

Ecommerce

Sistema web de entrenamiento



Descripción breve

En un principio la página web XXL Gym System fue pensada para implementación de turno y ejercicios a través de la web.

En la segunda etapa del año 2022 se concentró en el ecommerce de productos relacionados con el deporte.

El mismo se podrá gestionar por un usuario o por un administrador.

Autores: Gustavo Godoy



Contenido:

- Pag 1 - Descripción breve
- Pag 2 - Contenidos - Introducción
- Pag 3 - Sprint 0
- Pag 4 - Sprint 1
- Pag 5 - Sprint 2
- Pag 6 - Sprint 2 + 3
- Pag 7 - Sprint 3
- Pag 8 - Sprint 3
- Pag 9 - Sprint 3
- Pag 10 - Sprint: 4
- ag 11 - The scrum framework / El marco de Scrum
- Pag 12 - Referencia

Introducción

- Este documento tiene la finalidad de documentar un sistema web ecommerce, realizado en Fronted con HTML, JavaScript, Angular, CSS, con Bakend utilizando Spring Tool y Php con base de datos phpMyAdmin.
- En esta etapa, el sistema web, podrá auto generarse un usuario, editarse y realizar alguna compra, de acuerdo a los productos en stock.
- También tendrá la posibilidad de ser administrado los usuarios y productos por un usuario con el rol de administrados.



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.

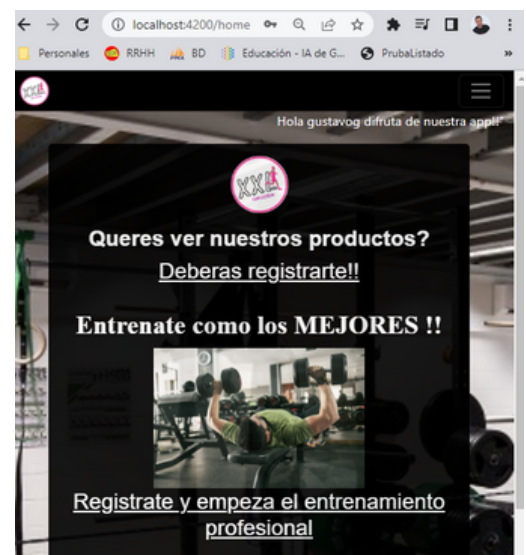
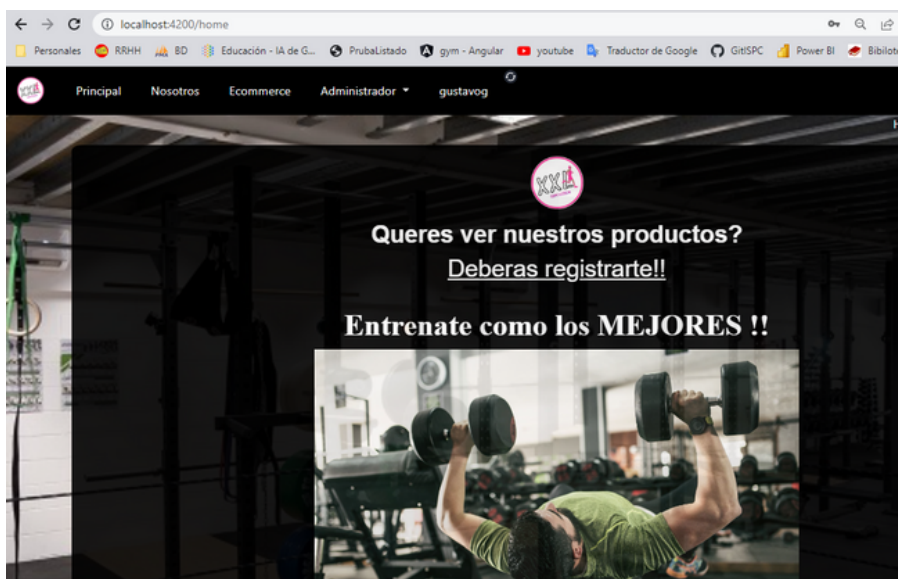
Al proyecto desarrollado en 2021, la “consultora ISPC”, solicita en esta oportunidad, incorporar un módulo de e-commerce para comercializar sus productos y/o servicios online Antes de comenzar

- 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[1] 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, realizando mejoras del mismo
- ● Plantear Historias de Usuarios y Tareas dependientes de las US para incorporarlas en el repositorio remoto gitHub. (Issues y Milestones) - Tener en cuenta la redacción 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.



Grupo 24



(353) 154191741



grgodoy1984@gmail.com

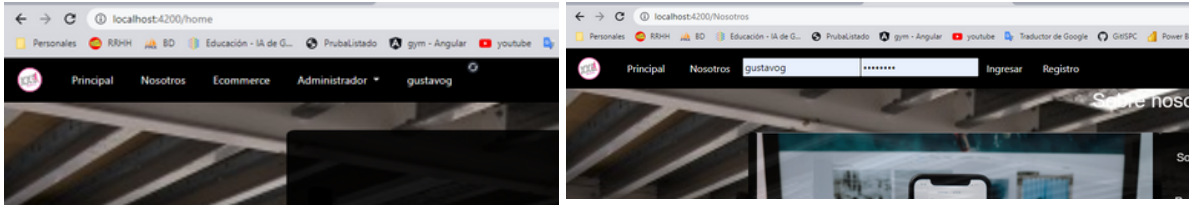
Sprint: Sprint 1

Fecha Inicio = 27/06/2022 - Fecha de Fin = 08/08/2022*

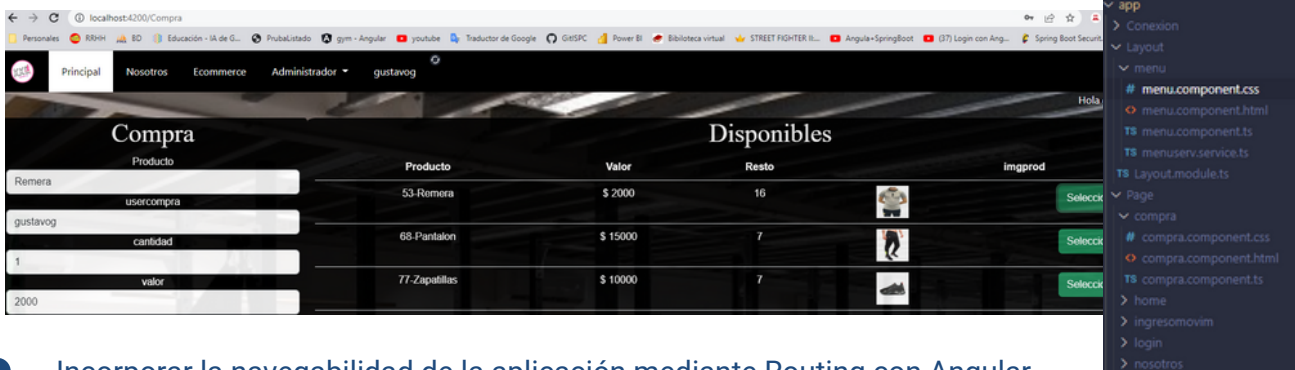
*se considera el receso invernal (09/07/2022 al 24/07/2022)

IMPLEMENTACIÓN FRONTEND

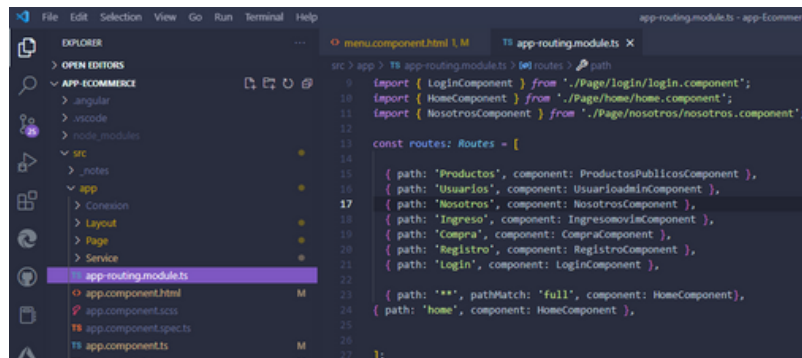
- Convertir los archivos .html del proyecto previo, en una SPA de Angular con módulos (Layouts y Pages por ejemplo) con sus components correspondientes. (HOME, REGISTRO, LOGIN, DASHBOARD)



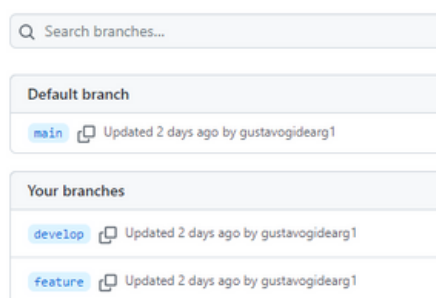
- Crear módulos y componentes para la tienda virtual o ecommerce. (PRODUCTO o SERVICIOS, PRODUCTO INDIVIDUAL, COMPRA).



- Incorporar la navegabilidad de la aplicación mediante Routing con Angular.



- Subirla al repo grupal, con GIT , en una branch por desarrollador para que cada uno tenga una copia. Luego crear una branch feature[1], para realizar nuestros cambios que no se encuentran aún en condiciones de incorporarse a la branch develop[2].

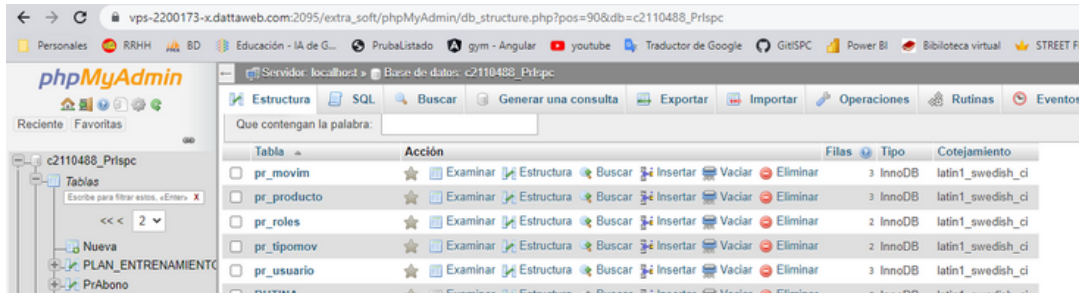


Sprint 2

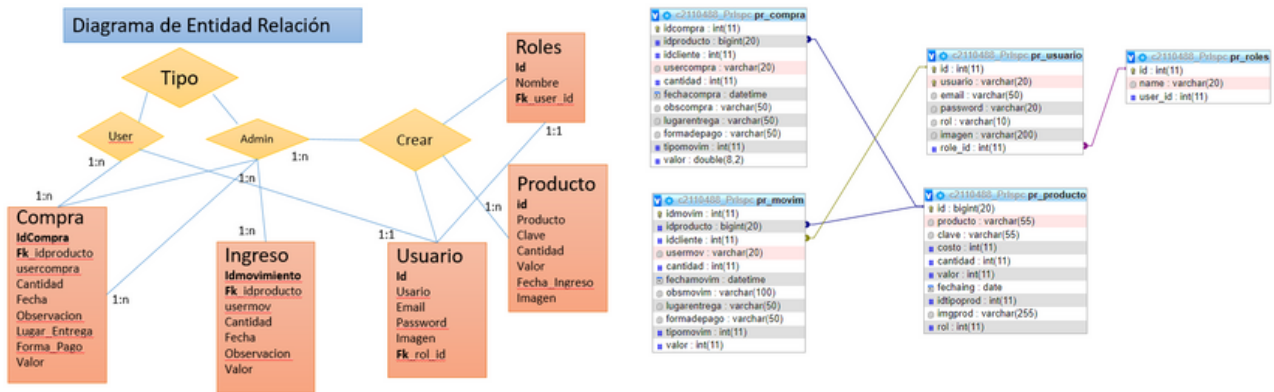
Fecha Inicio = 15/08/2022 - Fecha de Fin = 29/08/2022

DISEÑO DE BASE DE DATOS

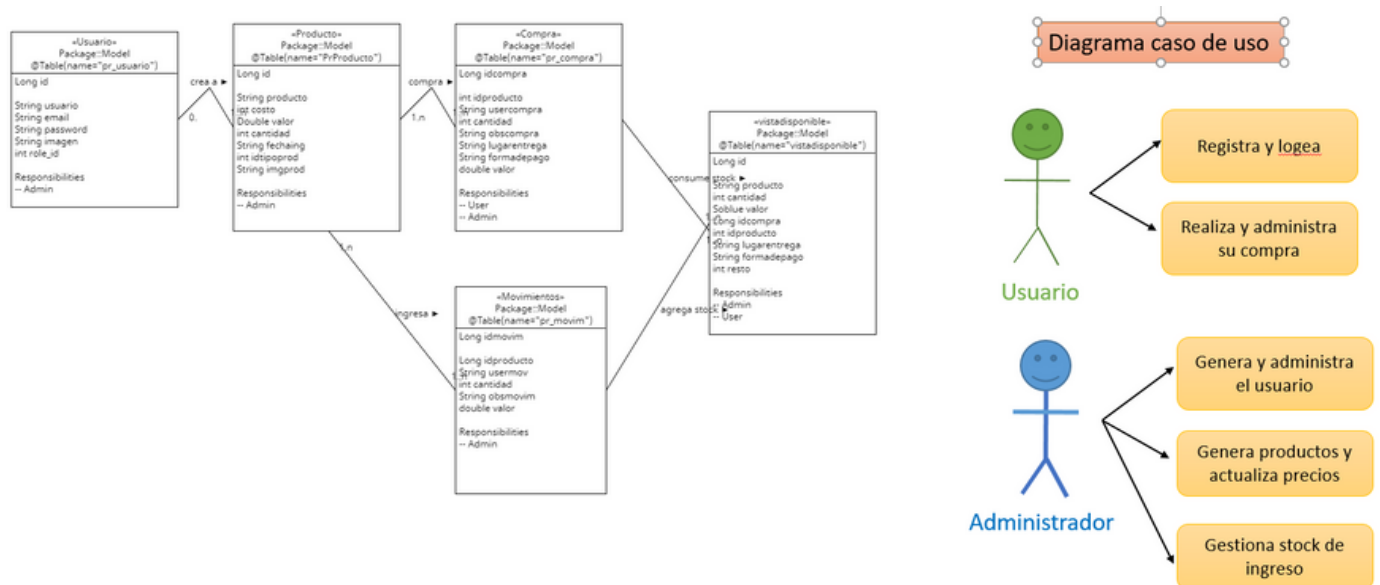
- Revisar la base de datos previa en MySQL si es funcional, e incorporar nuevas tablas necesarias para el funcionamiento del módulo e-commerce. (Ver el Modelo Relacional publicado en el libro Desafío E-commerce - Proyecto de trabajo Integrador)



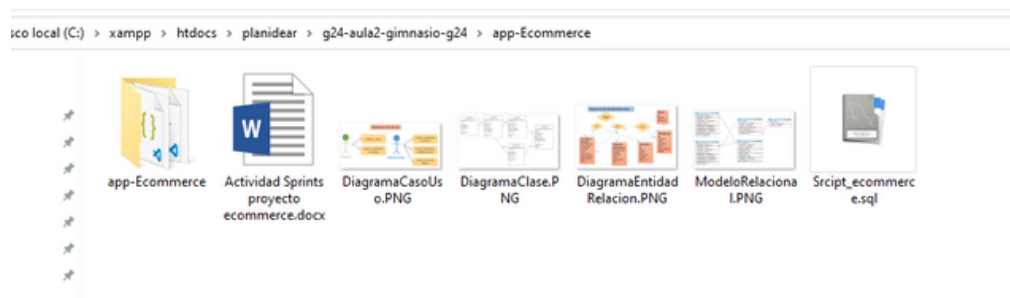
- Crear su propio DER y Modelo relacional para documentar las tablas en la DB.



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



- Script sql de la base de datos actualizada con las tablas nuevas.



- Actualizar el repositorio grupal, creando una carpeta dentro de la branch

Sprint 3

Fecha Inicio = 29/09/2022 -Fecha de Fin = 17/10/2022

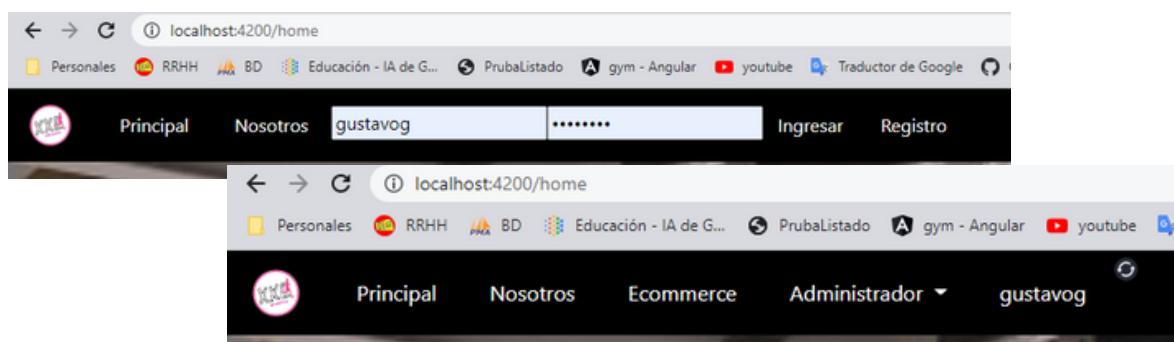
Requerimientos en Incremento al producto:

- Definición del backend y funcionalidad en login y registro.

- Registro de usuarios

- Login de usuarios

- Una vez logeado desaparece los campos de ingresar y registrarse, para aparecer el botón de usuario, y en caso que sea admin, se mostrara el botón para administrar productos y usuarios.



- Tablero de control (Gestión) o Dashboard con módulos planteados (Maquetación Front)

Los formularios deberán tener estilos semánticos que provee Bootstrap por ejemplo:

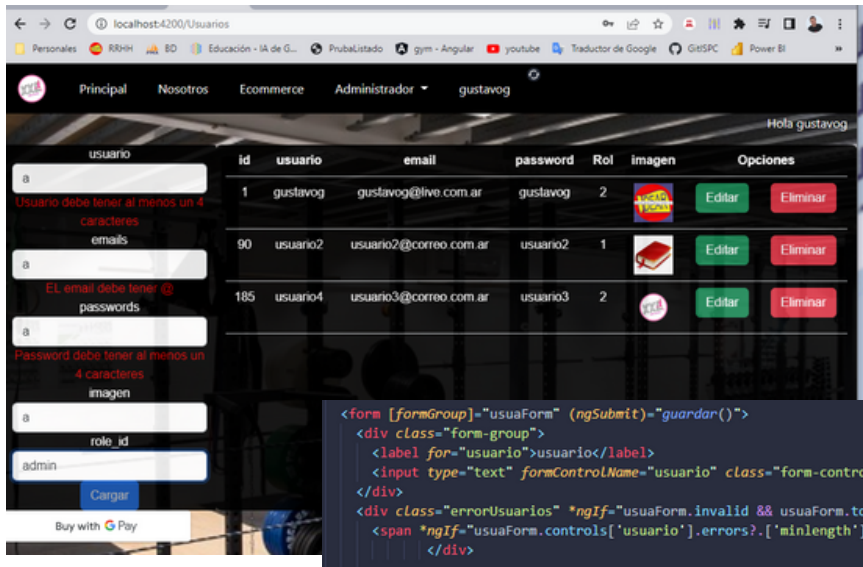
```
<span class="border border-success"></span>
```

```
<span class="border border-danger"></span>
```

Validaciones en TypeScript . Ejemplo: campos requeridos, formato de datos (mail, longitud cadena, numérico). Emplear los input type adecuados en cada campo (type=number, type=email, type=date, etc).

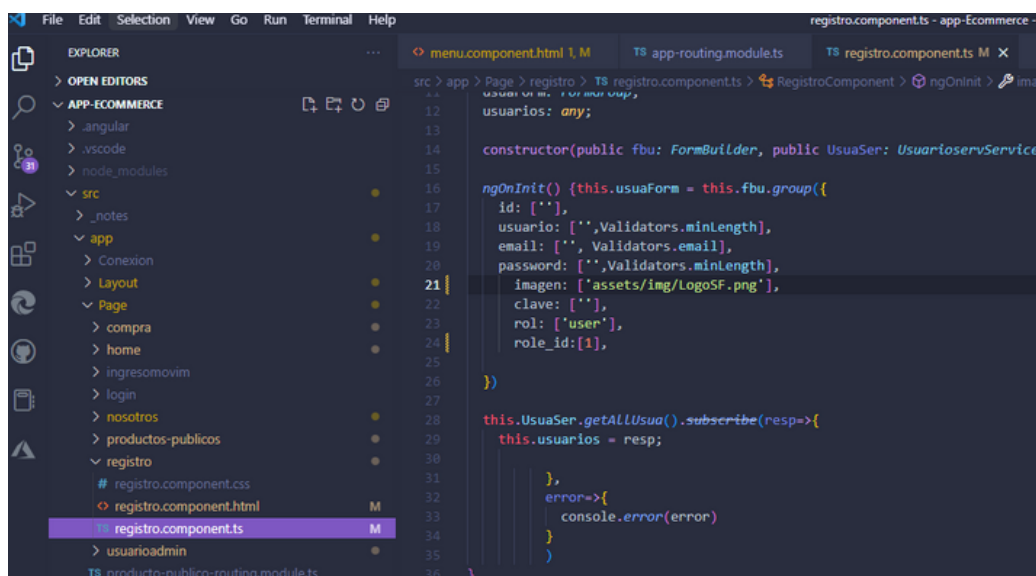
<https://developer.mozilla.org/es/docs/Web/HTML/Element/input>

- Ejemplo en la pagina y en el codigo de HTML en angular:



```
<form [formGroup]="usuaForm" (ngSubmit)="guardar()">
  <div class="form-group">
    <label for="usuario">usuario</label>
    <input type="text" formControlName="usuario" class="form-control" id="usuario" name="usuario" minlength="4" required >
  </div>
  <div class="errorUsuarios" *ngIf="usuaForm.invalid && usuaForm.touched">
    <span *ngIf="usuaForm.controls['usuario'].errors['minlength']>Usuario debe tener al menos un 4 caracteres</span><br>
  </div>
</form>
```

- Para acceder al formulario de administrador usuario, solo puede hacer los usuarios Admin. El usuario normal que se registra solo, por defecto se genera con el rol user. Esto viene desde el código de TypeScript:



● OPCIONAL: Token validación del login con JWT.

- El logeo y autenticación no lo llegue a realizar con Java ya que me falta aprender y copiarlo desde otro sitio no me pareció correcto. Si lo realice con backend de Php, solo para el logeo. Todo lo demás se realizó con Spring Tool:

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a package named `com.usuarios` with sub-packages like `Conexion`, `Conexion.Repository`, and `Conexion.RepositoryImpl`. The code editor shows a Java class `Usuario` with fields `email`, `password`, `rol`, and `role_id`. The `role_id` field is highlighted.

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a package named `com.usuarios` with sub-packages like `Conexion`, `Conexion.Repository`, and `Conexion.RepositoryImpl`. The code editor shows a TypeScript file `compraserv.service.ts` with imports for `HttpClient`, `Injectable`, `Observable`, and `Movimientos`. The file contains a class `CompraservService` with methods `getALLMov()`, `getALLMovDisp()`, and `saveMov`.

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a package named `com.usuarios` with sub-packages like `Conexion`, `Conexion.Repository`, and `Conexion.RepositoryImpl`. The code editor shows a PHP file `conexion.php` with a function `require` and a class `Result`.



Sprint 4

Fecha Inicio = 17/10/2022 -Fecha de Fin = 16/11/2022

Requerimientos

Incremento del producto:

Funcionalidades completas (front + back integrado)

- Validaciones con Angular: servicios - inyección de dependencias.

- La pagina tiene validaciones importadas de angular:

`import { FormBuilder, FormGroup, Validators } from '@angular/forms';`

```
ngOnInit() {this.usuaForm = this.fbu.group({
  id: [''],
  usuario: ['',Validators.minLength],
  email: ['', Validators.email],
  password: ['',Validators.minLength],
  imagen: ['assets/img/LogoSF.png'],
  clave: [''],
  rol: ['user'],
  role_id:[1],
});
```

usuario

a

Usuario debe tener al menos un 4 caracteres

emails

a

EL email debe tener @

passwords

- Comunicación entre componentes a través de servicios. En este caso el usuario logeado del componente Menú a las demás componentes. Si no está logeado, no se puede acceder por ejemplo a compras, si no está logeado como Admin, no puede acceder a Productos o Usuarios.

```
APP-E-COMMERCE
src
  app
    Conexion
    Layout
      menu
        menu.component.css
        menu.component.html
        menu.component.ts
        menu.service.ts
        layout.module.ts
    Page
      compra
      home
      ingresomovim
      login
      nosotros
      productos-publicos
      registro
        registro.component.css
        registro.component.html
        registro.component.ts
```

```
@Injectable({
  providedIn: 'root'
})
export class MenuService {
  @Output() consultaMenu: EventEmitter<any> = new EventEmitter();
  url = 'http://localhost:824-aula2-gimnasio-g24/app-Ecommerce/';
  /* url = "https://planteador.com.ar/my-app/conexion/"; */

  constructor(private http: HttpClient) {}

  ngOnInit() {
    this.loginService.consultaMenu.emit(this.Usuarios.usuario);
  }

  refresh(): void {
    this.loginService.consultaMenu.subscribe((data: any) => {
      this.usuario = data;
    });
  }
}
```

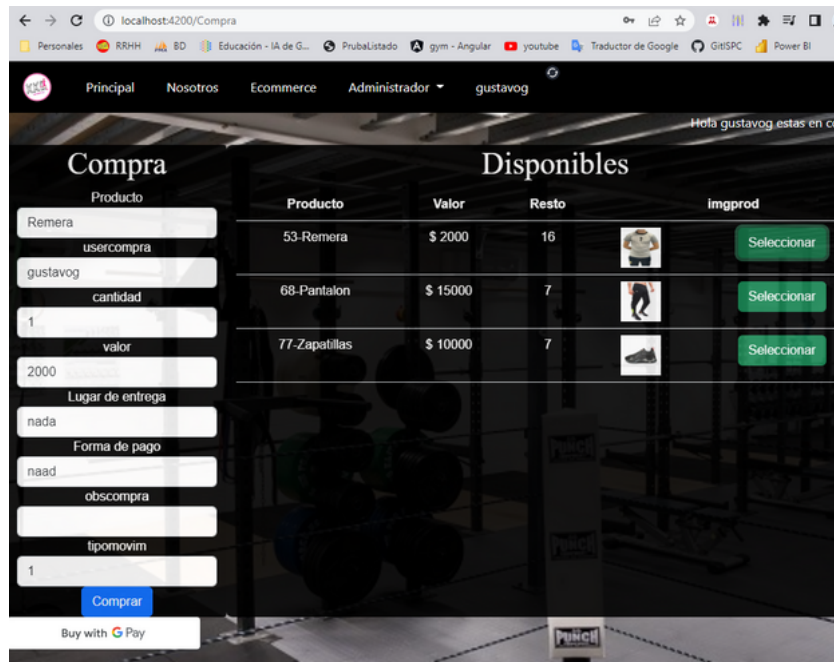
```
<header class="menu container-fluid">
  <nav class="navbar navbar-expand-md navbar-dark estilo_navbar">
    <div class="container-fluid " (click)="enviarData()">
      <a routerLink="home">
        
      </a>
      <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#MenuNavegacion">
        <span class="navbar-toggler-icon"></span>
      </button>
    </div>
  </nav>
</header>
```

```
# menu.component.css
<? menu.component.html
menu.component.ts
menu.service.ts
Layout.module.ts
Page
  compra
  home
  ingresomovim
```

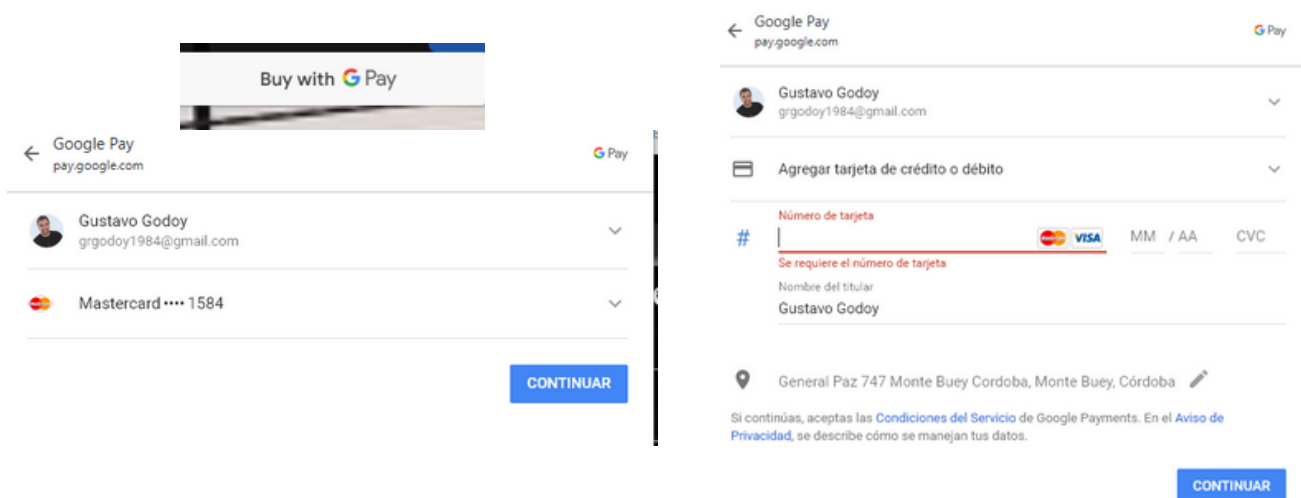
```
ngOnInit() {
  this.loginService.consultaMenu.subscribe((data: any) => {
    this.usuario = data;
  });
}
```

```
<div *ngIf="usuario" class="container-fluid">
  <div class="presentacion">Hola {{usuario}} estas en compras</div>
  <div class="row">
    <div class="col-3 CentroHome" id="Colliz">
      <div class="form-group">
        <input type="text" value="usuario">
      </div>
      <div class="form-group">
        <input type="password" value="password">
      </div>
      <div class="form-group">
        <input type="button" value="Enviar" />
      </div>
    </div>
    <div class="col-9">
      <div class="form-group">
        <input type="text" value="usuario">
      </div>
      <div class="form-group">
        <input type="password" value="password">
      </div>
      <div class="form-group">
        <input type="button" value="Enviar" />
      </div>
    </div>
  </div>
```

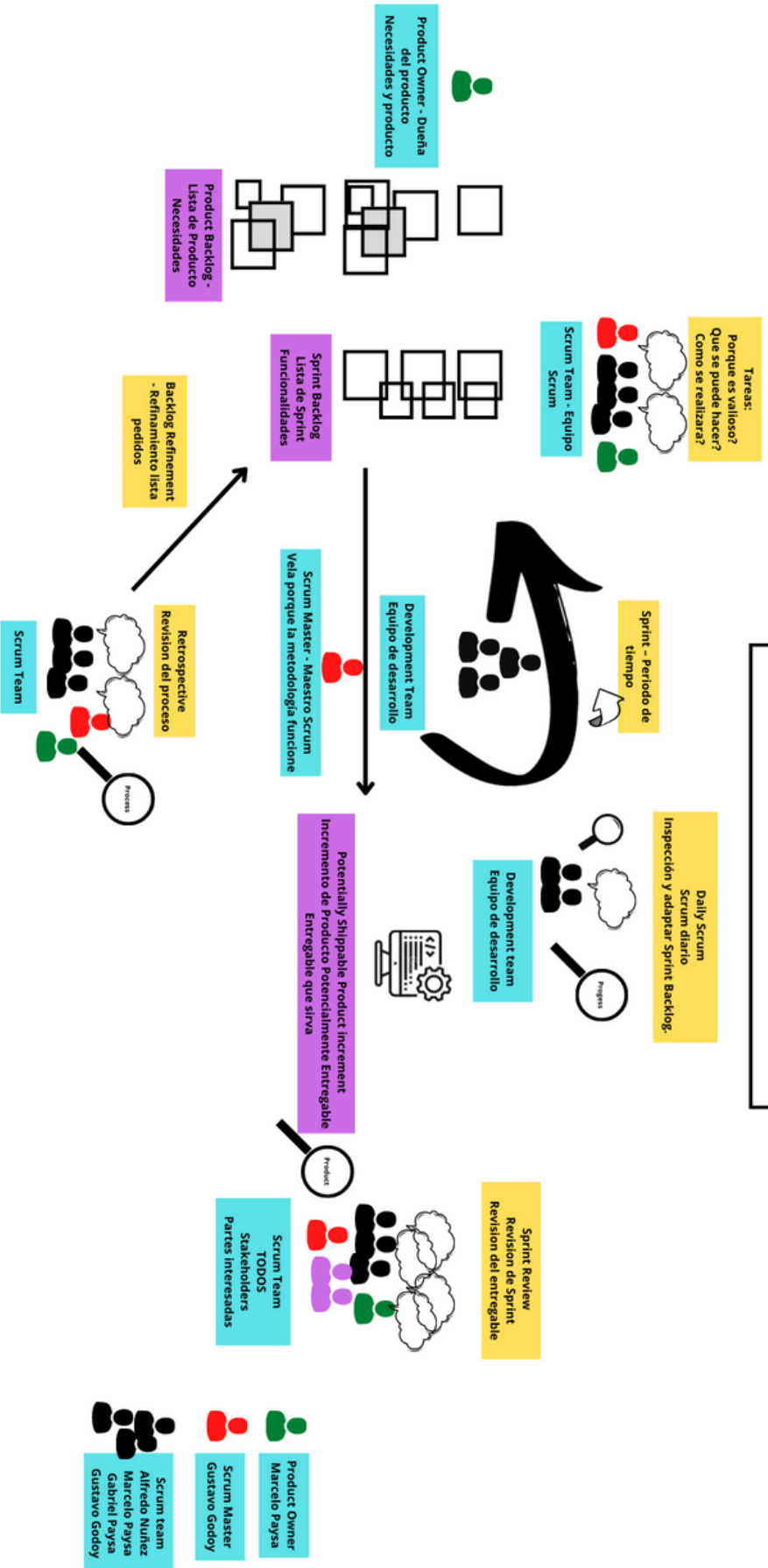
- Tablero de control para usuarios: Mostrar catálogo (mínimo 2 productos o servicios o suscripción). Seleccionar para comprar. Simular Pago con API Mercado Pago o API similar
 - La página muestra un listado de productos Disponibles, al seleccionar un producto lleva al formulario los datos, y desde el componente que tiene el usuario (en este caso el menú), completa la celda de usercompra (por ahora se hace visible solo para la materia, en caso de que sea una compra real, no se debería ver)



- Con Respecto a la forma de pago, honestamente me hubiera gustado tener más tiempo para desarrollar algo mejor, pero tome la forma de pago de google pay, ya que tiene un entorno de prueba:
 - La tarjeta se carga previamente.
 - La version de prueba se ve en la pagina de pay google console



The scrum framework / El marco de Scrum



Referencia:

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

Links	Referencia
GitHub	https://github.com/PPROF2-2022ProgWeb/g24-aula2-gimnasio-g24
Pagina prueba	http://planidear.com.ar/g24-aula2-gimnasio-g24/index.html
Yotube (Canal Yo Androide)	https://www.youtube.com/watch?v=oMxVV37gQb4&list=PLI4dAv2GvnrQqCC1cjUdoE_lq4ke-EOmx&index=4
Educacion IT	Curso Angular

Personal Involucrado:

Nombre	Responsabilidad - Rol	e-mail
Gustavo Godoy	Desarrollo, base de datos, actualizacion GitHub, Diseño y Desarrollo, Scrum Master e informe IEEE	grgodoy1984@gmail.com

