

CUSTOMER UNION MARKET



Creativos expertos que entregan resultados

I.E.S FRANCISCO AYALA

CURSO 2022/2023

Realizado por: Miguel Hurtado Mesa y Raúl González Díaz

1. Descripción de la empresa

En este proyecto se recogen los aspectos más relevantes del proyecto de iniciativa empresarial desarrollado por los promotores del Proyecto Empresarial "CUM" en la materia Entorno y Desarrollo a lo largo del curso 2022-2023.

- **Nombre de la empresa:** Customer Union Market
- **Actividad empresarial:** Servicio de marketing digital

1.2 Sedes

Actualmente nuestra empresa se desarrolla en una **oficina** que tenemos en **alquiler**, junto con un **almacén**.



Nuestra oficina se encuentra en calle Recogidas, N°12.

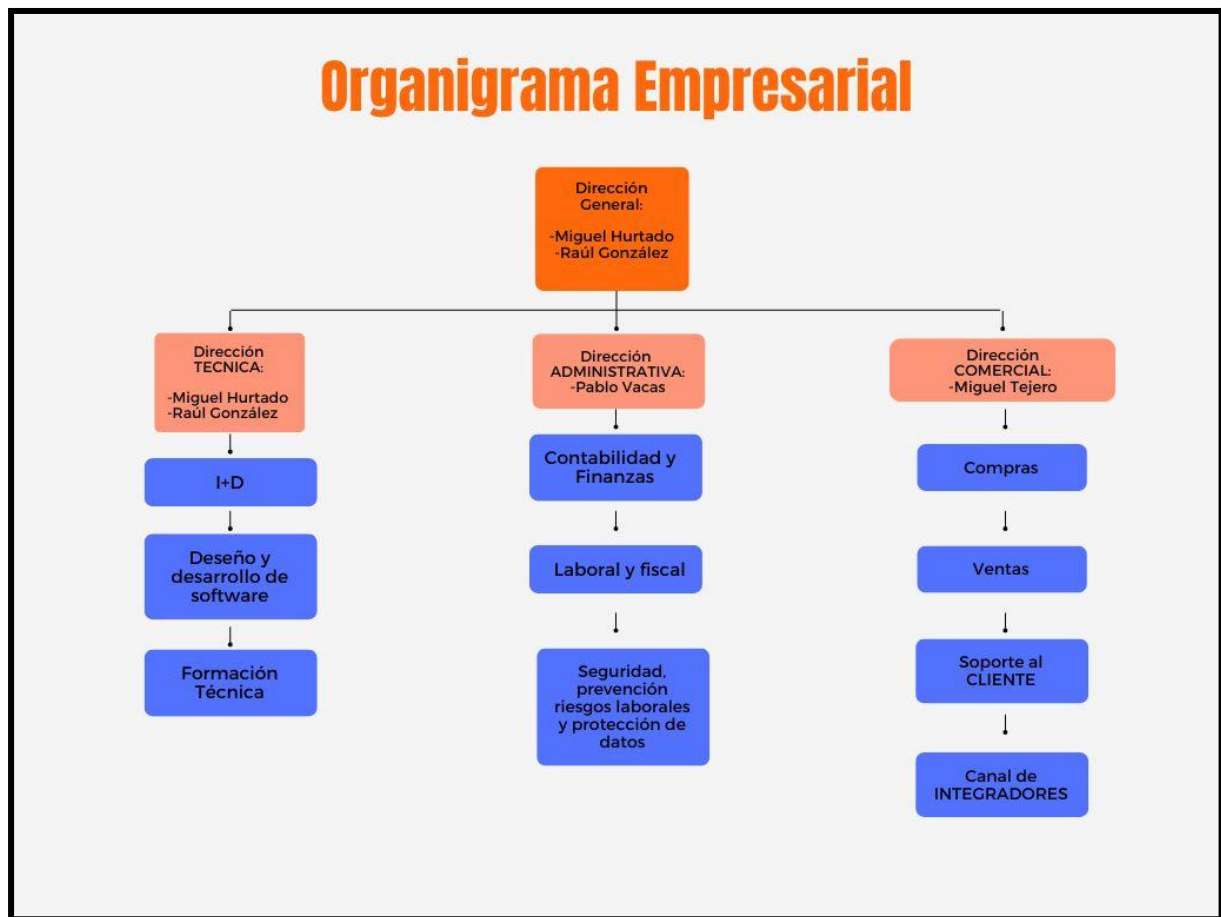
La **función** de esta oficina es única y exclusivamente para desarrollar y mantener el software de la empresa.

En este lugar se encuentran los programadores y administrativos de finanzas.

- **La oficina se divide en 3 grandes salas:** Sala de Programadores, Sala de Administración Comercial y Sala de Contabilidad y finanzas.

Nuestro almacén sirve para almacenar los distintos **sistemas** informáticos y **componentes** electrónicos. Además disponemos de productos de **merchandising** que almacenamos también en este almacén.

1.3 Organigrama



1.4 Empleados

Emprendedores de la empresa: Contamos con un equipo formado por Miguel Hurtado y Raúl González

Nosotros nos encargaremos de llevar a cabo la idea de nuestra empresa y nos caracterizamos como **emprendedores** de la empresa para posteriormente ser los **empresarios** que lleven a cabo la labor de la empresa y sacar adelante nuestra idea.

Miguel Hurtado como director ejecutivo de la empresa y socio ha hecho cursos como trafficker digital, community manager y fotografía. En la actualidad está haciendo un curso de inversión en divisas además de uno basado en creación de páginas web para empresas.

Raúl Gonzalez también es director ejecutivo y socio de la empresa, cuenta con una cierta experiencia en las redes sociales al llevar varias cuentas en instagram específicamente, además de otras plataformas como twitter y facebook. También es experto en la creación y desarrollo de software.

Ambos tenemos ciertos conocimientos en el mercado online y somos expertos en la tecnología.

Pablo Vacas y Miguel Tejero llegaron más tarde al proyecto, pero también están avanzando a pasos agigantados y están adquiriendo muchos conocimientos que hacen que la empresa vaya por el buen camino.

1.5 Descripción del servicio

Este Proyecto presenta un servicio de **lanzamiento** de la marca o empresa **al mundo online** y al mercado virtual creando más oportunidad de ventas y cantidad de clientes además de visibilidad de la marca en sí.

Ayudamos a las marcas a **desarrollar** y definir su voz. Los tiempos han cambiado, pero nuestro amor por la creatividad y la resolución de problemas, no.

Nos impulsan tanto la **tecnología** como la **innovación** con el fin de garantizar que cumplamos nuestra misión clave de ayudar a nuestros clientes a alcanzar su **futuro exitoso**.

También integraremos a nuestra empresa un servicio de **Community Manager**, el cual se basa en controlar, orientar y administrar la comunidad online y gestionar la identidad y la imagen de marca, creando y manteniendo relaciones estables y duraderas con sus clientes, sus fans en internet etc, además de facilitarles el uso de darse a conocer y ampliar su comunidad ofreciéndoles así anuncios en cualquier plataforma online o internet.

2. Recursos disponibles en la empresa

- **Sala de programadores:** sala con 8 equipos completos, conectados en red y con un sistema operativo basado en linux preparados para nuestro equipo de programadores.
- **Sala de Administración Comercial:** tenemos 4 equipos básicos, con una red telefónica para los encargados de la Administración Comercial.
- **Sala de Contabilidad y finanzas:** en esta sala tenemos instalados 2 equipos con acceso a la base de datos y con Microsoft Office instalado para poder realizar la contabilidad de la empresa.
- **Base de Datos:** tenemos una base de datos construida por nuestro equipo de programadores basada en Oracle en la cual almacenamos la información de los clientes dados de alta con sus servicios contratados y también la flota de productos de merchandising y los pedidos realizados.
- **Servidor:** Tenemos un servidor propio donde alojamos nuestra infraestructura y las páginas webs de nuestros clientes.
- **Internet:** plan de fibra contratado con 1Gbps simétricos para que nuestros empleados puedan trabajar con la máxima rapidez posible.

3. Requisitos de la aplicación

Nuestra aplicación web se va a dividir en distintos módulos y pestañas:

- **Sobre nosotros**

En este módulo se detalla una breve **descripción** del producto y servicio principal que ofrece la empresa.

Además se incluirá una **introducción** sobre como surgió la idea de emprendimiento, y un **resumen** de los **integrantes** de la empresa.

- **Contacto**

Se incluye un **formulario** para contactar mediante la página web, que posteriormente llega un aviso a nuestros correos electrónicos.

Estas consultas las resolveremos desde nuestra propia web donde tendremos acceso tanto para leer como para responder.

Para poder acceder al formulario se necesita un registro previo.

También se incluye **información de contacto**, como puede ser el teléfono o correo electrónico.

Por si al cliente le sirve de ayuda se presentará un conjunto de **preguntas frecuentes** para así evitar consultas reiterativas.

Como información adicional pondremos disponible también la dirección de nuestras **redes sociales**.

- **Descripción de servicios**

En esta parte se explican todos los servicios que se ofrecen en la empresa a modo de consulta con su debida **información**.

Cada servicio se organiza en un **artículo** donde se desplegará un botón para solicitar un precio. En esta parte accederemos a la pestaña de venta de servicios.

- **Venta de Servicios**

Cada servicio se organizará de forma esquemática con su debida **descripción** (breve), rango de **precio** (no se detalla el precio exacto) y una **imagen**.

Se presenta un **botón** para acceder a la página de contacto para que el cliente pueda solicitar un precio según el tipo de producto que esté interesado. Para acceder a este módulo se necesita el previo registro.

En esta parte se le dará un **plan específico**, y se le habilitará al cliente un módulo personalizado para poder proceder al pago y así desbloquear un **servicio de mensajería** más privado con los propios programadores.

- **Venta de Productos de Merchandising**

En este módulo se ofrecerán productos de merchandising.

Se dividirá de forma que quede el título del producto a primera vista, seguido de una imagen y un precio. Se pondrá también su talla, medidas...etc

Los clientes podrán ver las unidades restantes del producto antes del final de stock.

- **Inicio de sesión**

La página ofrecerá un **registro** previo y necesario para cualquier tipo de compra o solicitud que se quiera mandar.

De esta forma los clientes quedarán registrados en una **base de datos** para así poder también enviarles promociones a través de email.

El inicio de sesión también tendrá un apartado personalizado como se ha detallado antes en otro módulo.

Este apartado será exclusivo para clientes que hayan realizado una compra. Adicionalmente se les añadirá una nueva pestaña donde aparecerá **“Área Clientes”**

Procesos funcionales

Los clientes pueden acceder a la página principal y navegar por cada módulo anteriormente descrito. También para poder contratar los servicios o hacer algún tipo de compra y pedir presupuesto o contactar con atención al cliente, el cliente deberá darse de alta, pudiéndose hacer desde la pestaña **“registro”**; si ya es cliente, podrá acceder a su cuenta desde la pestaña **“inicio de sesión”**

Nosotros como dueños de la web podemos editar toda la página, dar de baja y bloquear a algunos clientes o IPs que consideremos sospechosas de posible ataque a la red. Además tenemos la opción de crear nuevos usuarios y claves a algún cliente que pueda tener problemas para poder darse de alta o acceder a su cuenta.

Procesos no funcionales

- La página web a los 20 minutos de inactividad, automáticamente volverá a la página de inicio y se cerrará la sesión, para ahorrar espacio en el servidor.
- A los cinco años de una cuenta sin ningún tipo de actividad, se eliminará automáticamente.
- Cuando quieras crear una nueva cuenta o iniciar sesión, deberás verificar que eres humano mediante un quiz sobre un pequeño puzzle de imágenes aleatorias que tendrás que encajar correctamente para finalizar dicha verificación.
- Sistema de verificación constante con el servidor que controlará que la conexión es correcta y estable.
- Automáticamente hará una recopilación de las preguntas más repetidas en el apartado de atención al cliente y las almacenará en el apartado de **FAQ** (Frequently Asked Questions).
- Cuando trates de iniciar sesión y tu usuario o contraseña sea incorrecta, aparecerá un mensaje de error, a los 10 fallos, la cuenta se bloqueará y la deberá desbloquear desde su correo electrónico a través de un mensaje que se enviará automáticamente.

Especificación de Casos de Uso

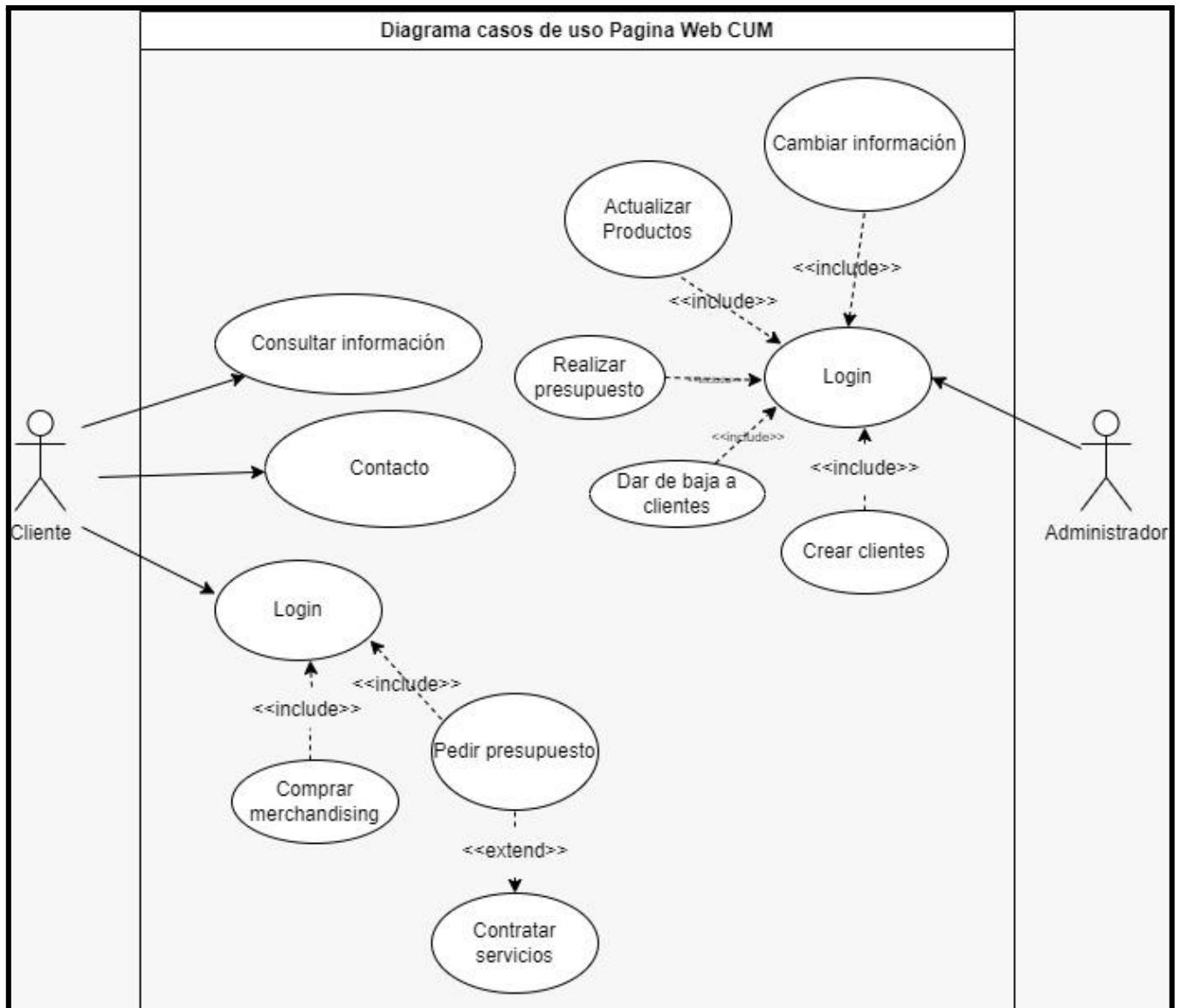
CUM (Customer Union Market)

Fecha: 20/12/2022



Creativos expertos que entregan resultados

Diagrama de Casos de Uso



Descripción de Actores

[CLIENTE]

Actor	<u>Cliente</u>	Identificador: 456421394
Descripción	Cliente que va a interactuar con nuestro software web	
Características	Cliente que debe estar registrado previamente. NO distinguimos entre sexo, puede ser cualquier tipo de cliente. El cliente será virtual.	
Relación	Debe registrarse y podrá iniciar sesión para pedir presupuesto al administrador para posteriormente contratar un servicio otorgado por el administrador o hacer una compra de merchandising.	
Referencias	Login, registro, pedir presupuesto, comprar merchandising, contratar servicios, contacto, consultar información.	

[ADMINISTRADOR]

Actor	<u>Administrador</u>	Identificador: 8519515952
Descripción	Administrador con acceso a la página web	
Características	El administrador debe de iniciar sesión para entrar en su área de edición. Dentro de ahí podrá moverse por las distintas herramientas para modificar el sitio web.	
Relación	Después de iniciar sesión, podrá cambiar la información de la web, actualizar los productos para que el cliente vea los cambios. Puede realizar presupuestos al cliente. Puede dar de baja a los clientes. Y puede crear clientes.	
Referencias	Login, registro, cambiar información, actualizar productos, realizar presupuesto, dar de baja a clientes, crear clientes	

Especificación de Casos de Uso

[Consultar Información]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición			
Descripción		Consultar Información	
Secuencia normal	Paso	Acción	
	1	Entrar en la página web	
	2	Seleccionar Consultar información	
	3		
	4		
	5		
	6		
	7		
	8		
	9		
	10		
	11		

[Contacto]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición		Entrar en la página web	
Descripción		Contacto	
Secuencia normal	Paso	Acción	
	1	Seleccionar Contacto	
	2	Rellenar el formulario de contacto	
	3	Enviar formulario de contacto	
	4		
	5		
	6		
	7		
	8		
	9		
	10		
	11		
Postcondición			
Excepciones	Paso	Acción	

[Iniciar Sesión]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición		Estar registrado en la pagina	
Descripción		Iniciar sesión	
Secuencia normal	Paso	Acción	
	1	Seleccionar Iniciar sesión	
	2	Introducir usuario	
	3	Introducir contraseña	
	4		
	5		
	6		
	7		
	8		
	9		
	10		
	11		
Postcondición			
Excepciones	Paso	Acción	
	2	Usuario incorrecto, vuelve a pedir el usuario	
	3	3.1	Contraseña incorrecta, vuelve a pedir la contraseña
		3.2	Si la contraseña es incorrecta 5 veces, se envia un correo para restablecerla

[Comprar Merchandising]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición		Iniciar sesión	
Descripción		Comprar merchandising	
Secuencia normal	Paso	Acción	
	1	Seleccionar la opción comprar merchandising	
	2	Añadir al carrito los articulos que desee	
	3	Al terminar la seleccion, ir a finalizar compra	
	4	Pagar los articulos seleccionados	
	5	confirmar la compra desde el correo recibido	
	6		
	7		
	8		
	9		
	10		
	11		
Postcondición			
Excepciones	Paso	Acción	
	2	Si no hay stock del producto seleccionado saldra un mensaje de no disponible	
	4	4.1	Si el pago no puede ser efectuado saldrá un mensaje para volver a intentarlo
		4.2	Si no se puede realizar en 5 intentos, se cancelara la compra

[Pedir Presupuesto]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición		Login como cliente	
Descripción		Pedir presupuesto	
Secuencia normal	Paso	Acción	
	1	Seleccionar pedir presupuesto	
	2	Describir los servicios deseados	
	3	Cuando se pida el presupuesto se recibirá un correo avisando de que lo ha solicitado	
	4	A los administradores les llega una notificación de nuevo presupuesto	
	5	Cuando se ha realizado el presupuesto, al cliente se le notificará	
	6	El cliente selecciona aceptar o rechazar presupuesto	
	7		
	8		
	9		
	10		

[Contratar Servicios]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición		El cliente deberá aceptar el presupuesto	
Descripción		Contratar servicios	
Secuencia normal	Paso	Acción	
	1	Cuando el cliente acepta el presupuesto deberá realizar una parte del pago	
	2	Cuando se realiza el pago, se notifica al equipo para que empiece el trabajo	
	3	Cuando el trabajo se termina se notifica al cliente	
	4	Si está de acuerdo con el resultado, aceptará el proyecto y deberá pagar la parte restante	
	5		
	6		
	7		
	8		
	9		
	10		
	11		
Postcondición			
Excepciones	Paso	Acción	
	1	Si el pago no se puede realizar, saldrá un mensaje de error y deberá reintentarlo	
	4	Si el pago no se puede realizar, saldrá un mensaje de error y deberá reintentarlo	

[Actualizar Productos]

NOMBRE		FUNCIÓN	
Versión			
Dependencias			
Precondición		Login como administrador	
Descripción		Actualizar productos	
Secuencia normal	Paso	Acción	
	1	Seleccionar la opción de actualizar productos	
	2	Seleccionar el producto que deseas modificar	
	3	Realizar las modificaciones que desee	
	4	guardar los cambios	
	5		
	6		
	7		
	8		
	9		

MANUAL DE USUARIO

CUSTOMER UNION MARKET

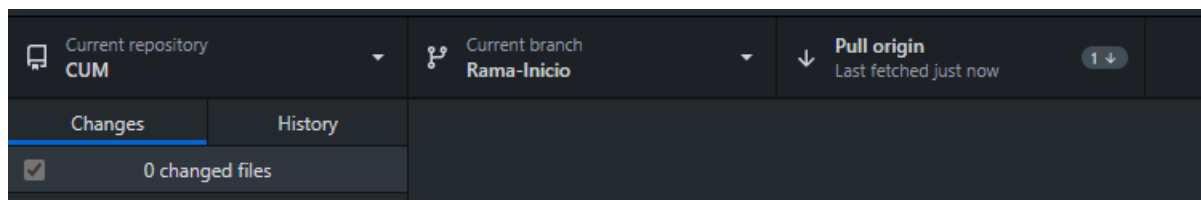
REPOSITORIO EN GITHUB

Hemos creado un repositorio compartido para poder trabajar de manera más individualizada y también para poder tener acceso a nuestro contenido desde la nube, de esta manera el acceso siempre está garantizado y evitamos perdidas de archivos o información.

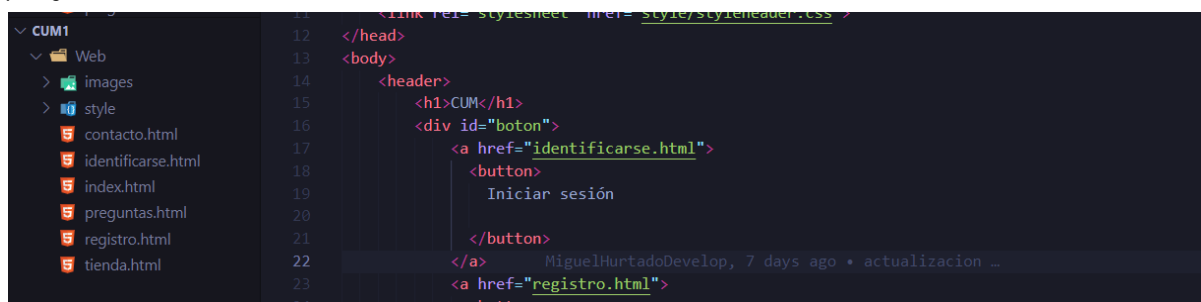
Nuestro repositorio tiene el nombre de **“CUM”**.

- **Enlace al repositorio:** [Click aquí](#)

Hemos utilizado la herramienta de escritorio de GitHub Desktop para poder actualizar las vistas y los commits de una manera más rápida.



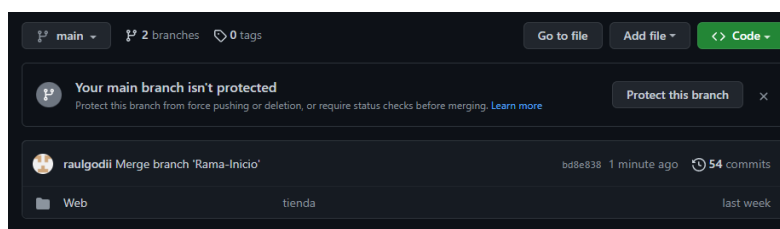
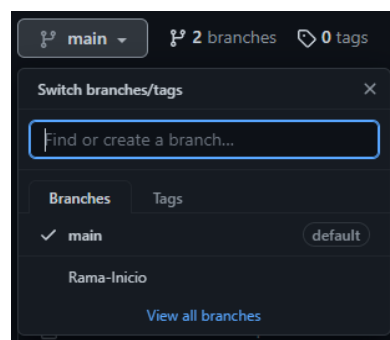
También nos hemos apoyado en el uso del entorno de desarrollo Visual Studio Code. Ha sido una herramienta fundamental para poder organizar nuestro proyecto de una manera más sencilla.



Para poder trabajar de una manera más organizada hemos creado dos ramas dentro del propio proyecto:

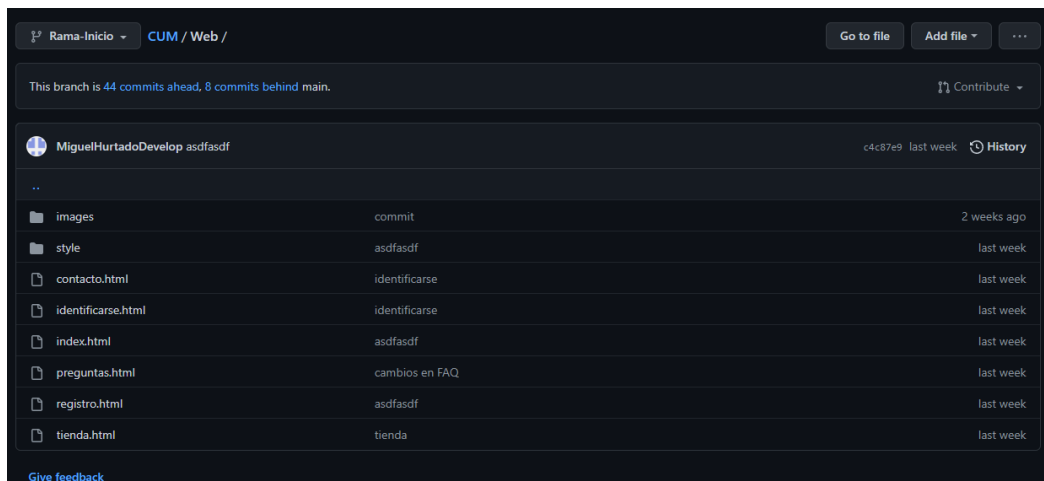
1. Rama “Main”

Esta rama la hemos fusionado con la segunda rama descrita a continuación una vez terminado el proyecto. Así damos por finalizada la primera versión de nuestra aplicación web.

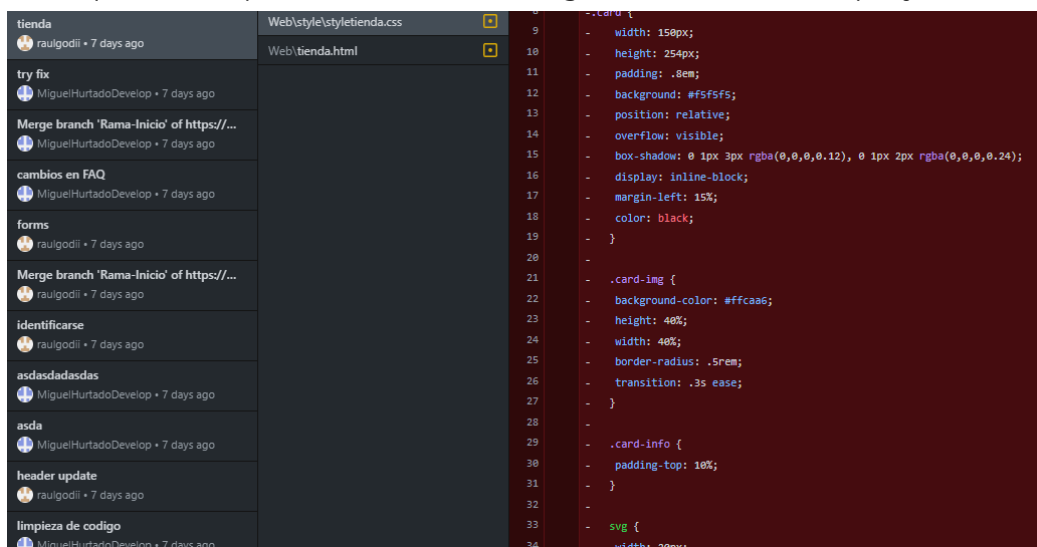


2. Rama “Rama-Inicio”

En esta rama hemos desarrollado todo nuestro proyecto de manera paralela a la rama ‘Main’ para poder hacer las modificaciones necesarias antes de lanzar nuestra versión final.



Aquí podemos observar varias modificaciones (con su commit correspondiente) desarrolladas a lo largo de la historia del proyecto:



PÁGINA WEB

Nuestra página web ha sido creada y desarrollada para poder facilitar al mundo del marketing digital una creación rápida, efectiva y económica de una página web.

- **Enlace a nuestra página web:** [Click aquí](#)

A continuación se procederá a relatar y explicar las distintas partes de nuestra web y sus respectivas utilidades:

VISTA CLIENTE

- Aquí tenemos el header de la página, con un menú para navegar por las diferentes páginas de nuestra web. También tenemos dos botones que redireccionan a las páginas de inicio de sesión y registro.



- Esta es la página de inicio, con una información orientativa de quien somos y qué hacemos la cual puede consultar cualquier usuario que acceda a la página. La página se divide en tres secciones :

1. La primera parte habla sobre nuestra historia como empresa y cuales son nuestros objetivos.



- Esta segunda parte habla de cuáles son los campos que trabajamos en nuestra empresa y lo que pretendemos conseguir con cada uno de nuestros trabajos.

¿QUE HACEMOS?



PUBLICIDAD

Muestra e impulsa la visibilidad de tu marca para ganar clientes y adherirte a un público respectivo a tu empresa. Nosotros tenemos la clave para que tu idea de negocio salga adelante.



DISEÑO WEB

Impulsa tu identidad de marca y no pierdas potenciales clientes con nuestro extraordinario servicio de Diseño web. Trabajamos juntos para combinar nuestras visiones creativas e idear algo realmente espectacular.

- Y en esta tercera encontramos los productos de nuestro merchandising más destacados.

¡Compra nuestro merchandising!



Camiseta Basica CUM


Product description and details


15.00€ 



Gorra CUM

Product description and details

9.00€ 

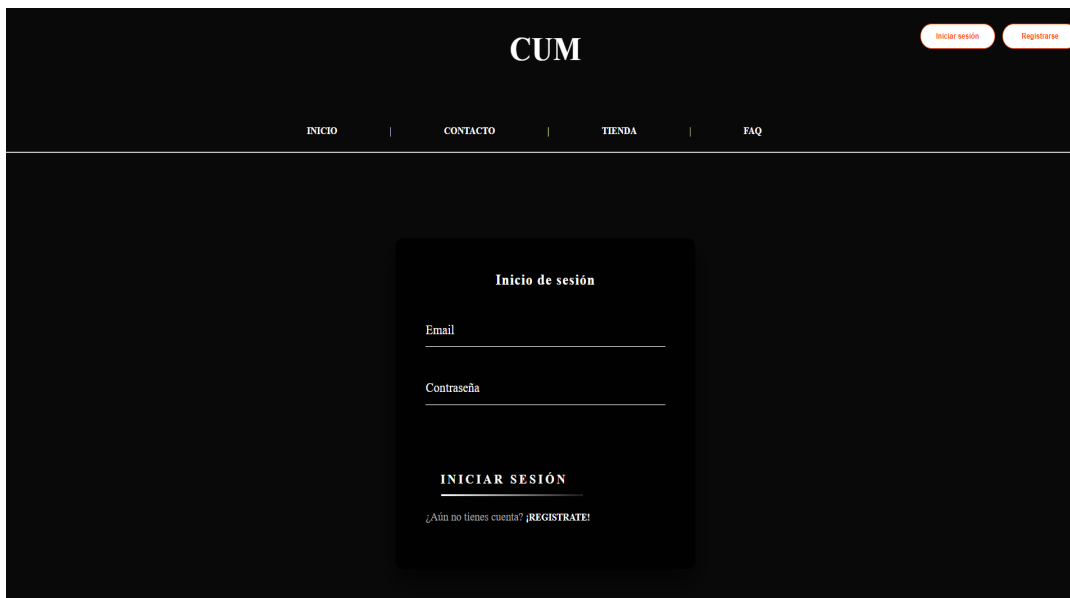


Bolsa CUM

Product description and details

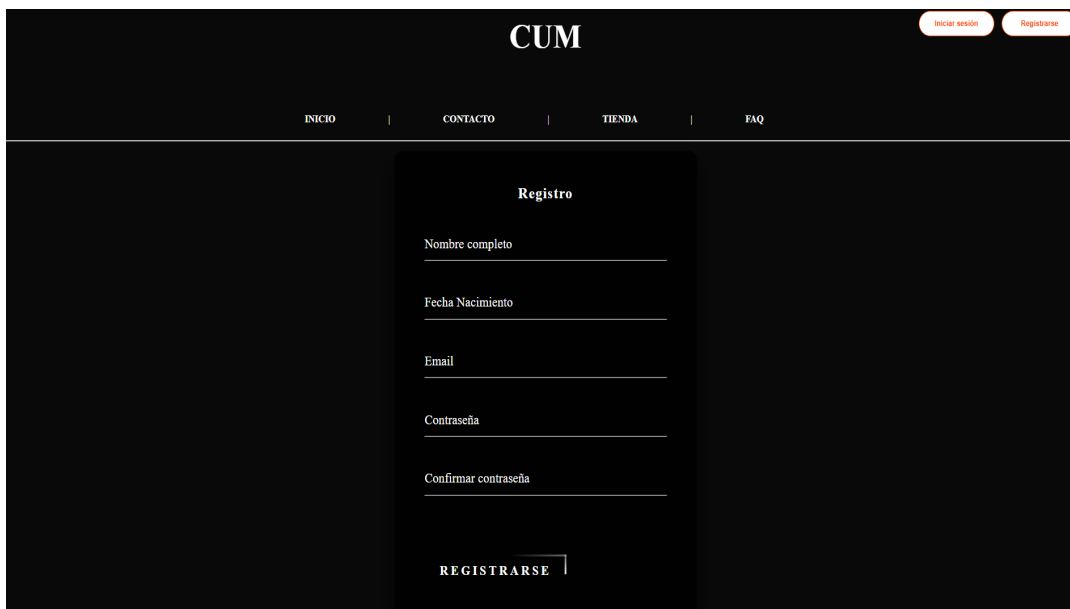
4.90€ 

- Aquí tenemos la página de inicio de sesión, donde si previamente te has registrado, podrás acceder a tu cuenta de cliente.



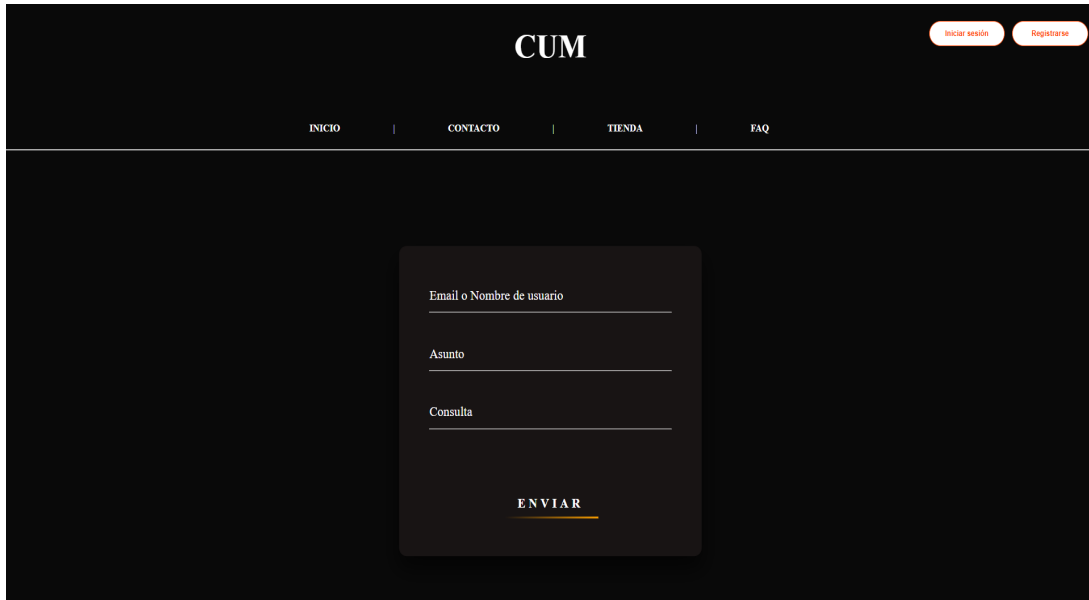
The screenshot shows the login page of the CUM website. The header features the CUM logo and navigation links: INICIO, CONTACTO, TIENDA, and FAQ. In the top right corner, there are buttons for 'Iniciar sesión' and 'Registrarse'. The main content area is a dark gray box with the title 'Inicio de sesión'. It contains two input fields labeled 'Email' and 'Contraseña'. Below these fields is a button labeled 'INICIAR SESIÓN'. At the bottom of the box, there is a link that says '¿Aún no tienes cuenta? ¡REGISTRATE!'.

- Esta es la página de registro, donde deberás introducir tus datos para crear una nueva cuenta de cliente.



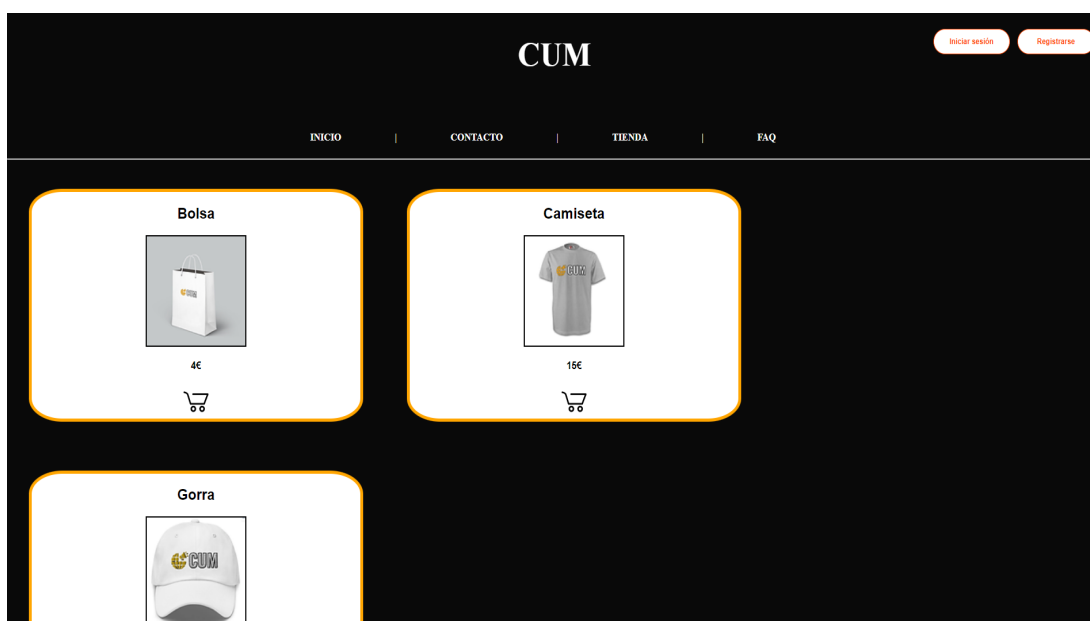
The screenshot shows the registration page of the CUM website. The header is identical to the login page, with the CUM logo, navigation links, and 'Iniciar sesión'/'Registrarse' buttons. The main content area is a dark gray box with the title 'Registro'. It contains five input fields labeled 'Nombre completo', 'Fecha Nacimiento', 'Email', 'Contraseña', and 'Confirmar contraseña'. Below these fields is a button labeled 'REGISTRARSE'.

- Nos encontramos en la página de contacto, para poder contactar con nosotros el usuario deberá de haberse registrado previamente. Mediante este apartado podrán preguntarnos cualquier duda y pedir presupuestos aproximados de cualquiera de nuestros servicios.



The screenshot shows the CUM website's contact page. At the top, the CUM logo is centered, with 'Iniciar sesión' and 'Registrarse' buttons on the right. Below the logo is a navigation bar with links: INICIO, CONTACTO, TIENDA, and FAQ. The main content area features a dark gray contact form with three input fields labeled 'Email o Nombre de usuario', 'Asunto', and 'Consulta'. Below these fields is a button labeled 'ENVIAR'.

- Este es el apartado de tienda, donde puedes ver todos nuestros productos. Para comprar cualquier producto será necesario iniciar sesión con tu cuenta.



- Y por último tenemos la página de Preguntas Frecuentes donde recopilamos las preguntas más frecuentes de nuestros clientes para que todos los usuarios puedan consultarlas con un simple vistazo.



VISTA ADMINISTRADOR

Para acceder a la parte de administrador, primero deberás de iniciar sesión en la página de inicio de sesión con los datos propios de cada administrador.

- En la página inicial podrá modificar, quitar y añadir secciones para poder actualizar la página.
- En el apartado de contacto podrá ver y contestar todos los mensajes que han enviado nuestros clientes y derivarlos al departamento adecuado dentro de nuestra empresa.
- En la tienda, podrá modificar, quitar y añadir nuevos productos además de ver los pedidos efectuados por los clientes y derivarlos al departamento de ventas.
- En la página de preguntas frecuentes podrá modificar, quitar y añadir preguntas con su respectiva respuesta.

IMPLEMENTACIÓN EN JAVA

CUSTOMER UNION MARKET

PARTE 1: IMPLEMENTACIÓN EN JAVA

CASO DE USO 1 - LOGIN

Hemos desarrollado el caso de uso "login.java" que consiste en el inicio de sesión de nuestros clientes. Se ha llevado a cabo el desarrollo de la siguiente clase en java:

```
public class Login {
    private Map<String, String> users;

    public Login() {
        users = new HashMap<>();
        // Agregar usuarios y contraseñas
        users.put( key:"usuario1", value:"contrasena1");
        users.put( key:"usuario2", value:"contrasena2");
        users.put( key:"usuario3", value:"contrasena3");
    }

    public boolean authenticate(String username, String password) {
        // Verificar si el usuario existe y la contraseña es correcta
        if (users.containsKey( key:username) && users.get( key:username).equals( anObject:password)) {
            return true; // Autenticación exitosa
        }
        return false; // Autenticación fallida
    }

    public static void main(String[] args) {
        Login login = new Login();

        // Ejemplo de uso
        String username = "usuario1";
        String password = "contrasena1";
        boolean isAuthenticated = login.authenticate(username, password);

        if (isAuthenticated) {
            System.out.println( x:"Inicio de sesión exitoso");
        } else {
            System.out.println( x:"Inicio de sesión fallido");
        }
    }
}
```

La clase **Login** tiene un mapa (**users**) que almacena los nombres de usuario como claves y las contraseñas correspondientes como valores. En el constructor de la clase, se agregan algunos usuarios de ejemplo al mapa.

El método ***authenticate*** toma un nombre de usuario y una contraseña como parámetros y verifica si el usuario existe en el mapa y si la contraseña coincide. Devuelve true si la autenticación es exitosa y false en caso contrario.

En el método ***main***, se crea una instancia de la clase ***Login*** y se realiza un ejemplo de uso llamando al método ***authenticate*** con un nombre de usuario y contraseña específicos. Dependiendo del resultado, se muestra un mensaje indicando si el inicio de sesión fue exitoso o no.

CASO DE USO 2 - PEDIR PRESUPUESTO

Hemos desarrollado el caso de uso “presupuesto.java” que consiste en recibir un presupuesto tras elegir una serie de servicios. Se ha llevado a cabo el desarrollo de la siguiente clase en java:

```
public class Presupuesto {
    private Map<String, Double> preciosServicios;
    private List<String> serviciosSeleccionados;

    public Presupuesto() {
        preciosServicios = new HashMap<>();
        serviciosSeleccionados = new ArrayList<>();

        // Configurar los precios de los servicios
        preciosServicios.put( key: "Marketing", value: 500.0);
        preciosServicios.put( key: "Publicidad", value: 1000.0);
        preciosServicios.put( key: "Community Manager", value: 800.0);
        preciosServicios.put( key: "Diseño gráfico", value: 600.0);
        preciosServicios.put( key: "Desarrollo web", value: 1500.0);
    }

    public void seleccionarServicio(String servicio) {
        // Verificar si el servicio está disponible y agregarlo a la lista de servicios seleccionados
        if (preciosServicios.containsKey( key: servicio)) {
            serviciosSeleccionados.add( e: servicio);
            System.out.println("Servicio \" + servicio + "\" seleccionado.");
        } else {
            System.out.println("El servicio \" + servicio + "\" no está disponible.");
        }
    }

    public void mostrarServiciosSeleccionados() {
        System.out.println( x: "Servicios seleccionados:");
        for (String servicio : serviciosSeleccionados) {
            System.out.println("- " + servicio);
        }
    }
}
```



```
public double obtenerPresupuesto() {
    double presupuestoTotal = 0.0;
    for (String servicio : serviciosSeleccionados) {
        if (preciosServicios.containsKey(key:servicio)) {
            presupuestoTotal += preciosServicios.get(key:servicio);
        }
    }
    return presupuestoTotal;
}

public static void main(String[] args) {
    Presupuesto presupuesto = new Presupuesto();

    // Ejemplo de uso
    presupuesto.seleccionarServicio(servicio:"Marketing");
    presupuesto.seleccionarServicio(servicio:"Publicidad");
    presupuesto.seleccionarServicio(servicio:"Desarrollo web");
    presupuesto.seleccionarServicio(servicio:"SEO"); // Servicio no disponible

    presupuesto.mostrarServiciosSeleccionados();

    double totalPresupuesto = presupuesto.obtenerPresupuesto();
    System.out.println("Presupuesto total: $" + totalPresupuesto);
}
```

La clase **Presupuesto** es una representación simple de un sistema de solicitud de presupuestos en una tienda de marketing digital. Permite a los clientes seleccionar servicios disponibles y calcular el presupuesto total en función de los servicios seleccionados.

La clase **Presupuesto** contiene los siguientes atributos:

preciosServicios: Un mapa que almacena los precios de los servicios disponibles. La clave es el nombre del servicio y el valor es el precio asociado.

serviciosSeleccionados: Una lista que almacena los servicios seleccionados por el cliente.

La clase tiene los siguientes métodos:

Presupuesto(): El constructor de la clase. Inicializa los atributos y configura los precios de los servicios disponibles.

seleccionarServicio(String servicio): Permite al cliente seleccionar un servicio especificado por su nombre. Verifica si el servicio está disponible y lo agrega a la lista de servicios seleccionados.

mostrarServiciosSeleccionados(): Muestra por pantalla la lista de servicios seleccionados.

obtenerPresupuesto(): Calcula y devuelve el presupuesto total sumando los precios de los servicios seleccionados.

El método ***main*** es un ejemplo de uso de la clase ***Presupuesto***. En este ejemplo, se crea una instancia de la clase, se seleccionan algunos servicios y se muestra la lista de servicios seleccionados. Luego, se llama al método ***obtenerPresupuesto*** para calcular el presupuesto total y se muestra en la consola.

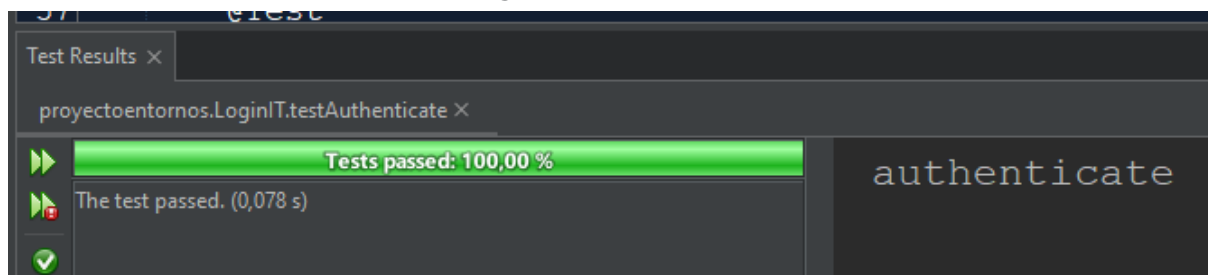
PARTE 2: PRUEBAS

Para comprobar que nuestros métodos y clases son efectivas hemos desarrollado gracias a JUnit una serie de tests. Los resultados se muestran a continuación:

- CASO DE USO 1: LOGIN

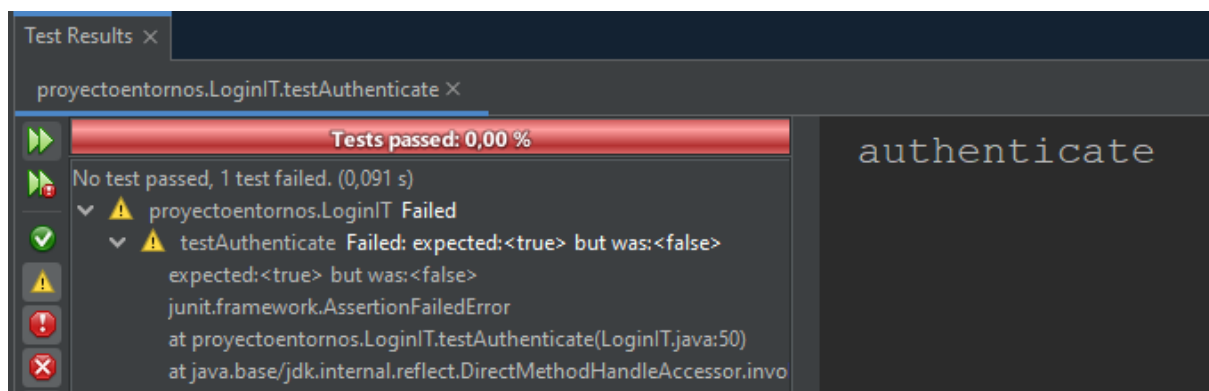
```
/**
 * Test of authenticate method, of class Login.
 */
@Test
public void testAuthenticate() {
    System.out.println(x:"authenticate");
    String username = "usuario1";
    String password = "contrasena1";
    Login instance = new Login();
    boolean expectedResult = true;
    boolean result = instance.authenticate(username, password);
    assertEquals( expected:expectedResult, actual:result);
    // TODO review the generated test code and remove the default call to fail.
}
```

Hemos ejecutado el test con el usuario: *usuario1* y *contrasena1* que previamente estaba creado en la clase como puede comprobarse más arriba. El resultado ha sido el siguiente:



Para terminar de comprobar la eficiencia del método hemos comprobado con *usuario1* y *contrasena2* y este ha sido el resultado:

```
public void testAuthenticate() {
    System.out.println( x:"authenticate");
    String username = "usuario1";
    String password = "contrasena2";
    Login instance = new Login();
    boolean expResult = true;
    boolean result = instance.authenticate(username, password);
    assertEquals( expected:expResult, actual:result);
    // TODO review the generated test code and remove the default call to fail.
}
```



De esta manera afirmamos que nuestro método actúa correctamente y se comporta como deseamos. Finalizamos las pruebas de este método con satisfacción.

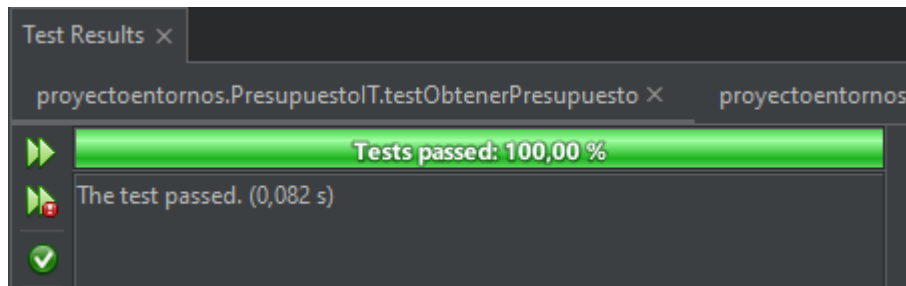
- **CASO DE USO 2: PRESUPUESTO**

A continuación hemos comprobado la eficiencia de nuestra segunda clase **Presupuesto**.

Para ello vamos a comprobar el método **obtenerPresupuesto()** y lo hemos ejecutado sin valores, sin servicios seleccionados, por lo que el resultado esperado debería de ser 0.

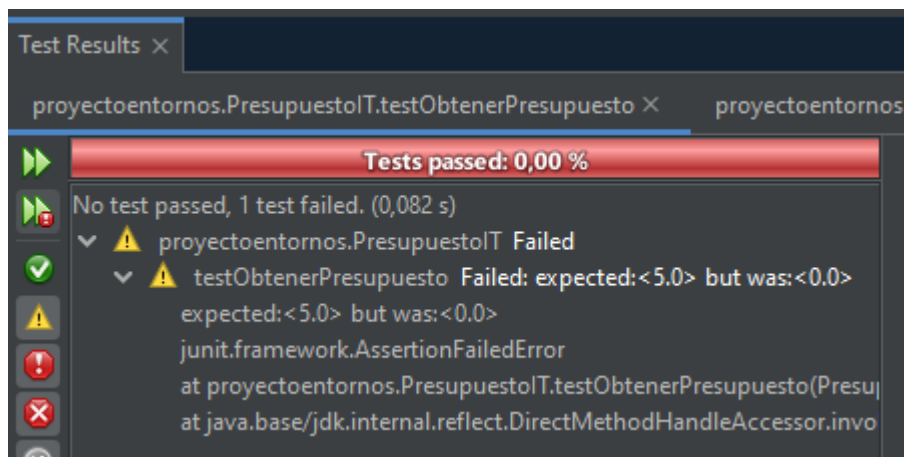
```
public void testObtenerPresupuesto() {
    System.out.println( x:"obtenerPresupuesto");
    Presupuesto instance = new Presupuesto();
    double expResult = 0.0;
    double result = instance.obtenerPresupuesto();
    assertEquals( expected:expResult, actual:result, delta:0);
    // TODO review the generated test code and remove the default call to fail.
}
```

Y este es el resultado:



Si cambiamos el valor esperado a "5" el resultado es el siguiente:

```
public void testObtenerPresupuesto() {  
    System.out.println( x: "obtenerPresupuesto");  
    Presupuesto instance = new Presupuesto();  
    double expectedResult = 5;  
    double result = instance.obtenerPresupuesto();  
    assertEquals(expected: expectedResult, actual: result, delta: 0);  
    // TODO review the generated test code and remove the default call to fail.  
}
```



Por lo tanto afirmamos y comprobamos que nuestro método **ObtenerPresupuesto()** funciona correctamente.

Para terminar de probar el programa hemos ejecutado el método **main()** con unos previos valores ya seleccionados en el método **main()**:

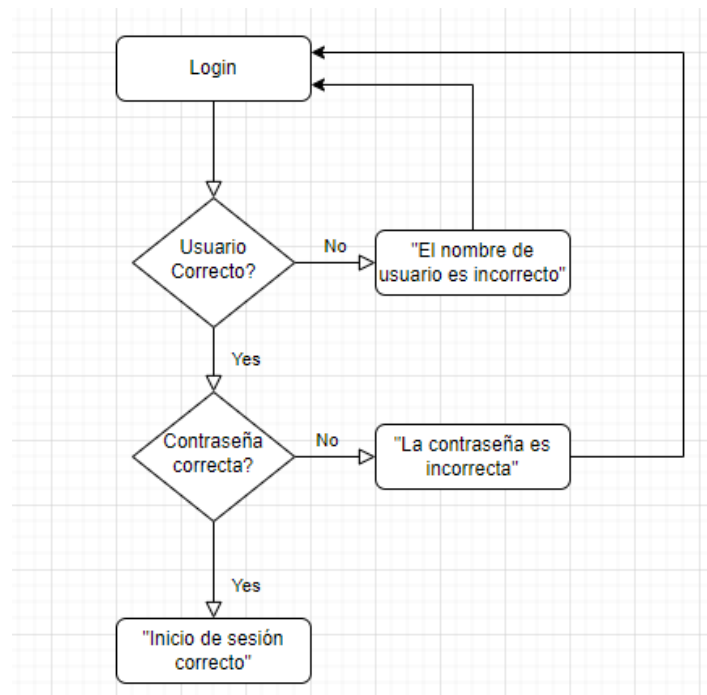
```
Tests passed: 100,00 %
The test passed. (0,084 s)
```

```
Servicio "Desarrollo web" seleccionado.
El servicio "SEO" no está disponible.
Servicios seleccionados:
- Marketing
- Publicidad
- Desarrollo web
Presupuesto total: $3000.0
```

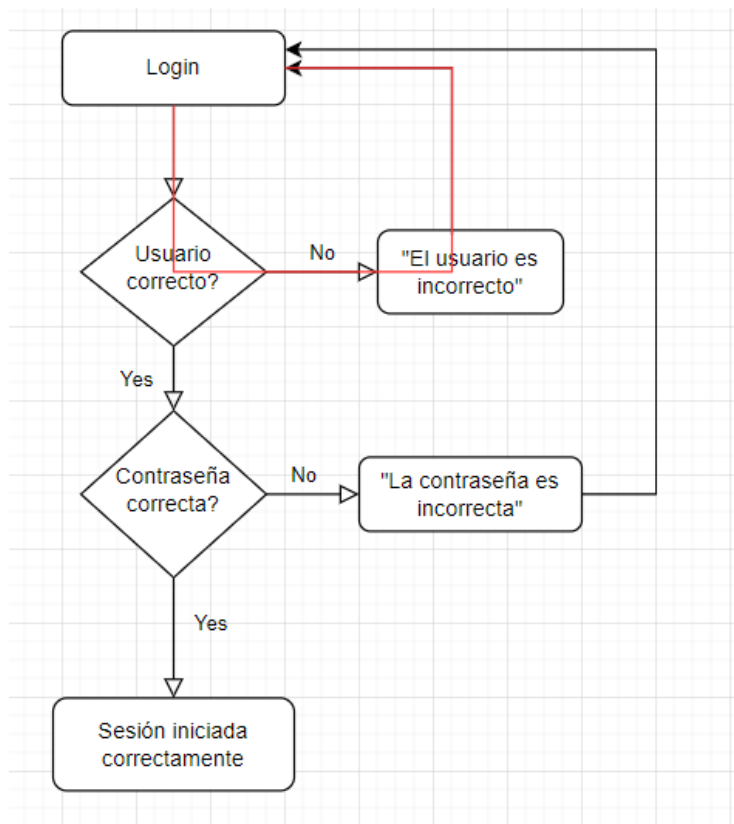
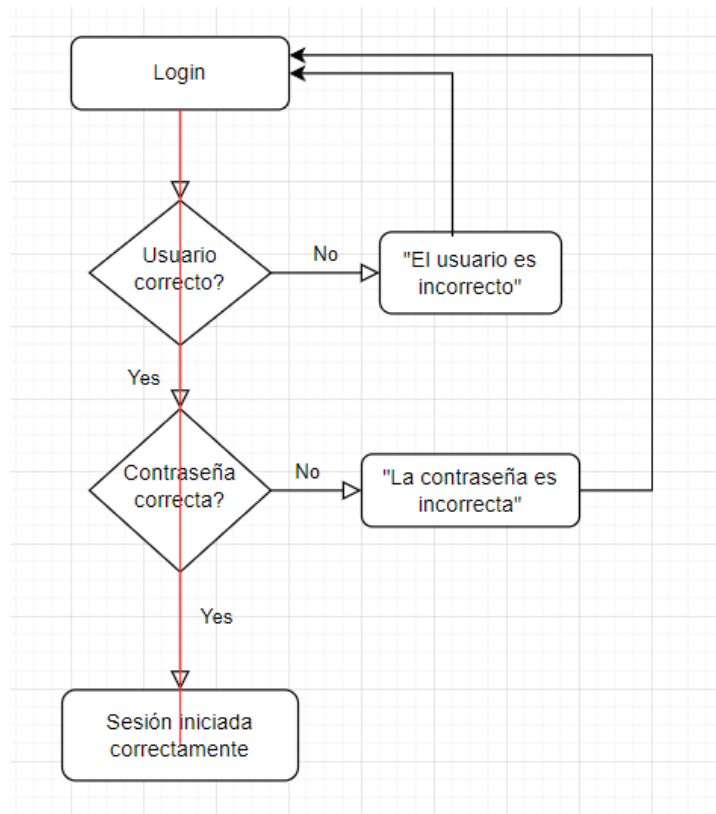
Finalmente afirmamos que la clase funciona perfectamente y a medida.

Diagramas de flujo

Diagrama de flujo (clase Login)



- **Caminos Básicos**



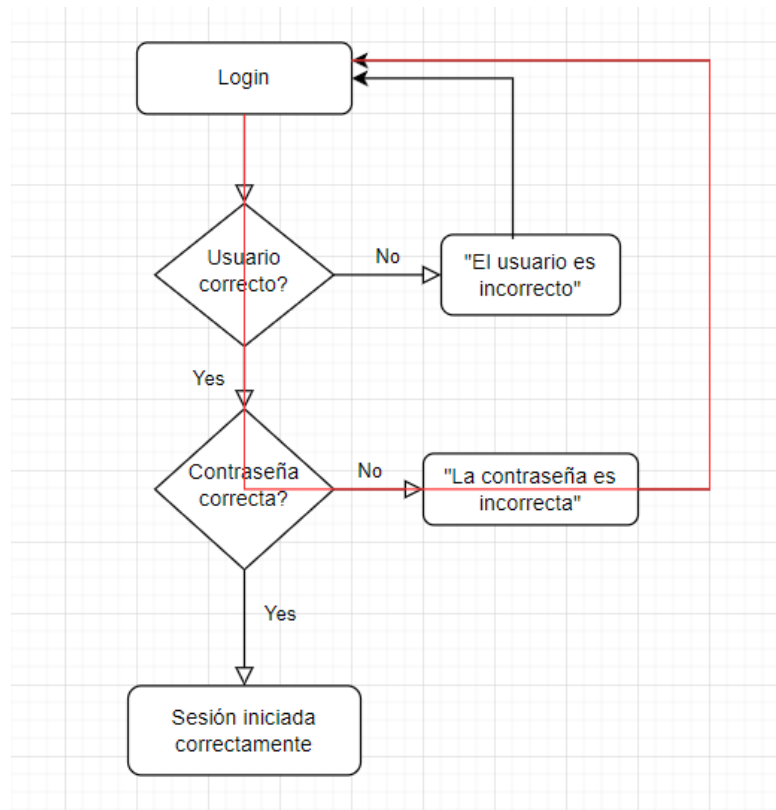
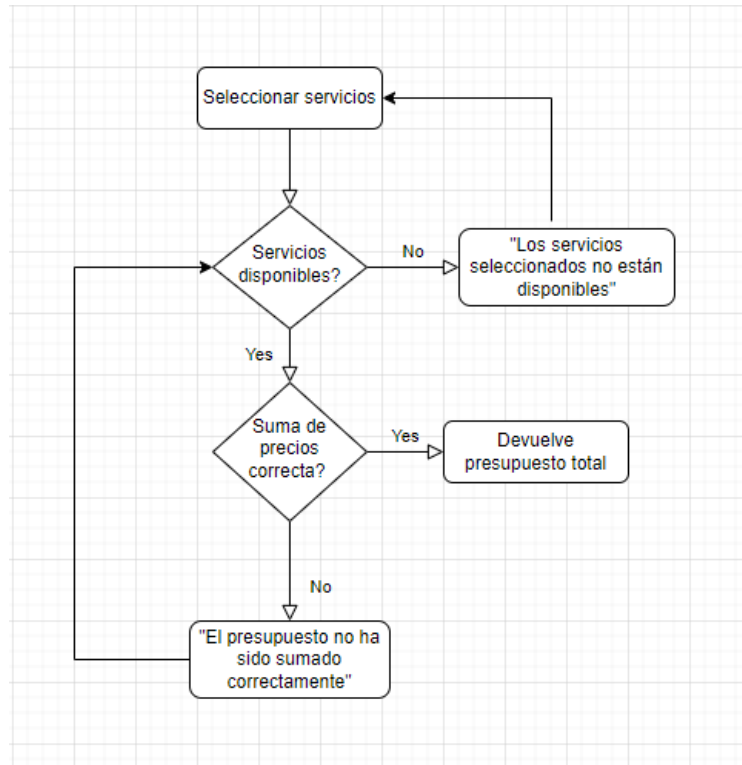


Diagrama de flujo (Clase Pedir Presupuesto)



- Caminos básicos

