las dependencias del proyecto.

```
npm install
```

17. Abre el archivo "**pom.xml**" del proyecto en el editor de tu preferencia.

- Compila el proyecto con el editor de tu preferencia. Este proceso puede variar dependiendo del editor que estés utilizando.
- Una vez que el proyecto haya sido compilado, podrás utilizarlo según sea necesario.



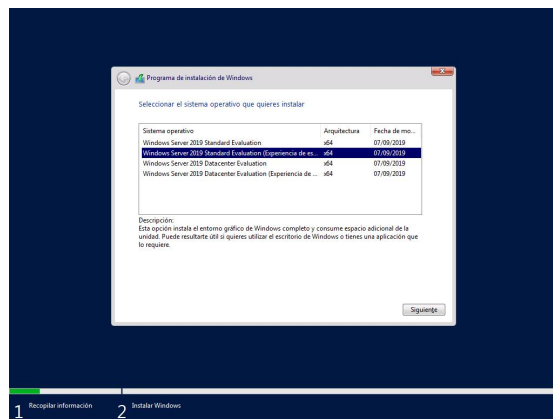
## 14. Publicación

### Publicar en servidor Windows

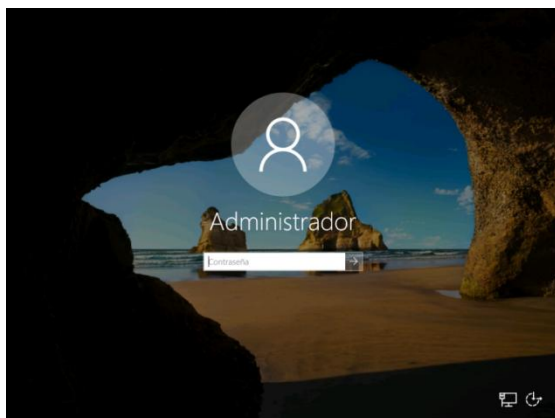
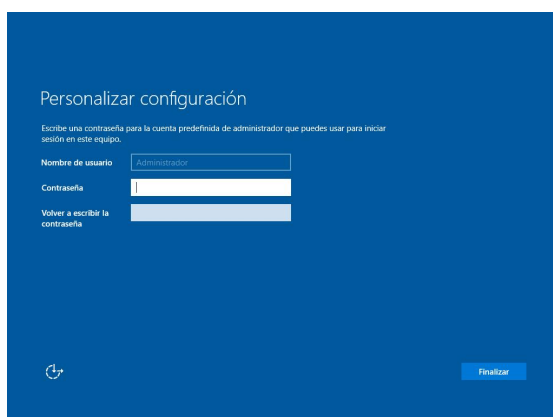
Para poder instalar la aplicación en cualquier ordenador con sistema operativo Windows Server, debemos de seguir las siguientes instrucciones:

a. Primero, vamos a instalar el sistema operativo mencionado anteriormente.

- Para ello, debemos crear una máquina virtual nueva o utilizar un **USB** con **Windows** en mano. Luego, debemos insertamos la imagen de disco **ISO** (ISO Universal Disk Format) y te aparecerá una ventana similar a la siguiente.



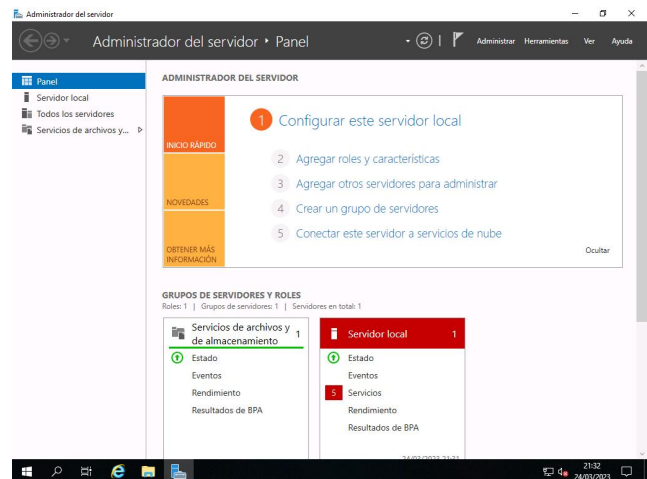
- Una vez que hayamos escogido nuestro sistema operativo, debemos de rellenar el siguiente **formulario**, donde nos piden el nombre del usuario que queremos crear y darle una contraseña.



- b. Una vez hayamos creado el usuario, aparecerá la siguiente pantalla donde efectivamente podremos **iniciar sesión** en la cuenta **Administrador** o **ROOT** del servidor para **gestionar** los usuarios que utilizará el cliente de **MySQL** y **AseregeServer**.

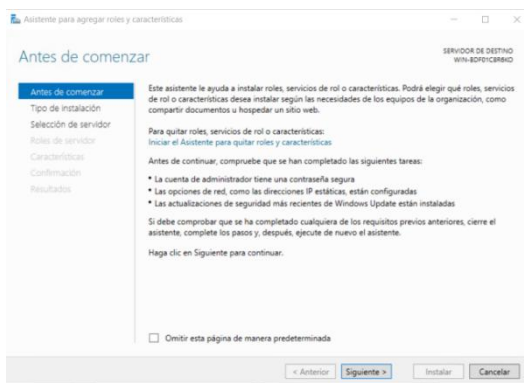
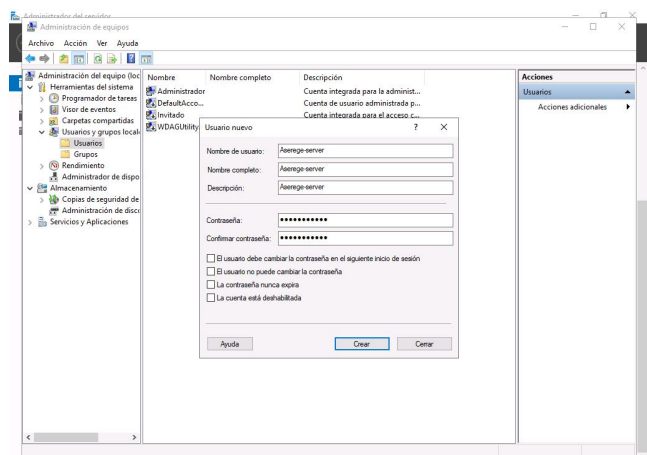
- c. A continuación, vamos al apartado de **Administrador del servidor** en el podemos observar que características dispone el servidor.

- Este panel incluye las herramientas para gestionar todo lo relacionado con el **HOST**.
- Nos permite **gestionar** además de todos los estados del servidor y sus aplicaciones.
- En este caso utilizaremos únicamente el paquete de **DHCP** para IP automática.



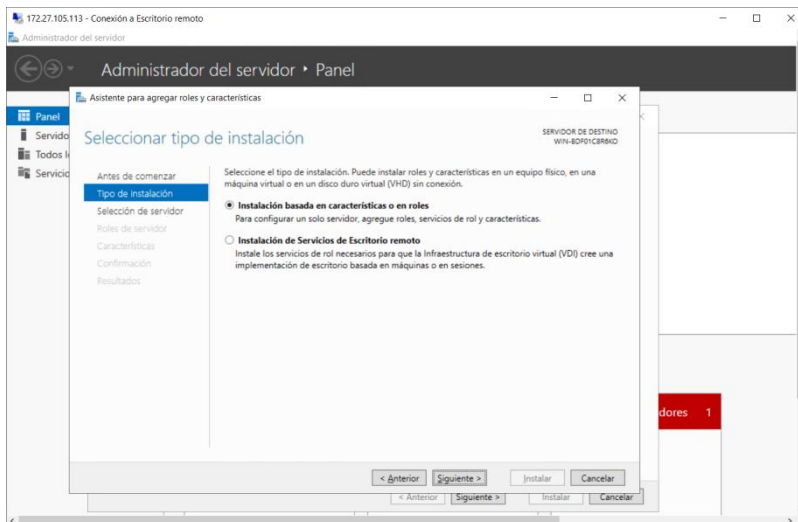
- d. Una vez estemos dentro, vamos a darle **permisos** de administrador.

- Para ello, debemos ir al apartado **Administración de equipos**, y desplegar el apartado **usuarios**.
- Ahora, debemos hacer click en el botón derecho de nuestro **mouse** y **definir** el **nombre** del administrador y su **contraseña**.

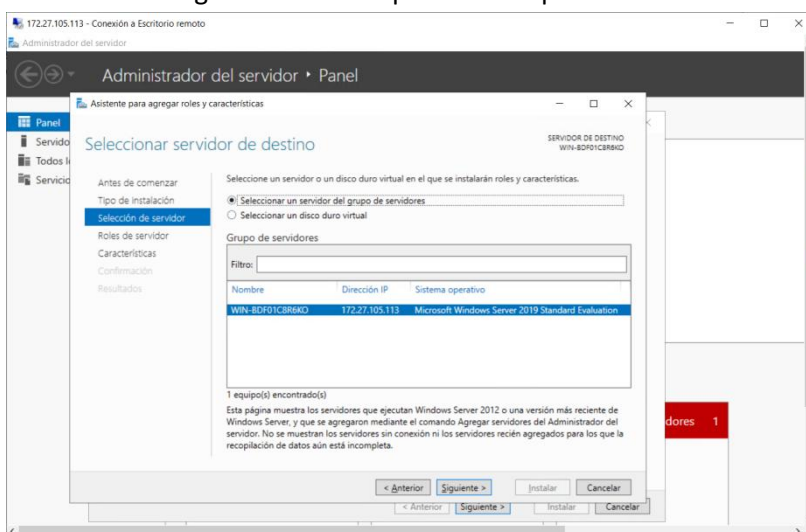


- e. Ahora, vamos a subirlo a un servidor. Para ello, debemos ir al apartado **Agregar roles y características**, que se encuentra en la parte superior de la derecha. Una vez hayas hecho ese paso, te aparecerá una ventana similar a la de la imagen.

- f. A continuación, hacemos click en siguiente (Viene por defecto). Pd: si os aparece marcada la segunda opción, debemos hacer escoger la **primera** opción.



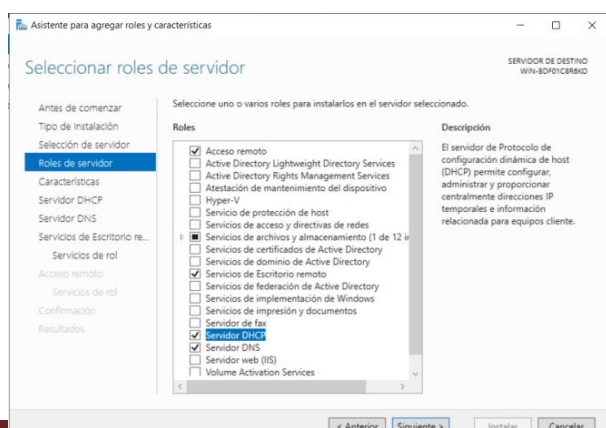
- g. Seleccionamos la opción un servidor del grupo de servidores y hacemos click en el en el botón siguiente que aparece en el asistente.



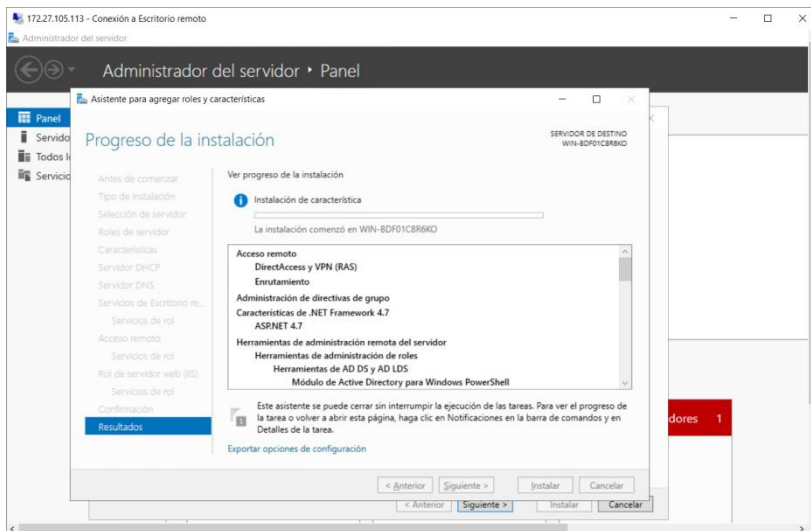
- h. Definimos los **roles** del servidor. Hay que tener **las mismas opciones** como está representado en la imagen. Además debemos instalar el **servidor DHCP** para poder gestionar las conexiones.

Haremos **click** en siguiente.

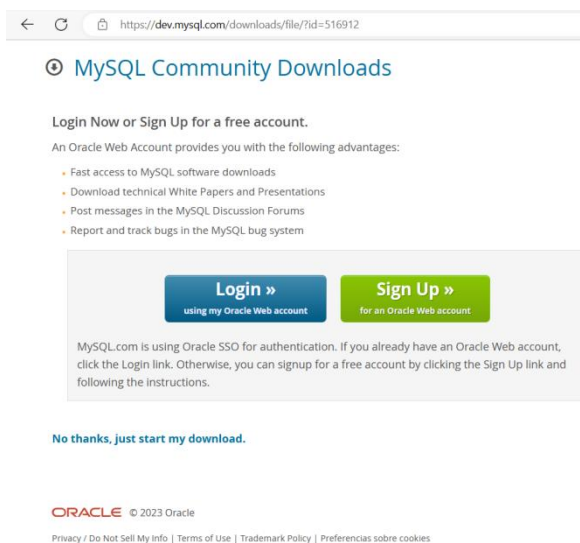
**NOTA: Este paso es únicamente si disponemos de un windows Server ya que en Windows 10 es innecesario ya que ya contiene configurado un servidor de este tipo.**



- i. Si todo ha ido bien, la maquina se reiniciará automáticamente para instalar las características del servidor. El resultado de todos los pasos mencionados anteriormente, se puede apreciar en la siguiente foto.



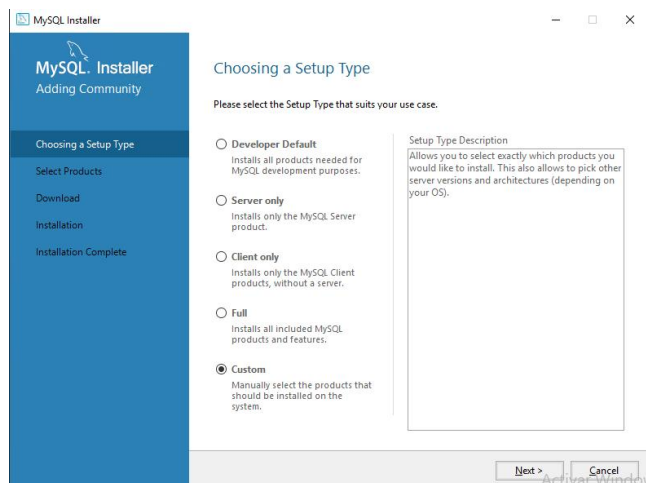
- j. Ahora vamos a instalar la Base de datos MySQL



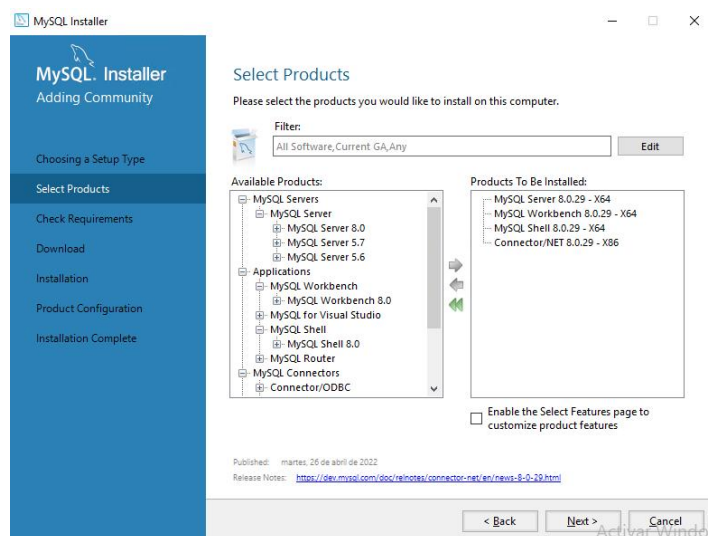
- Para ello, nos dirigimos a este enlace [MySQL](https://dev.mysql.com/downloads/file/?id=516912) donde tenemos preparado la descarga que vamos a procesar.
- En el caso de que en la descarga da error, vuelve a intentarlo haciendo click en [“No thanks, just download”](#).
- A continuación, comenzará a descargarse el archivo de instalación, cuya capacidad es de 45’7 MB, pero una vez instalado, su tamaño rondará 1’24 GB.

Iniciem l'instal·lador i marquem l'opció personalitzada. Aquesta opció ens permet escollir entre diferents paquets sense la necessitat de instal·lar totes les eines que ens proporciona MySQL de Oracle.

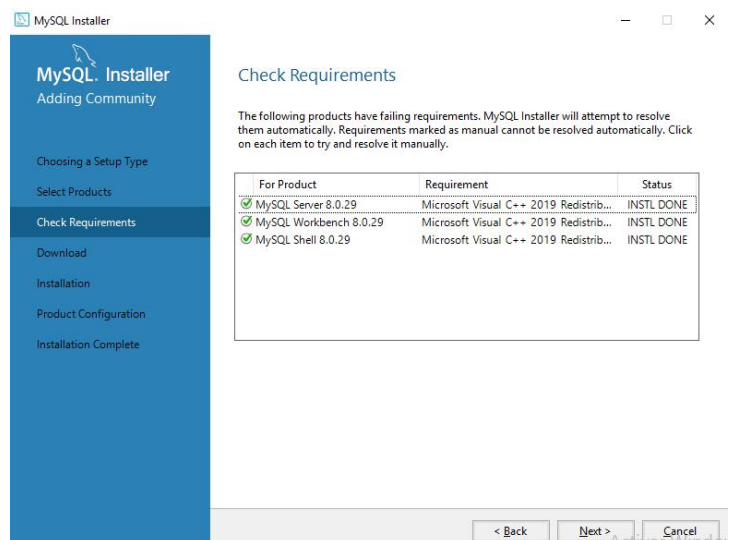
A més a més s'ha de fer menció que Forklite únicament requereix del servidor de base de dades i el connector, per tant es compatible amb MariaDB encara que s'instal·li MySQL.



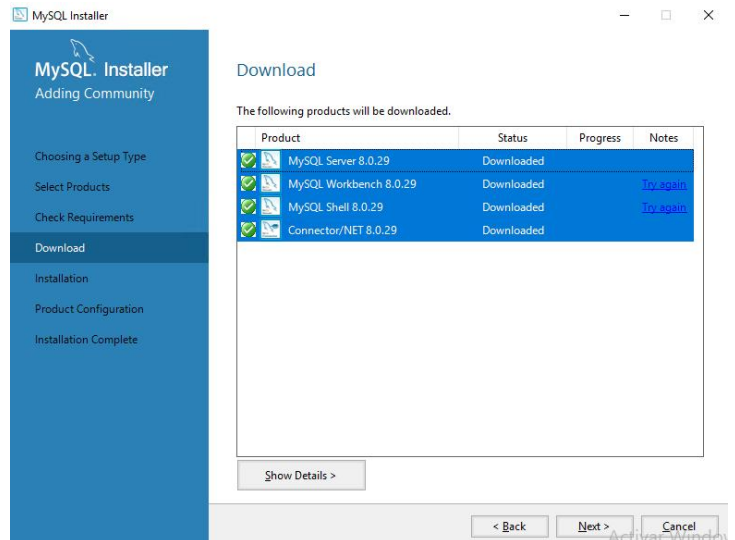
Per realitzar la instal·lació de la base de dades serà necessari obligatòriament els paquets de MySQL server i el connector de .NET però degut a que és disposa d'una base de dades prèviament s'afegirà el Workbench per poder restaurar el arxiu de base de dades .SQL



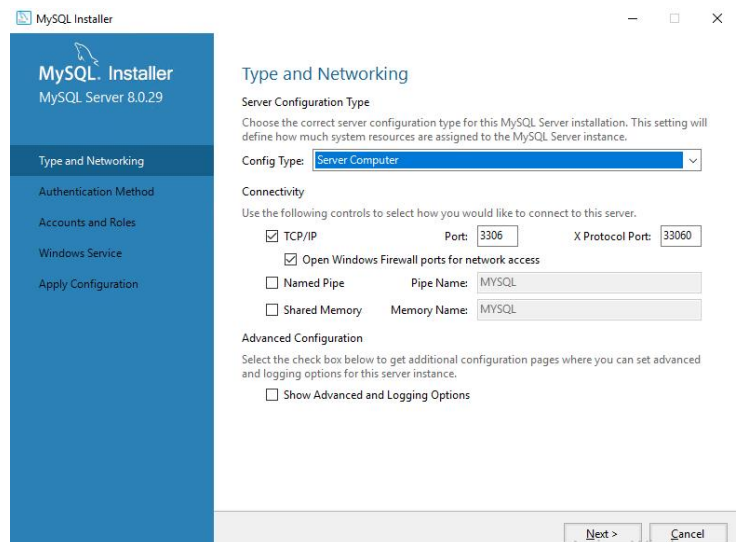
Premem Execute i Next. Durant el procés si no disposem de Visual C++ l'instal·lador afegirà els paquets a el nostre ordinador. La velocitat de descarrega pot influir en gran quantitat en la descompressió dels arxius.



Seguidament repetim el procés Execute Durant aquest pas l'aplicació pot fallar a l'hora de extraure els arxius. Premem **Try again** a els paquets que no continguin un senyal correcte. Un cop tots els paquets es troben validats premem Next

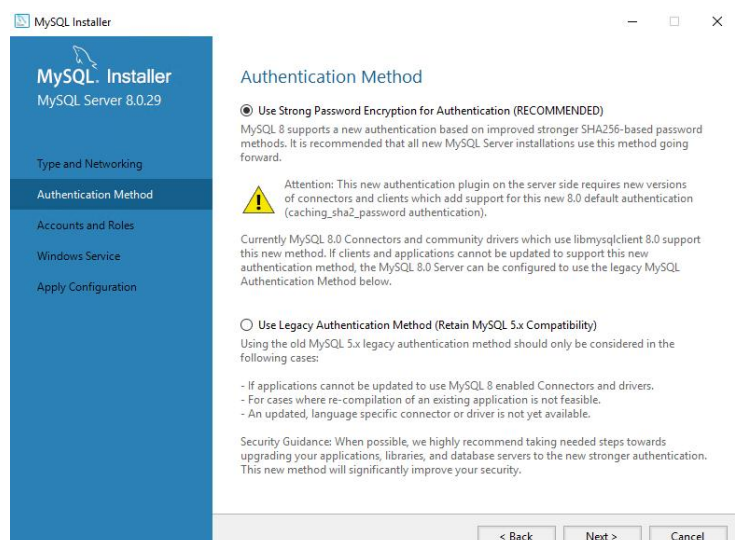


Seleccionem la opció Server Computer i activem la casella TCP/IP. A més a més indiquem el port 3306 i el port de protocol X 33060. Premem a Next i no modifiquem cap opció. Les opcions que Oracle ens proporciona per defecte al seu instal·lador ja ens serveixen.

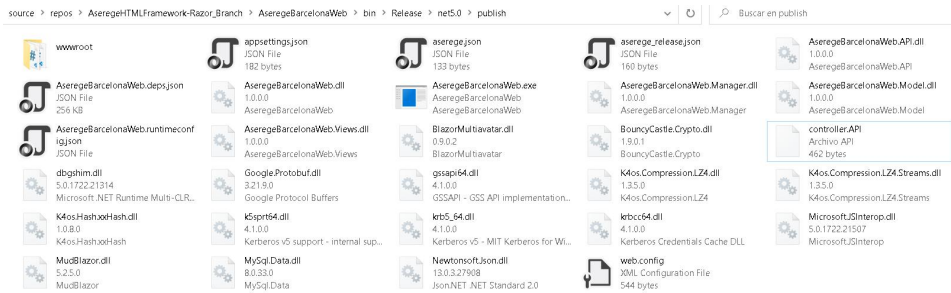


A continuació seleccionem la primera opció perquè ens proporciona un xifrat de contrasenyes més segur que la segona opció encara que si el teu dispositiu no es compatible no es necessari seleccionar la primera, encara que es recomana.

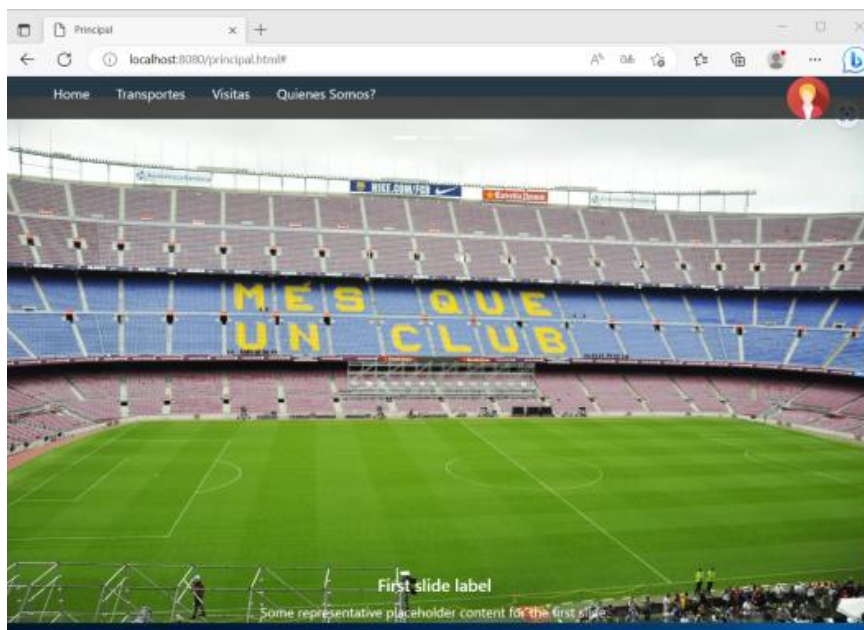
Premem Next per continuar

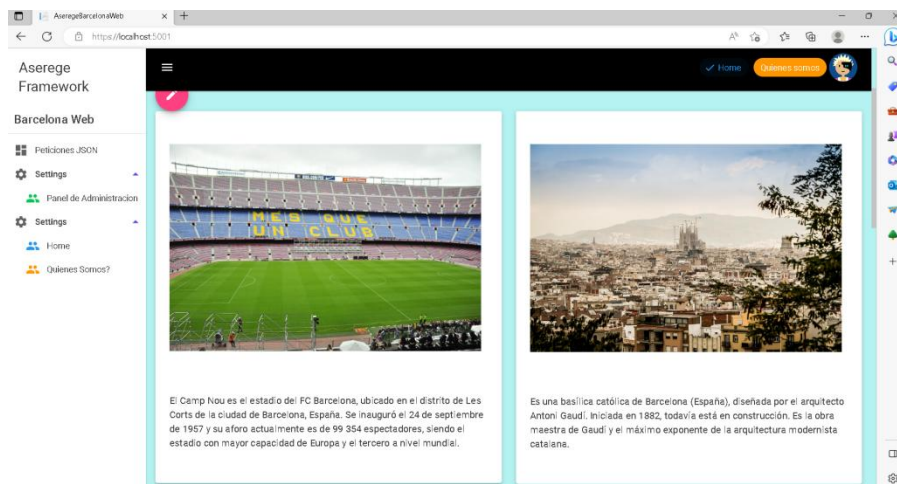






En esta imagen se muestra todas las bibliotecas incluidas en el programa desarrollado, incluyendo aquellas esenciales para garantizar la seguridad en el backend. También se han incorporado algunas extensiones para mejorar la funcionalidad del sitio web y los conectores API que permiten la conexión con la base de datos.





La ejecución de la aplicación producirá la imagen del portal principal de BarcelonaWeb. La página web seguirá mejorando constantemente para cumplir con los requisitos establecidos por el usuario. Se puede apreciar las imágenes el cambio de la web. En la primera imagen, se puede ver el anterior diseño, en el que la web no funcionaba aun. En la segunda foto, se puede apreciar su funcionalidad, dando como resultados información los contenidos que el usuario ha pedido.

## Publicar en servidor Linux



## 15. Configuración

Este JSON contiene todas las propiedades necesarias para hacer la conexión a la Base de datos de MySQL. Hay que tener en cuenta de que, si uno de las propiedades pones la configuración de manera errónea, automáticamente se cerrará la Base de datos, impidiendo conectarse a ella. A continuación, tendrás las explicaciones de cada propiedad, y un ejemplo del JSON.

- **"hostname"** indica el nombre del host donde se encuentra la base de datos, en este caso "localhost" indica que la base de datos está en el mismo servidor que la aplicación que se conectará a ella.
- **"port"** es el número del puerto donde se encuentra la base de datos. En este caso, el puerto es 3306, que es el puerto predeterminado para MySQL.
- **"username"** es el nombre de usuario para acceder a la base de datos. En este caso, es "root", que es el usuario predeterminado en MySQL con permisos completos.
- **"password"** es la contraseña para el usuario de la base de datos. En este caso, la contraseña es "root", pero en una configuración real, sería mejor utilizar una contraseña más segura y almacenarla de manera segura.
- **"databaseName"** indica el nombre de la base de datos a la que se quiere conectar. En este caso, es "barcelonaweb".

### Json de Configuración

```
{  
  "hostname": "localhost",  
  "port": 3306,  
  "username": "root",  
  "password": "root",  
  "databaseName": "freedb_barcelonaweb"  
}
```

## 16. Licencia y presupuesto

El framework Aserege tiene como licencia una EULA (End-User License Agreement) debido a que puedo hacer un mensaje conforme si el usuario acepta las condiciones para poder iniciar la instalación del software.

El documento sería el siguiente:

Aserege Framework Licencia de usuario final (EULA)

Versión 1.0, 2023

Este acuerdo de licencia de usuario final (en adelante, "Acuerdo") es un acuerdo legal entre usted (en adelante, "Licenciario") y Edgar Muñoz (en adelante, "Licenciante") para el software Aserege Framework (en adelante, "Software").

Al adquirir, instalar o utilizar el Software, el Licenciatario acepta los términos y condiciones de este Acuerdo. Si no acepta los términos y condiciones de este Acuerdo, no utilice ni instale el Software.

**Concesión de licencia** El Licenciante otorga al Licenciatario una licencia no exclusiva, intransferible y limitada para utilizar el Software en un solo dispositivo a la vez. La licencia es válida de manera permanente y no se puede transferir ni sublicenciar.

**Restricciones de uso** El Licenciatario no podrá modificar, descompilar, desensamblar ni realizar ingeniería inversa del Software. El Licenciatario no podrá revender ni transferir la licencia del Software a terceros sin la autorización previa y por escrito del Licenciante.

**Propiedad intelectual** El Software es propiedad exclusiva del Licenciante y está protegido por las leyes de propiedad intelectual. El Licenciatario reconoce que el Licenciante conserva todos los derechos de propiedad intelectual sobre el Software, incluidos los derechos de autor, patentes y marcas comerciales.

**Precio y pago** El Licenciatario pagará al Licenciante la cantidad de 2000€ para obtener la licencia completa del Software. Además, el Licenciatario pagará una tarifa cada tres meses para mantener la versión de demostración del Software.

**Garantía limitada** El Licenciante garantiza que el Software se ajusta a las especificaciones del Licenciante. El Licenciante no garantiza que el Software sea libre de errores o que funcionará sin interrupciones. El Licenciatario acepta que el uso del Software es bajo su propio riesgo.

**Limitación de responsabilidad** El Licenciante no será responsable de ningún daño especial, incidental, indirecto o consecuente que surja del uso o la imposibilidad de usar el Software, incluso si el Licenciante ha sido informado de la posibilidad de dichos daños.

**Término y terminación** Este Acuerdo entrará en vigor en la fecha de adquisición, instalación o uso del Software y continuará vigente hasta su terminación. El Licenciante podrá rescindir este Acuerdo de inmediato si el Licenciatario incumple alguna de las condiciones establecidas en este Acuerdo.

**Ley aplicable** Este Acuerdo se registrará e interpretará de acuerdo con las leyes del España, sin dar efecto a los principios de conflicto de leyes.

El presupuesto es de 2000€ debido a que se tiene en cuenta el tiempo de formación que se ha realizado durante los cursos de CFGM (Sistemas microinformáticos y Redes) y del CFGM (Desarrollo de Aplicaciones Web), las estadas en las empresas realizando las FCT (prácticas de empresa), el tiempo del desarrollo de la web y de todas las herramientas que se ha necesitado, puesto que hay que tener en cuenta que, dependiendo de cómo quieres hacer el FrontEnd y el BackEnd, además de la elección de la Base de Datos, tendrías que distribuir cada tarea para alcanzar el objetivo principal que te has propuesto.

Hay que tener en cuenta que la Base de datos de MySQL está ubicada en un servidor externo, por lo tanto, habrá que hacer conexiones remotamente.

## 17. Linias de mejora

BLBLABLA

## 18. Anexo: Código fuente

Para acceder al código fuente de la aplicación, se debe utilizar el enlace que se proporciona a continuación.

[Enlace](#)

## 19. Referencias

[Notion -Tu wiki, documentos y proyectos, todos juntos.] Notion Labs, inc , “*Notion.so*” Disponible en:

<https://notion.so>

(último acceso: Abril de 2023)

[MudAppBar API - MudBlazor] MudBlazor, “*AppBar*” Disponible en:

<https://www.mudblazor.com/api/appbar>

(último acceso: Abril de 2023)

[MudCarousel<string> API-MudBlazor] MudBlazor, “*Carousel*” Disponible en:

<https://www.mudblazor.com/api/carousel#pages>

(último acceso: Abril de 2023)

[Chips-MudBlazor] MudBlazor, “*Chips*” Disponible en:

<https://www.mudblazor.com/components/chips#api>

(último acceso: Abril de 2023)

[¿Qué es ASP NET Core?] Microsoft, “*ASP NET Core*” Disponible en:

<https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet-core>

(último acceso: Abril de 2023)

[MudTimeline API - MudBlazor] MudBlazor, “*Timeline*” Disponible en:

<https://www.mudblazor.com/api/timeline#properties>

(último acceso: Abril de 2023)

[MudFileUpload<images> API-MudBlazor] MudBlazor, “*FileUpload*” Disponible en:

<https://www.mudblazor.com/api/fileupload>

(último acceso: Abril de 2023)

[Webgrafía: qué es, características, cómo hacerla, ejemplos] Lifeder , “*Webgrafía*” Disponible en:

<https://www.lifeder.com/webgrafia/>

(último acceso: Abril de 2023)

[MudDialog API-MudBlazor] MudBlazor, “*MudDialog*” Disponible en:

<https://www.mudblazor.com/api/dialog>

(último acceso: Abril de 2023)