

## Sistema web – gestión de entrenamiento



### Descripción breve

La página web XXL será desarrollada y pensada para aquellas personas que tengan la necesidad de entrenarse con el asesoramiento de un profesional a través del sistema web. El usuario podrá ver su entrenamiento a seguir, junto con el profesional, ver sus avances o limitaciones. Además, el sistema gestionará la utilización de aparatos o maquinarias disponible tenga el Gimnasio.

Autores: Marcelo Peysa - Gustavo Godoy - Alfredo Nuñez - Gabriel Peysa



Grupo 24



(55) 1234-5678



correo@gymsistem.com.ar

# Contenido:

Pag 2 - Introducción  
Pag 3 - Esquema - Modelo relacional  
Pag 4 - Product Backlog - Sprint 0  
Pag 5 - Sprint 1  
Pag 6/10 - Breve explicación backend y frontend  
Pag 11/16 - Instalacion y configuracion pagina en angular  
Pag 17 - Referencia - Restricciones  
Pag 12 - Personal - Ficha del documento  
Pag 12 - Links explicación usuario final.

## Introducción

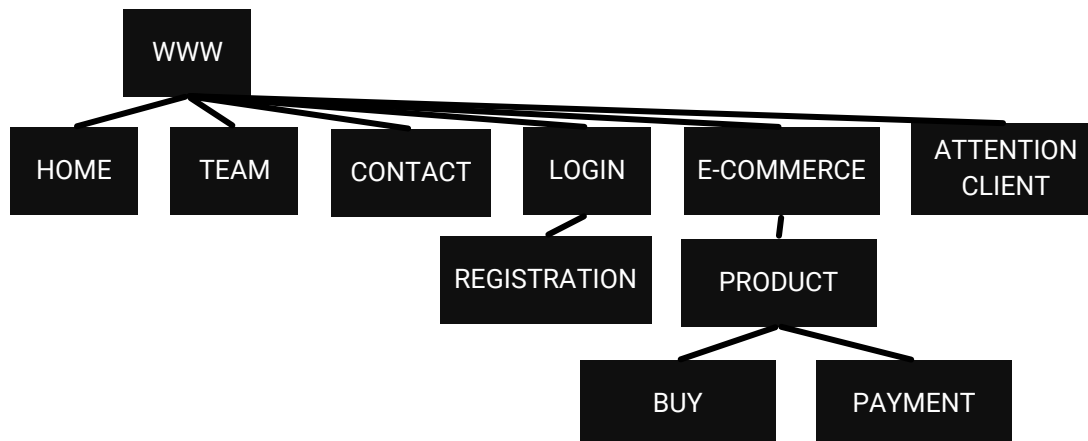
- Este documento tiene la finalidad de documentar un sistema web, realizado con HTML, JavaScript, Angular, CSS, etc. intentado explicar el funcionamiento del sistema y procesos de construcción. De esta manera facilitar a un usuario y al ISPC la navegación.
- El presente documento tiene como propósito definir las especificaciones funcionales, para el desarrollo de un sistema de información web que permitirá a los gimnasios llevar un control de la rutina de ejercicios de sus clientes como así también programar turnos asegurando acceso a las diferentes maquinas.
- El sistema va a permitir a los dueños de gimnasios generar diferentes usuarios y planes de entrenamientos para sus clientes organizando así los tiempos de rutinas de ejercicios. El sistema XLL va a ser desarrollado en una plataforma Web, esta aplicación va a permitir a los dueños de gimnasios administrar los planes de ejercicios de sus clientes y de las maquinas que van a usar en dichos planes, permitiendo una mejora en el tiempo de uso de cada máquina en los planes, teniendo el cliente siempre una maquina asegurada para su uso.

Además, el usuario (Cliente) podrá realizar su entrenamiento con un plan de ejercicio previamente cargado por un profesional, dependiendo solo del sistema y gestionando sus avance e historial desde la web.

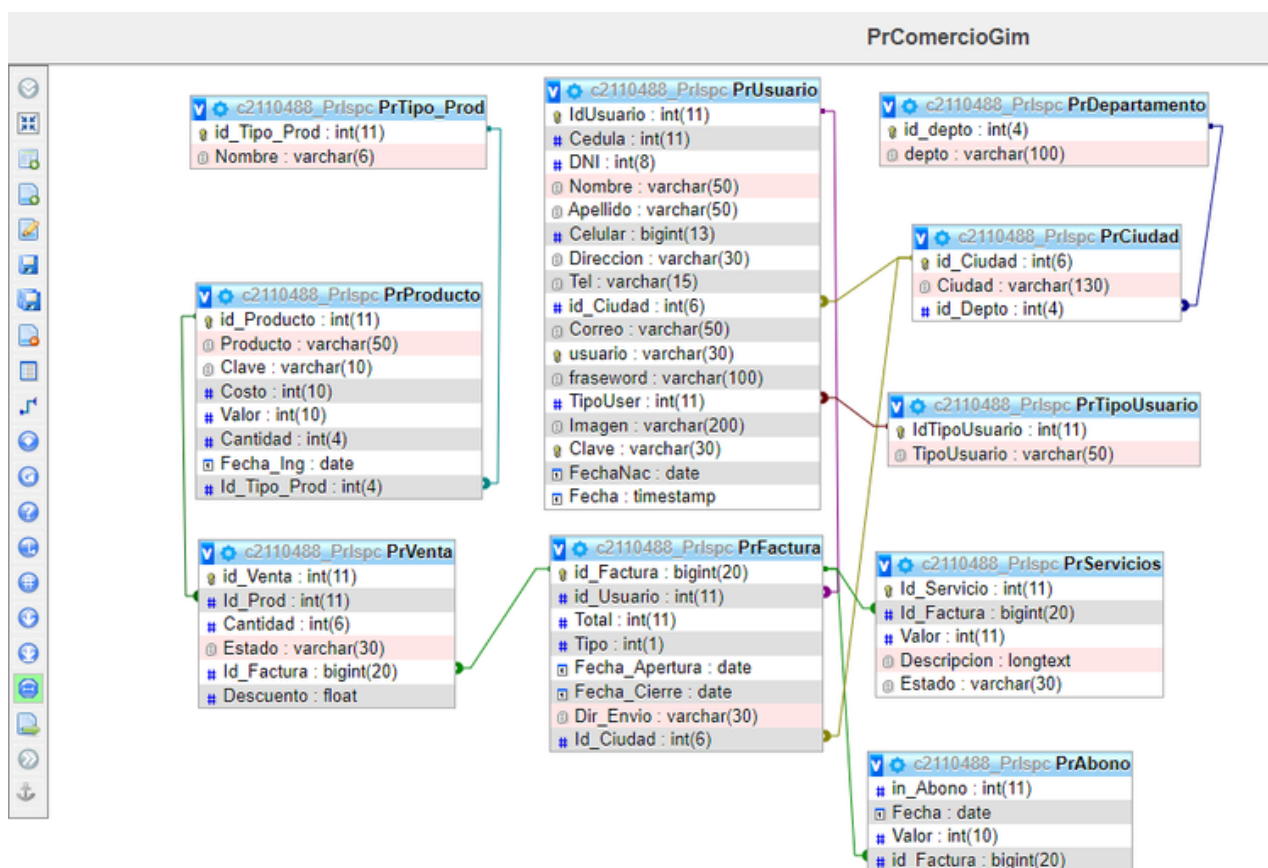
.



# Esquema



## Modelo Relacional



# Product Backlog 2021 - 2022

- Documento formato ieee-830...
- Control de paginas, menu y contenido
- Menú de navegación
- Formulario de Alta de Usuario.
- Diseño de logo de la aplicación XXL
- Base de datos
- Pagina Nosotros
- Página de contacto

## Sprint: Actividad 1 - Sprint 0

- Se les solicitará un proyecto aplicando todos los contenidos dados en los módulos de la cursada (HTML5, CSS3, BOOTSTRAP, JAVASCRIPT), pueden recurrir a la web de W3School para extraer recursos necesarios.

- Al proyecto desarrollado en 2021, la “consultora ISPC”, solicita en esta oportunidad, incorporar un módulo de e-commerce para comercializar sus productos y/o servicios online

### Antes de comenzar

- Definir Scrum Master inicial (No se registró en planilla) y registrarse todos dentro del nuevo repositorio. VER PLANILLA  
Importar el repositorio anterior al nuevo brindado en el foro de cada grupo.  
Revisar documentación IEEE830 si está completa y retomar desde allí para esta nueva etapa seguir documentando los avances.  
Revisar la estructura web Semántica y responsive (RWD)

Pasos a seguir de acuerdo a las Fases del Ciclo de Vida de un Software:

### ANÁLISIS

Definir requerimientos para el nuevo módulo a desarrollar e-commerce (colocarlos en el Product Backlog del Project), a su vez revisar si han cumplimentado todos los requerimientos previos, realizar mejoras  
Plantear Historias de Usuarios y Tareas dependientes de las US para incorporarlas en el repositorio remoto gitHub. (Issues y Milestones) - Tener en cuenta la redaccion adecuada para las US y nomenclatura, ej “#US01 Como usuario quiero ingresar al carrito para poder comprar” .



Definir tareas dentro de las Historias de Usuario (GITHUB) ej dentro de las ISSUES #TK01 importar repositorio.

### DISEÑO

Revisar la base de datos previa en MySQL si es funcional, e incorporar nuevas tablas necesarias para el funcionamiento del módulo e-commerce. (Ver el Modelo Relacional publicado en el libro Desafío E-commerce - Proyecto de trabajo Integrador

Crear su propio DER y Modelo relacional para documentar las tablas en la DB.

Crear un Diagrama de Clases y Casos de Uso para facilitar el modelado en POO.

### IMPLEMENTACIÓN

Convertir el index.html (su home) en una SPA de Angular con módulos y components.

## Sprint: Sprint 1

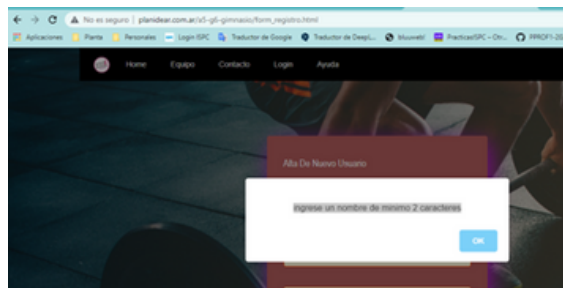
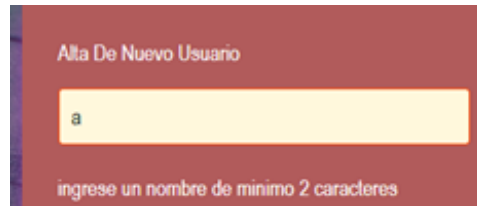


# Breve explicacion backend y frontend

1. Validación de campos vacíos mediante Bootstrap (de acuerdo al template de la web oficial), luego con Javascript la longitud de los campos, tipo de datos en los input. Los campos nombre y apellido deberán validarse con más de 2 caracteres.

Una de las validaciones están en el formulario registro de usuario: ([http://planidear.com.ar/a5-g6-gimnasio/form\\_registro.html](http://planidear.com.ar/a5-g6-gimnasio/form_registro.html))

El nombre y apellido luego de hacer un cambio en elemento con `onChange="validaNombre()"`



El código en el archivo funcionesGrupo6.js

```
function validaNombre(){
  if(document.getElementById("txtNombre").value.length <= 2 ){
document.getElementById ("AvNombre").innerHTML = "ingrese un nombre de minimo 2 caracteres" ;
    swal ( "ingrese un nombre de minimo 2 caracteres" ) ;
    return false;
  }else{
    document.getElementById ("AvApellido").innerHTML = "ok con el ingreso nombre" ;
  }
}

function validaApellido(){
  if(document.getElementById("txtApellido").value.length <= 2){
document.getElementById ("AvApellido").innerHTML = "ingrese un nombre de minimo 2 caracteres" ;
    swal ( "ingrese un Apellido de minimo 2 caracteres" ) ;
    return false;
  }else{
    document.getElementById ("AvApellido").innerHTML = "ok con el ingreso apellido" ;
  }
}
```

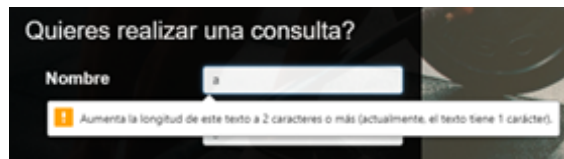


## En el formulario:

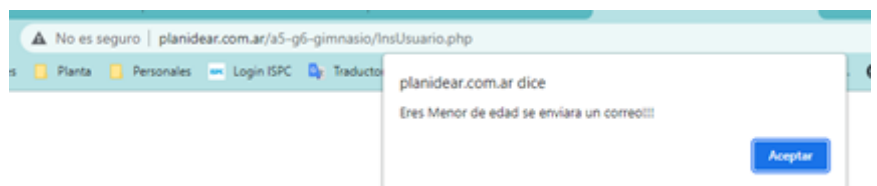
```
<input class="Controls" type="text" name="txtNombre" id="txtNombre" onChange="validaNombre()" maxlength="20"
placeholder="Ingrese su nombre">
<h4 id="AvNombre"></h4>
```

```
<input class="Controls" type="text" name="txtApellido" id="txtApellido" placeholder="Ingrese su Apellido"
onChange="validaApellido()" maxlength="20" required>
<h4 id="AvApellido"></h4>
```

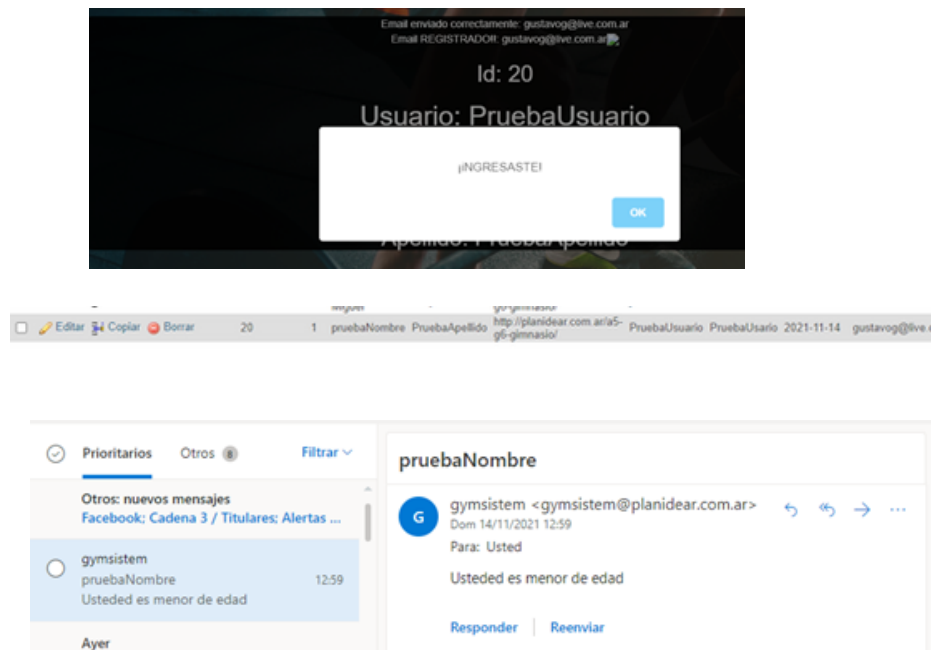
En un pequeño formulario en el index (<http://planidear.com.ar/g24-aula2-gimnasio-g24/index.html>) Se agrego restricciones con el html5:



- ☐ 2. Crear una función en JavaScript para mostrar un cálculo de fechas (edad, día de turno, u otro pertinente al proyecto en desarrollo).
- ☐ En el formulario registro de usuario: ([http://planidear.com.ar/g24-aula2-gimnasio-g24/form\\_registro.html](http://planidear.com.ar/g24-aula2-gimnasio-g24/form_registro.html))
- ☐ Al ingresar la fecha de nacimiento además de crear el registro y cargarlo a la base de datos, valida si el usuario es menor de edad. En caso de que sea menor de edad además de aparecer una alerta, se enviara un correo a su correo cargado:
- ☐

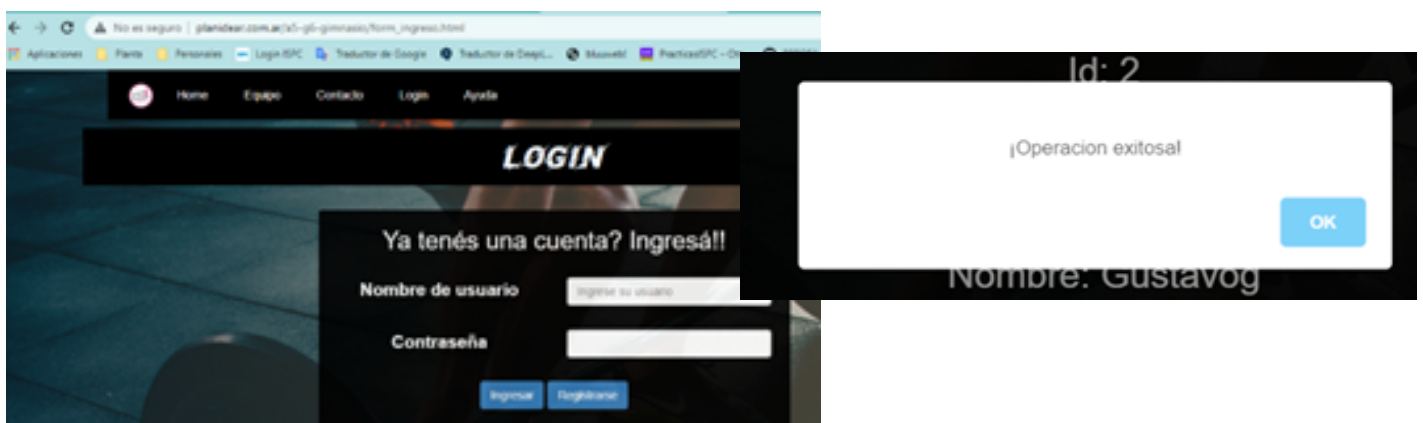


En el correo:



3. Uno de los formularios debe tener funcionalidad en el botón Enviar, mostrando un Alert de operación exitosa.

El formulario de login ([http://planidear.com.ar/g24-aula2-gimnasio-g24/form\\_ingreso.html](http://planidear.com.ar/g24-aula2-gimnasio-g24/form_ingreso.html)) después de hacer el post y validar los datos además de que no este vacío da la alerta ingresado con éxito:



4. Una vez procesado el formulario, mostrar en una pantalla siguiente, los datos procesados, la cual debe mantener la estética del sitio, luego de unos segundos, redireccionar a una página de sitio (por ejemplo al index).

En el ejemplo anterior, si el usuario no existe o esta vacío lo direcciona a la página login nuevamente:

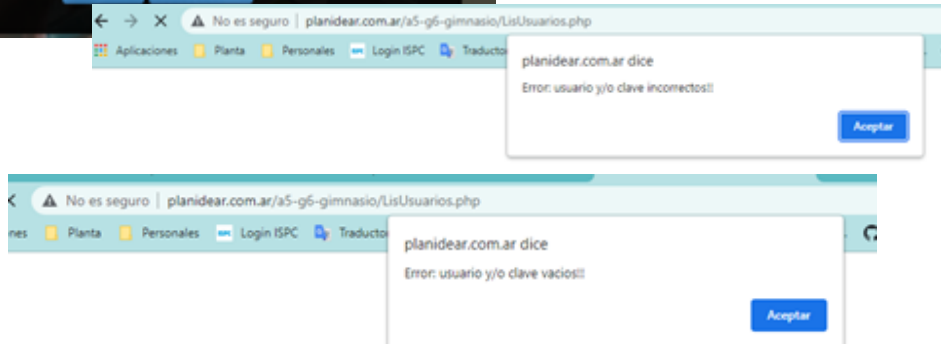




Ya tenés una cuenta? Ingresá!!

Nombre de usuario

Contraseña



5. Las funciones en JavaScript deberán estar en un archivo llamado funcionesGrupo99.js .  
Agregar al menos 2 eventos de Javascript para que el usuario interactúe con el DOM.

| Este equipo > Windows 10 (C:) > xampp >htdocs > planidear > a5-g6-gimnasio > js |                       |                     |        |  |
|---|-----------------------|---------------------|--------|--|
| Nombre  | Fecha de modificación | Tipo                | Tamaño |  |
| notes   | 14/11/2021 12:57      | Carpeta de archivos |        |  |
| Archivo.js  | 10/11/2021 13:56      | Archivo JavaScript  | 1 KB   |  |
| funcionesGrupo6.js  | 14/11/2021 12:57      | Archivo JavaScript  | 2 KB   |  |
| scriptContacto.js   | 10/11/2021 13:56      | Archivo JavaScript  | 2 KB   |  |
| scriptForm.js   | 10/11/2021 13:56      | Archivo JavaScript  | 1 KB   |  |

```
//funciones.js
//valida la edad después de enviar los datos desde
//http://planidear.com.ar/a5-g6-gimnasio/form_registro.html
function EresMayorEdad(){
    alert("Eres mayor de edad");
    //swal ( "Eres mayor de edad" ) ;
}
function EresMenorEdad(){
    alert("Eres Menor de edad se enviara un correo!!!");
    //swal ( "Eres Menor de edad se enviara un correo!!!" ) ;
}
//Esta función está en la página index.html
//cuando se pasa el mouse por encima del logo en el cuerpo de la pagina
//esta cambia de tamaño
function CambiarImagen(){
    document.getElementById("logo").style.height="200px"
    document.getElementById("logo").style.width="200px"
}
```



```

//las validaciones validar nombre y apellido lo hace cuando existe un cambio
//en el formulario registro, si tiene menos de los caracteres da un aviso, ademas cambia el DOM h4
function validaNombre(){
    if(document.getElementById("txtNombre").value.length <= 2 ){
document.getElementById ("AvNombre").innerHTML = "ingrese un nombre de minimo 2 caracteres" ;
        swal ( "ingrese un nombre de minimo 2 caracteres" ) ;
        return false;

    }else{
        document.getElementById ("AvNombre").innerHTML = "ok con el ingreso nombre" ;

    }

}

function validaApellido(){
    if(document.getElementById("txtApellido").value.length <= 2){
document.getElementById ("AvApellido").innerHTML = "ingrese un nombre de minimo 2 caracteres" ;
        swal ( "ingrese un Apellido de minimo 2 caracteres" ) ;
        return false;

    }else{
        document.getElementById ("AvApellido").innerHTML = "ok con el ingreso apellido" ;

    }

}

```



Nuestra página fue realizada con HTML, JS y PHP, ahora deberemos pasar la misma página en Angular, para luego hacer un sistema venta de productos.

Lo primero que debemos ver es que versión tenemos de node, luego si tenemos instalado angular Ej:

Para ver la version de node:

```
PS C:\xampp\htdocs\planidear\app-lspc> node -v  
v16.15.0
```

Para ver la version de angular:

```
npm -v  
PS C:\xampp\htdocs\planidear\app-lspc> npm -v  
8.12.2
```

En caso de no tener instalado angular, realizar la instalación con el siguiente comando en la terminal de Sistema:

☐ Instalar angular cli

npm install -g @angular/cli

☐ Crear una nueva aplicación:

ng new my-app, en este caso appgym

☐ Cambiar de carpeta:

☐ cd my-app

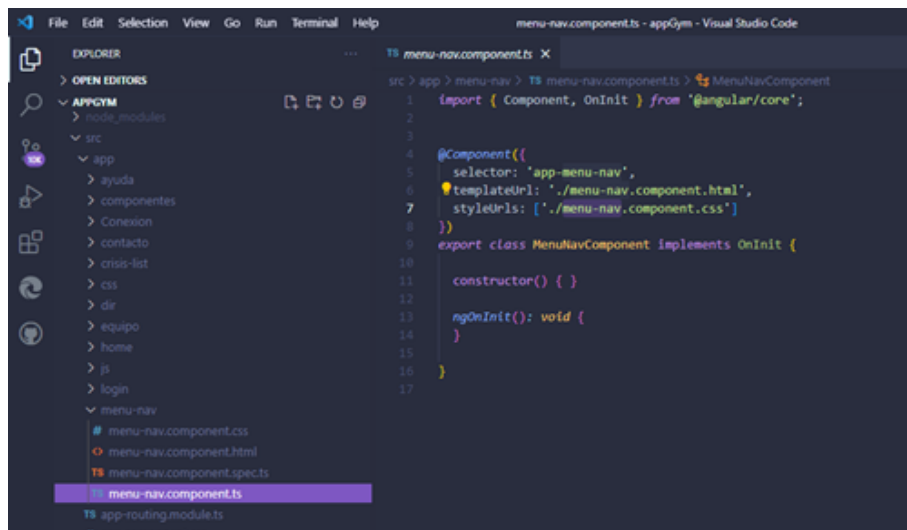
Abrir la aplicación:

ng serve - -open

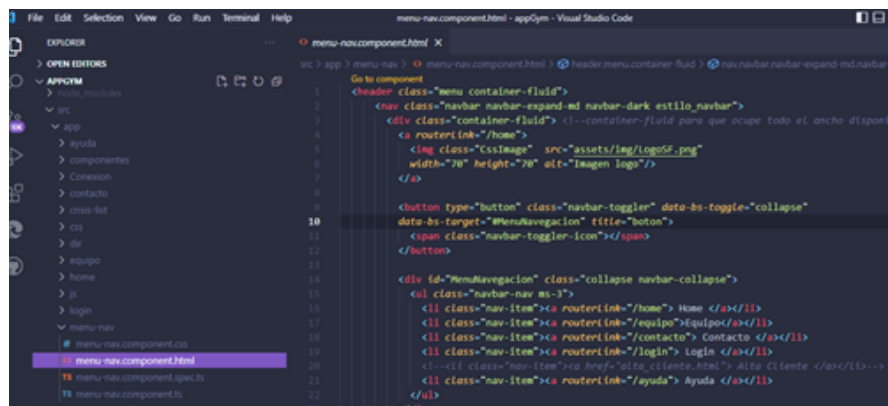
Para crear un nuevo componente, como por ejemplo el menu:

Debemos agregar en la terminal (dentro de nuestro app):

ng new menu-nav



En el archivo menu-nav.component.html, agregar el contenido html a mostrar

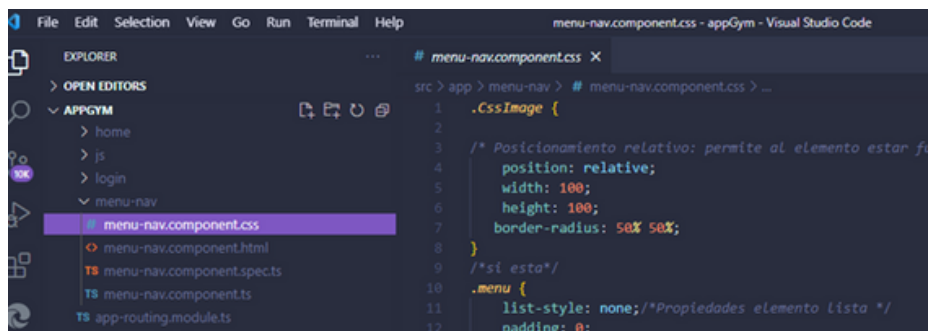


```
1 <header class="menu container-fluid">
2   <nav class="navbar navbar-expand-md navbar-dark estilo_navbar">
3     <div class="container-fluid"> <!--container-fluid para que ocupe todo el ancho disponible-->
4       <a routerLink="/home">
5         
7       </a>
8
9       <button type="button" class="navbar-toggler" data-bs-toggle="collapse"
10        data-bs-target="#MenuNavegacion" title="boton">
11         <span class="navbar-toggler-icon"></span>
12       </button>
13
14       <div id="MenuNavegacion" class="collapse navbar-collapse">
15         <ul class="navbar-nav ms-3">
16           <li class="nav-item"><a routerLink="/home"> Home </a></li>
17           <li class="nav-item"><a routerLink="/equipo">Equipo</a></li>
18           <li class="nav-item"><a routerLink="/contacto"> Contacto </a></li>
19           <li class="nav-item"><a routerLink="/login"> Login </a></li>
20           <li class="nav-item"><a href="alta_cliente.html"> Alta Cliente </a></li>
21           <li class="nav-item"><a routerLink="/ayuda"> Ayuda </a></li>
22         </ul>
23       </div>
24     </div>
25   </nav>
26 </header>
```

Para mostrar la imagen, primero debemos guardar la carpeta de imágenes dentro de assets:

`
2   <nav class="navbar navbar-expand-md navbar-dark estilo_navbar">
3     <div class="container-fluid"> <!--container-fluid para que ocupe todo el ancho disponible-->
4       <a routerLink="/home">
5         
7       </a>
8
9       <button type="button" class="navbar-toggler" data-bs-toggle="collapse"
10        data-bs-target="#MenuNavegacion" title="boton">
11         <span class="navbar-toggler-icon"></span>
12       </button>
13
14       <div id="MenuNavegacion" class="collapse navbar-collapse">
15         <ul class="navbar-nav ms-3">
16           <li class="nav-item"><a routerLink="/home"> Home </a></li>
17           <li class="nav-item"><a routerLink="/equipo">Equipo</a></li>
18           <li class="nav-item"><a routerLink="/contacto"> Contacto </a></li>
19           <li class="nav-item"><a routerLink="/login"> Login </a></li>
20           <li class="nav-item"><a href="alta_cliente.html"> Alta Cliente </a></li>
21           <li class="nav-item"><a routerLink="/ayuda"> Ayuda </a></li>
22         </ul>
23       </div>
24     </div>
25   </nav>
26 </header>
```

En menu-nav.component.css agregaremos el estilo de este componente menu-nav:



```
1 .CssImage {
2   position: relative;
3   width: 100px;
4   height: 100px;
5   border-radius: 50% 50%;
6 }
7
8 /*si esta*/
9 .menu {
10   list-style: none; /*Propiedades elemento Lista */
11   padding: 0;
12 }
```



Para trabajar con los enlaces de botones, ya no trabajamos con `<a href=""></a>`, sino que nos direccionamos a través de las rutas:

En el archivo `app-routing.module.ts`, dentro `src`, importar y agregar dentro de la constante `routes` las pistas de cada componentes. Los datos tomarlo del componentes terminado en `.ts`, así para poder navegar entre ellos:

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from '../home/home.component';
import { ContactoComponent } from '../contacto/contacto.component';
import { EquipoComponent } from '../equipo/equipo.component';
import { LoginComponent } from '../login/login.component';
import { AyudaComponent } from '../ayuda/ayuda.component';
import { MenuNavComponent } from '../menu-nav/menu-nav.component';

const routes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'contacto', component: ContactoComponent },
  { path: 'equipo', component: EquipoComponent },
  { path: 'login', component: LoginComponent },
  { path: 'ayuda', component: AyudaComponent },
];

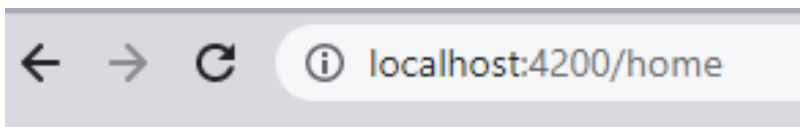
@NgModule({
  imports: [RouterModule.forRoot(routes)],
})
```

Dentro de la etiqueta de hipertexto agregar `routerLink="/comando">`:

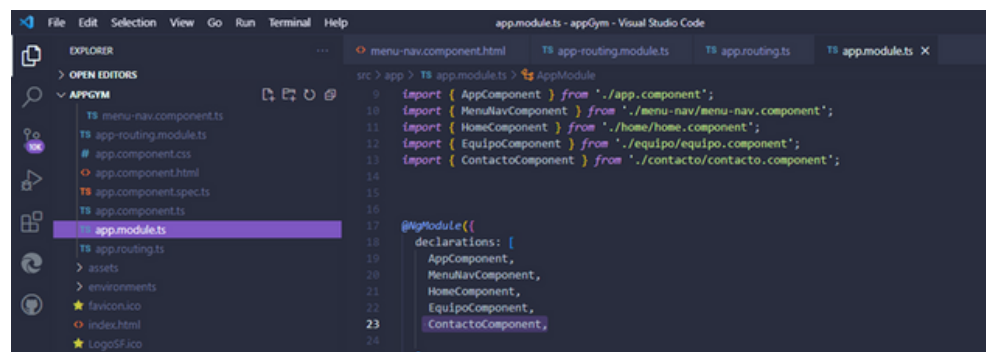
`<li class="nav-item"><a routerLink="/home"> Home </a></li>`

```
<li class="nav-item"><a routerLink="/equipo">Equipo</a></li>
<li class="nav-item"><a routerLink="/contacto"> Contacto </a></li>
<li class="nav-item"><a routerLink="/login"> Login </a></li>
<li class="nav-item"><a routerLink="/ayuda"> Ayuda </a></li>
```

De esta forma funcionaran los botones de navegación:



Para que se muestren los componentes/paginas, debemos primero importar y agregarla en el archivo `app.module.ts`:



Luego en el componente `app.component.html`, agregar app de menú-var:

Antes declarada en `app.module.ts`:

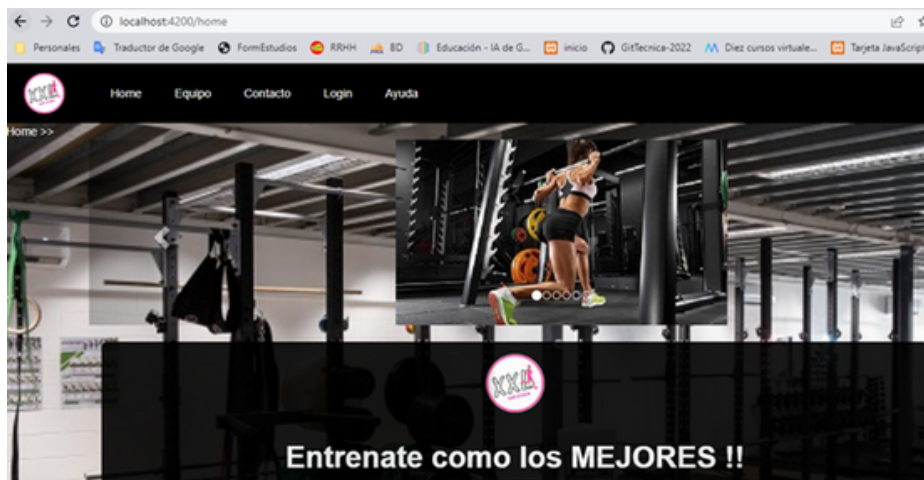
En `menu-nav.component.html` agregar `<router-outlet></router-outlet>` para que funcionen las rutas:

Para que la app ingrese con contenido, se agrego en el body `<app-root></app-root>` y código html:





La aplicación se ve de este modo:



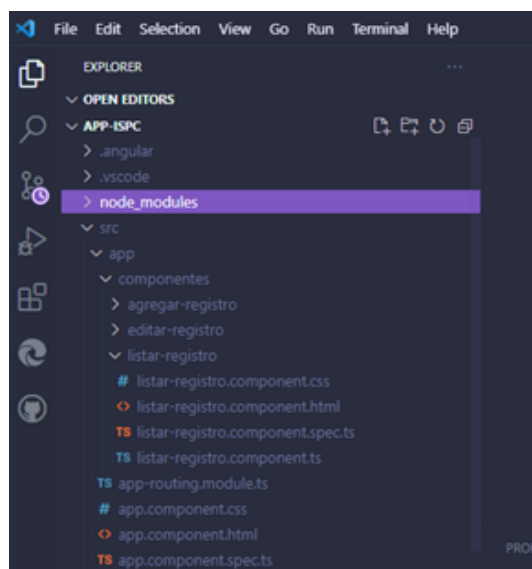
Consejos y atajos:

- Para crear la carpeta de componente y un componente nuevo:  
ng g c componentes/nuevoComponente

○

○

○



○

○

○

○

Para instalar bootstrap

Detener el servidor antes

npm install bootstrap

Para crear un menú rápido:

bs5-nav-minimal-a

```
<nav class="navbar navbar-expand navbar-light bg-light">
  <div class="nav navbar-nav">
    <a class="nav-item nav-link active" href="#">Home <span class="visually-
hidden">(current)</span></a>
    <a class="nav-item nav-link" href="#">Home</a>
  </div>
</nav>
```

○ ○ ○ ○



Grupo 24

(55) 1234-5678

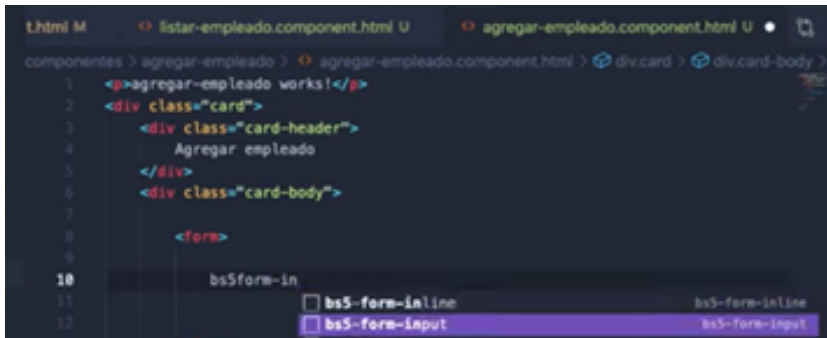
correo@gymsistem.com.ar

Modificar el script de Archivo angular.json

```
],  
  "styles": [  
    "src/styles.css",  
    "node_modules/bootstrap/dist/css/bootstrap.min.css"  
  ],
```

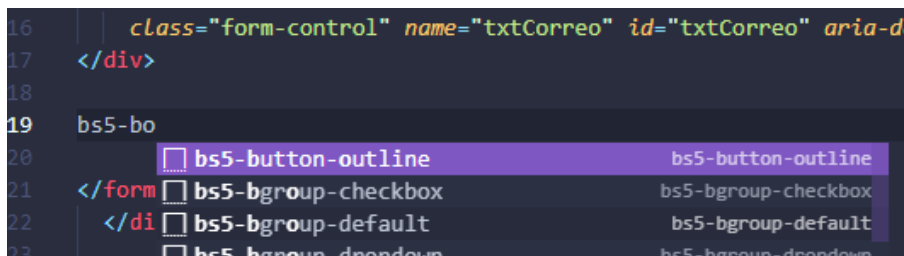
Crear una tabla rápido con bootstrap bs5-table-default

Crear un formulario rápido con bootstrap Bs5-form-input



```
1 <p>agregar-empleado works!</p>  
2 <div class="card">  
3   <div class="card-header">  
4     Agregar empleado  
5   </div>  
6   <div class="card-body">  
7  
8     <form>  
9  
10      bs5form-in  
11      bs5-form-inline  
12      bs5-form-input
```

Boton con bootstrap bs5-button-outline



```
16   class="form-control" name="txtCorreo" id="txtCorreo" aria-d  
17 </div>  
18  
19 bs5-bo  
20   bs5-button-outline  
21 </form> bs5-bgroup-checkbox  
22 </di bs5-bgroup-default  
23 bs5-bgroup-dropdown
```



## Referencia:

| Titulo del Documento     | Referencia                                                                                                                |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Standard IEEE 830 - 1998 | IEEE                                                                                                                      |
| Fuentes de Google        | <a href="https://fonts.google.com/">https://fonts.google.com/</a>                                                         |
| Estilo Bootstrap         | <a href="https://getbootstrap.com/">https://getbootstrap.com/</a>                                                         |
| Página de Ayuda          | <a href="https://www.w3schools.com/">https://www.w3schools.com/</a>                                                       |
| Guía Git                 | <a href="https://rogerdudler.github.io/git-guide/index.es.html">https://rogerdudler.github.io/git-guide/index.es.html</a> |
| Canva                    | <a href="https://www.canva.com/">https://www.canva.com/</a>                                                               |

| Links               | Referencia                                                                                                                              |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| GitHub              | <a href="https://github.com/PPROF2-2022ProgWeb/g24-aula2-gimnasio-g24">https://github.com/PPROF2-2022ProgWeb/g24-aula2-gimnasio-g24</a> |
| Pagina prueba       | <a href="http://planidear.com.ar/g24-aula2-gimnasio-g24/index.html">http://planidear.com.ar/g24-aula2-gimnasio-g24/index.html</a>       |
| Explicacion usuario | <a href="http://planidear.com.ar/g24-aula2-gimnasio-g24/help.html">http://planidear.com.ar/g24-aula2-gimnasio-g24/help.html</a>         |

## Restricciones:

- Interfaz para ser usada con internet
- Lenguajes y tecnologías en uso: HTML, JavaScript, angular, typescript, php, ect.
- Base de datos Mysql
- Bootstrap
- Navegadores Web.
- Alojamiento web y base de datos.



# Personal Involucrado:

| Nombre        | Responsabilidad - Rol                    | e-mail                     |
|---------------|------------------------------------------|----------------------------|
| Marcelo Peysa | Desarrollo y actualizacion GitHub        | mpeysa@gmail.com.ar        |
| Gustavo Godoy | Desarrollo, base de datos y informe IEEE | grgodoy1984@gmail.com      |
| Alfredo Nuñez | Diseño y Desarrollo.                     | alfredonunez0688@gmail.com |
| Gabriel Peysa | Scrum Master - Diseño y Desarrollo.      | gabriel.peysa@gmail.com    |

## Ficha del documento

| Año  | Revisión    | Autores                                                                                                                    |
|------|-------------|----------------------------------------------------------------------------------------------------------------------------|
| 2021 | Version 1.0 | Marcelo Peysa - Gabriel Rodenas - Cristian Martin Herrera - Gustavo Godoy - Alfredo Nuñez - Gabriel Peysa - Roger Ferreyra |
| 2022 | Version 2.0 | Marcelo Peysa - Gustavo Godoy - Alfredo Nuñez - Gabriel Peysa - Roger Ferreyra                                             |

