

Documentación del sitio Web



**UNIVERSIDAD
DE GRANADA**

Alvaro Vega Romero
Esther García Gallego
Sandra Reinoso Ortega
Alexander Collado Rojas
Javier Osaka Lozano

Análisis y Diseño del Sistema Web

Parte de Análisis

El sistema de la web será el encargado de centralizar los productos que se vayan a vender, de tal manera que aquí las funcionalidades serán respecto a la organización de ventas, creaciones de productos y administración de los usuarios.

1. Requisitos Funcionales

- Dar alta usuario
- Poner un producto a la venta
- Iniciar sesión como usuario
- Modificar un producto a la venta

Detalle	Descripción
RF #	1
Nombre	Dar alta usuarios
Descripción	Permite el alta de un usuario en el sistema
Entrada	Datos del cliente (Nombre, apellidos, email, teléfono, username, contraseña)
Procesamiento	Prerrequisito: acceso no identificado] Registro de la cuenta en la BD externa al sistema Postrequisito: acceso identificado.
Salida	Acceso a la aplicación web

Detalle	Descripción
RF #	2
Nombre	Poner un producto a la venta
Descripción	Se incluirá en el sistema un nuevo producto que se venderá
Entrada	Datos del producto (nombre, precio, descripción, tipo, estado, distancia)
Procesamiento	Se añadirá un producto a la BD de Productos
Salida	Mensaje de confirmación de inserción de producto

Detalle	Descripción
RF #	3
Nombre	Modificar un producto
Descripción	Permite modificar los datos de un producto, así como eliminarlo.
Entrada	Nuevos datos a modificar
Procesamiento	Prerrequisito: El producto debe existir en la BD Se cambiarán sus atributos en la BD

Salida	Mensaje de confirmación de actualización del producto
--------	---

Detalle	Descripción
RF #	4
Nombre	Iniciar sesión como usuario
Descripción	Permite que un usuario se identifique para acceder al sistema
Entrada	Nombre de usuario y contraseña
Procesamiento	Prerrequisito: El usuario exista en la BD
Salida	Acceso a la aplicación web

Detalle	Descripción
RF #	5
Nombre	Modificar un usuario
Descripción	Permite modificar los datos de un usuario. Así mismo, permite eliminar el usuario
Entrada	Nuevos datos a modificar
Procesamiento	Prerrequisito: El usuario exista en la BD
Salida	Mensaje de confirmación

2. Requisitos no funcionales

- **Facilidad de uso:**
 - El sistema tendrá opciones fáciles, intuitivas y concisas.
- **Rendimiento:**
 - Corto tiempo de respuesta a los accesos a la base de datos
- **Implementación:**
 - Interfaz interactiva
 - Desarrollado en Ruby on Rails
- **Evolutivo:**
 - El sistema tendrá la capacidad de ampliarse más y añadir más requisitos funcionales según avance.
- **Fiabilidad:** Capacidad de recuperación de fallo de sistema por pérdida de datos

3. Listado de criterios de calidad

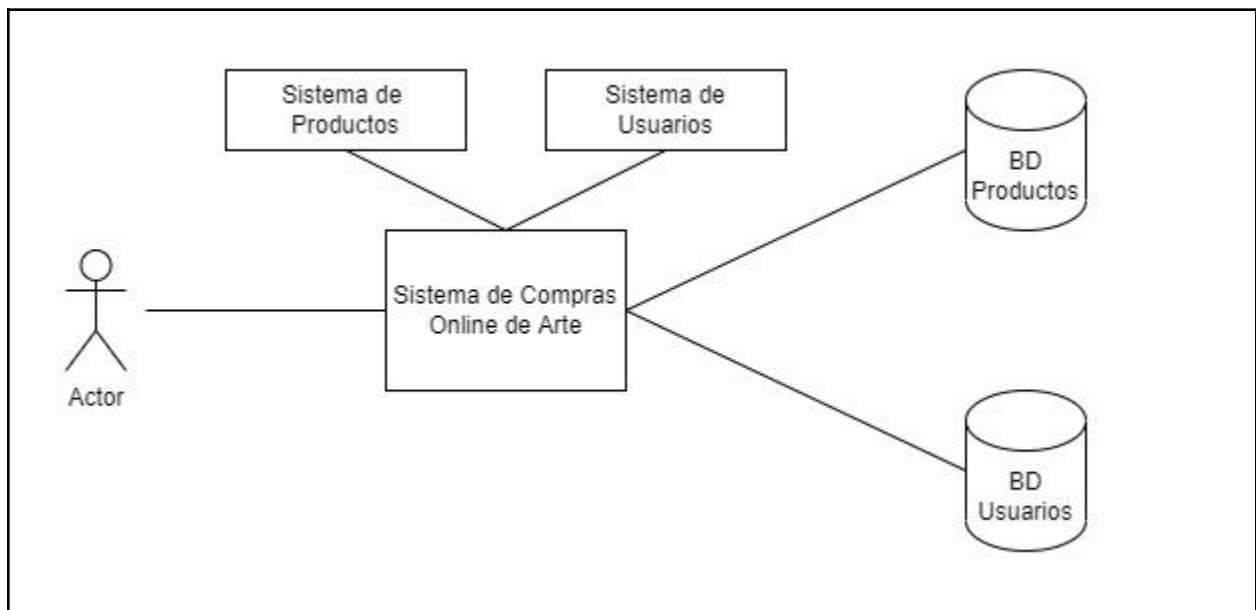
Para un proyecto más elaborado se realizan unos ciertos criterios de calidad éticos y que sigan las normativas legales.

- **Ético:**
 - Comercio Justo: Los artículos que se comercialicen cumplirán los criterios de solidaridad establecidos por la Coordinación Estatal de Comercio Justo.
- **Regulable:**
 - La aplicación debe cumplir la Ley Orgánica 3/2018, del 5 de diciembre, de Protección de Datos Personales y la garantía de los derechos digitales. Y la Ley 34/2002, del 11 de julio de servicios de la sociedad de la información y del comercio electrónico.
- **Accesible y fácil de usar:**
 - Por normativa, todas las páginas deben ser accesibles. Buscaremos que nuestra aplicación sea accesible y fácil de usar para todo tipo de personas.
- **Internacionalizable:**
 - La aplicación será fácilmente adaptable para distintas, idiomas, monedas...

4. Lista y descripción de las partes interesadas en el sistema

- **Desarrolladores:** Nosotros, encargados de elaborar la aplicación siguiendo un modelo arquitectónico y un diagrama de clases.
- **Vendedores:** Aquellos vendedores que ingresarán los productos que quieran vender a la página
- **Técnico de pruebas:** Al igual que los desarrolladores somos nosotros los encargados de realizar las pruebas necesarias para que todo funcione perfectamente.
- **Ingeniero de producción y mantenimiento y webmaster,** supervisando el mantenimiento de la aplicación y dando soporte hardware y middleware a esta.
- **Administrador del Sistema,** también para el mantenimiento de la aplicación.

5. Diagrama de Modelo de Contexto



6. Diseño de Pruebas de Sistema Integración

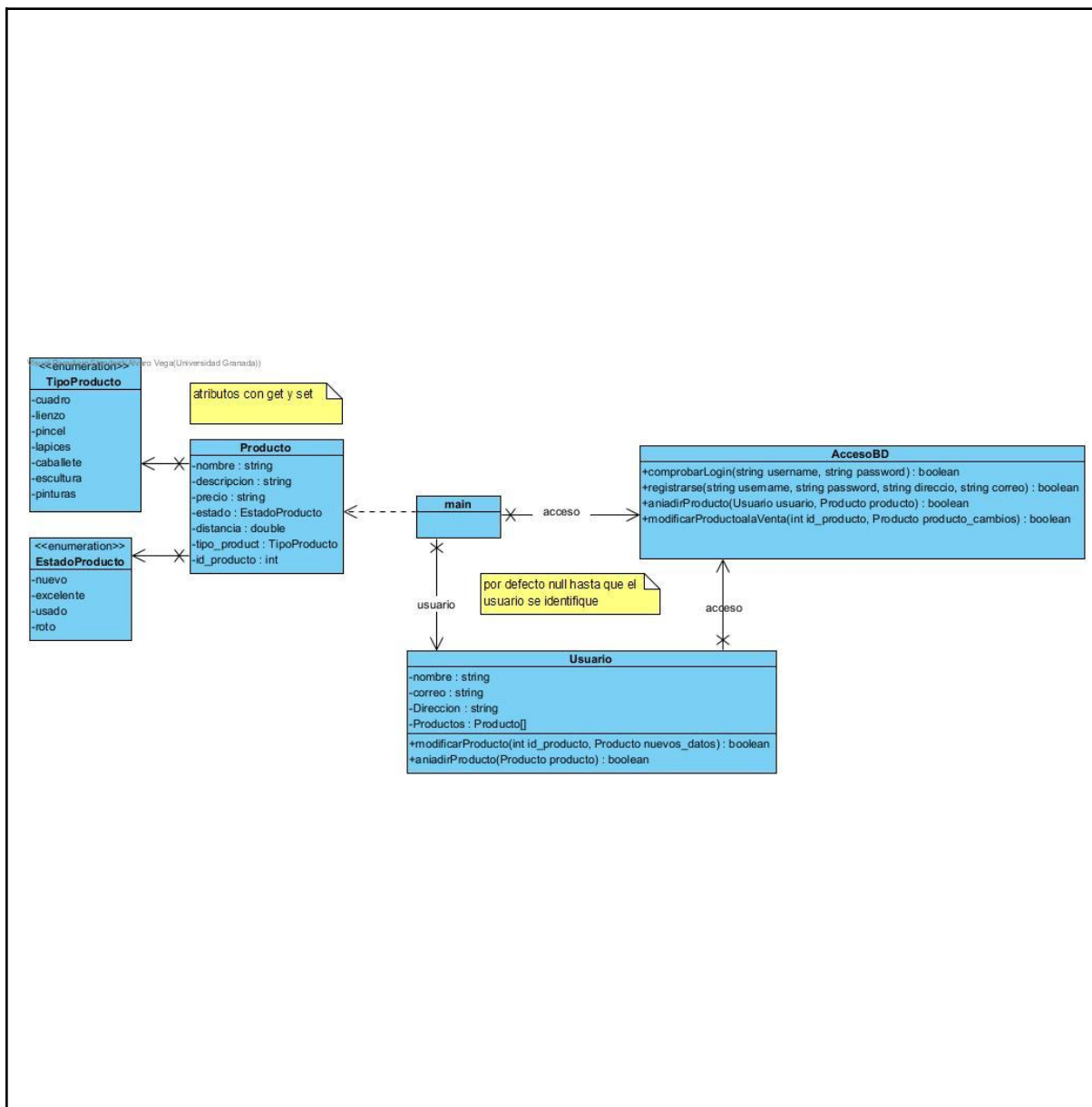
Planificación (decidir a grandes rasgos)		
¿Qué se quiere probar y cómo?	Se quieren probar los requisitos funcionales del sistema, cualquier interacción posible en busca de posibles errores. Se introducirán datos tanto correctos como erróneos esperando que el sistema detecte lo incorrecto y le pida al usuario cambiar los datos por correctos	
¿Quién lo hará y cuándo?	Normalmente sería un técnico de pruebas, pero en este caso lo haremos nosotros como grupo las pruebas pertinentes.	
¿Qué criterios se usarán para decidir cuándo terminar las pruebas?	Que los requisitos funcionales cumplan con sus funcionalidades básicas y puedan ejecutarse sin errores propios del sistema o errores de introducción de datos.	
Análisis (especificar las distintas condiciones de prueba)		
A partir de la especificación de requisitos funcionales/no funcionales de tu sistema, elige los elementos específicos cuyo comportamiento haya que probar, identificando las condiciones y datos requeridos para probarlos		
Elemento a probar	Condiciones	Datos Requeridos
Dar alta usuario	Introducción de datos de usuario de manera correcta e incorrecta.	Nombre, apellidos, email, teléfono, username, contraseña
Poner un producto a la venta	Introducción de los datos de productos. La inserción de tipos de datos específicos para cada campo	Nombre, precio, descripción, tipo, estado, distancia
Modificar un producto	Permite cambiar algún campo que se quiera modificar dentro de la tabla productos de un producto específico	Dato que se quiera modificar
Iniciar sesión como usuario	Introducción de datos para inicio de sesión. Se probará inserción correcta e incorrecta	Username y contraseña

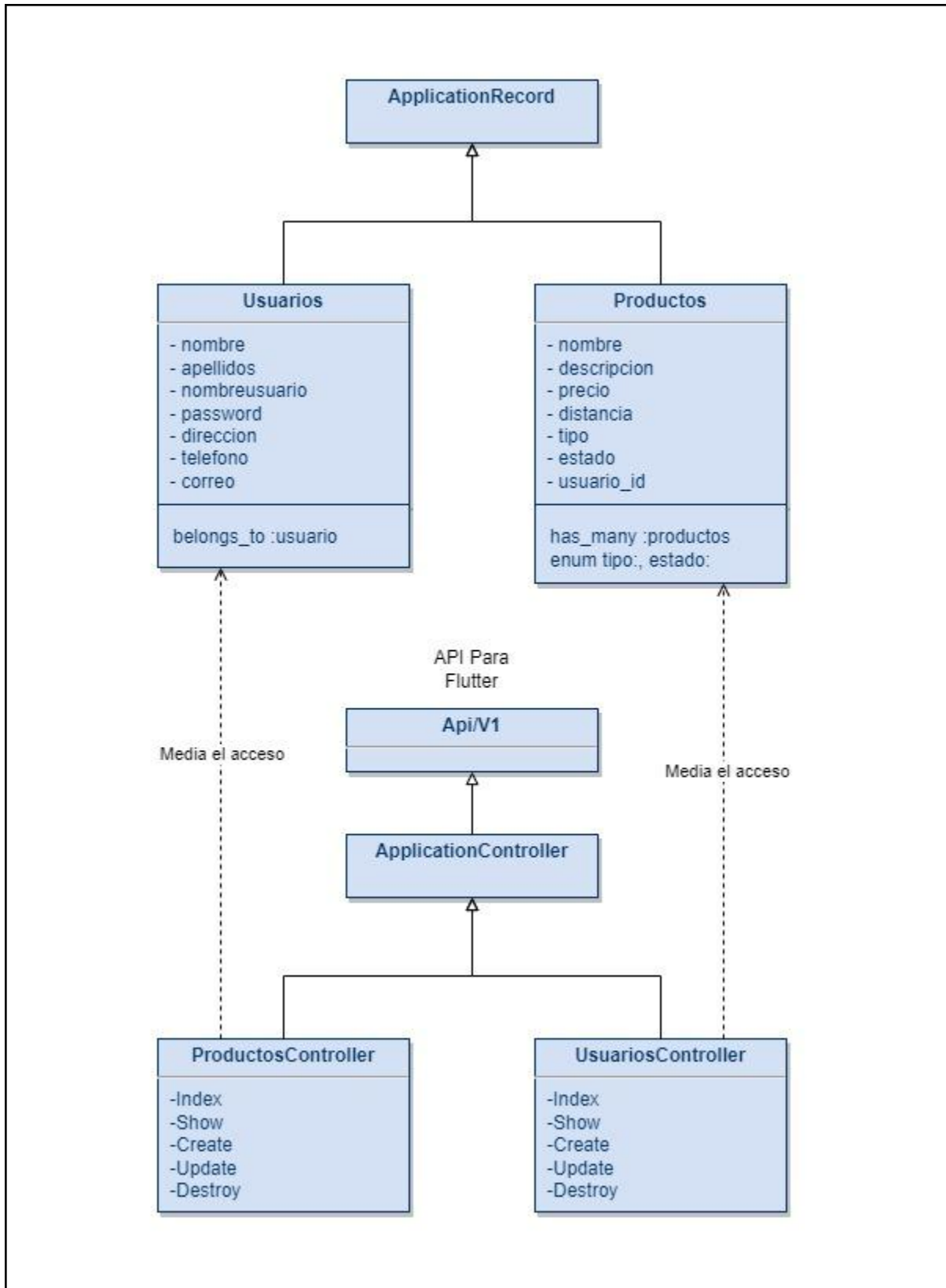
Modificar un usuario	Permite cambiar algún campo que se desee modificar	Datos que se busquen modificar
----------------------	--	--------------------------------

Parte de Diseño

7. Diagrama de clases

Nota: faltan algunos constructores triviales como crear objeto usuario con los 3 datos o crear un objeto accesoBD sin ningún parámetro.





8. Diseño de pruebas de unidad: condiciones de prueba y casos de prueba

Para este proyecto hemos ideado una serie de test que permitan probar el correcto funcionamiento de las distintas partes del sistema. Lo hemos dividido en cuatro grandes grupos:

Group 1 (Productos) - Test realizados sobre el modelo de Productos

- **Prueba 1.** Se creará un producto sin un propietario definido y se comprobará que no se puede crear debido a que el atributo *usuario_id* es una clave externa y necesariamente no puede ser nulo.
- **Prueba 2.** Se creará un producto con los campos tipo y estado en forma de "string" y se comprobará que se crea de forma satisfactoria.
- **Prueba 3.** Se creará un producto con los campos tipo y estado en forma de "int" y se comprobará que se crea de forma satisfactoria.

```
javier@DELL-JOL:~/DS/DS3_2$ rails test test/models/producto_test.rb
Run options: --seed 57375

# Running:

---

Finished in 1.085641s, 2.7633 runs/s, 2.7633 assertions/s.
3 runs, 3 assertions, 0 failures, 0 errors, 0 skips
javier@DELL-JOL:~/DS/DS3_2$
```

Group 2 (Usuarios) - Test realizados sobre el modelo de Usuarios

- **Prueba 1.** Se creará un usuario sin ningún producto asociado y se comprobará que se crea de forma satisfactoria
- **Prueba 2.** Se creará un usuario nuevo y se comprobará que la base de datos encripta la contraseña del usuario al guardarla.

```
javier@DELL-JOL:~/DS/DS3_2$ rails test test/models/usuario_test.rb
Run options: --seed 3396

# Running:

--

Finished in 0.911800s, 2.1935 runs/s, 2.1935 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
javier@DELL-JOL:~/DS/DS3_2$
```

Group 3 (web) - Test realizados de Integración

- **Prueba 1.** Se realizarán varias peticiones a la web para este test. Una a la pantalla de inicio de sesión, otra a la pantalla de *Usuarios* y otra a la pantalla de *Productos*.

```
javier@DELL-JOL:~/DS/DS3_2$ rails test test/integration/prueba_test.rb
Run options: --seed 48181

# Running:

---

Finished in 1.201408s, 2.4971 runs/s, 2.4971 assertions/s.
3 runs, 3 assertions, 0 failures, 0 errors, 0 skips
javier@DELL-JOL:~/DS/DS3_2$
```

Group 4 (Controladores) - Test realizados de los controladores de Productos y Usuarios

- **Prueba 1.** Se realizarán los test generados automáticamente por rails de manera que compruebe todas las acciones REST en el modelo de *Productos y Usuarios*.

```
javier@DELL-JOL:~/DS/DS3_2$ rails test test/controllers/productos_controller_test.rb
Run options: --seed 6336

# Running:

...F

Failure:
ProductosControllerTest#test_should_create_producto [/home/javier/DS/DS3_2/test/controllers/productos_controller_test.rb:19]:
"Producto.count" didn't change by 1.
Expected: 3
Actual: 2

rails test test/controllers/productos_controller_test.rb:18

.F

Failure:
ProductosControllerTest#test_should_update_producto [/home/javier/DS/DS3_2/test/controllers/productos_controller_test.rb:38]:
Expected response to be a <3XX: redirect>, but was a <422: Unprocessable Entity>

rails test test/controllers/productos_controller_test.rb:36

.:
Finished in 2.429762s, 2.8809 runs/s, 3.2925 assertions/s.
7 runs, 8 assertions, 2 failures, 0 errors, 0 skips
```

```
javier@DELL-JOL:~/DS/DS3_2$ rails test test/controllers/usuarios_controller_test.rb
Run options: --seed 48250

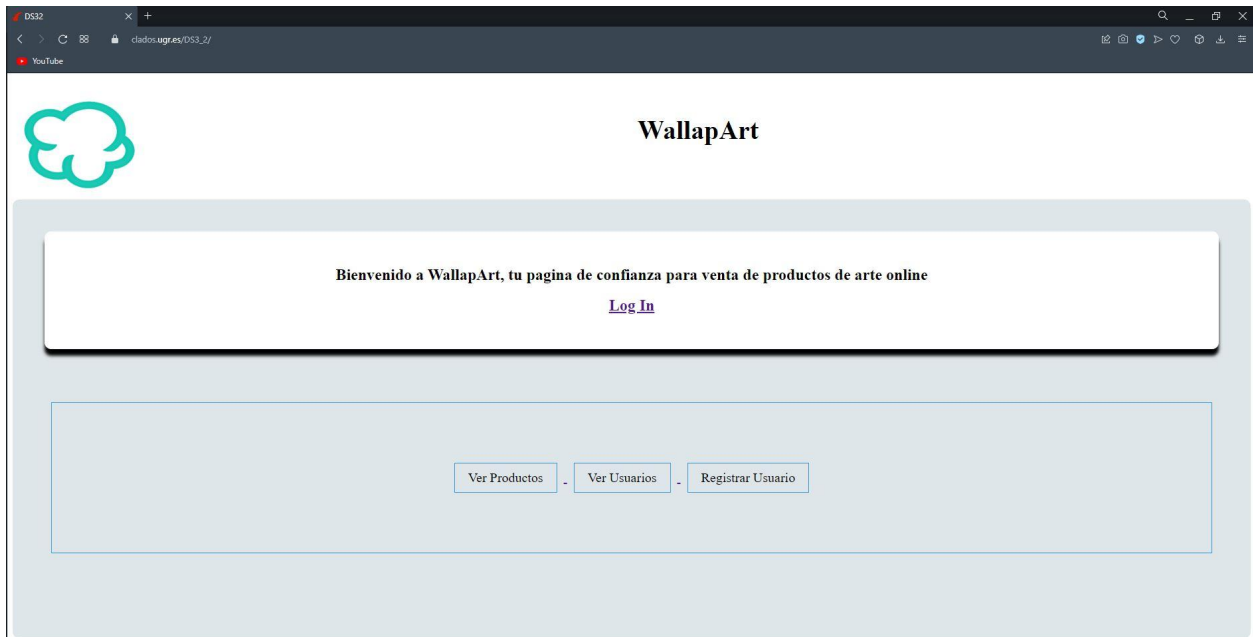
# Running:

-----

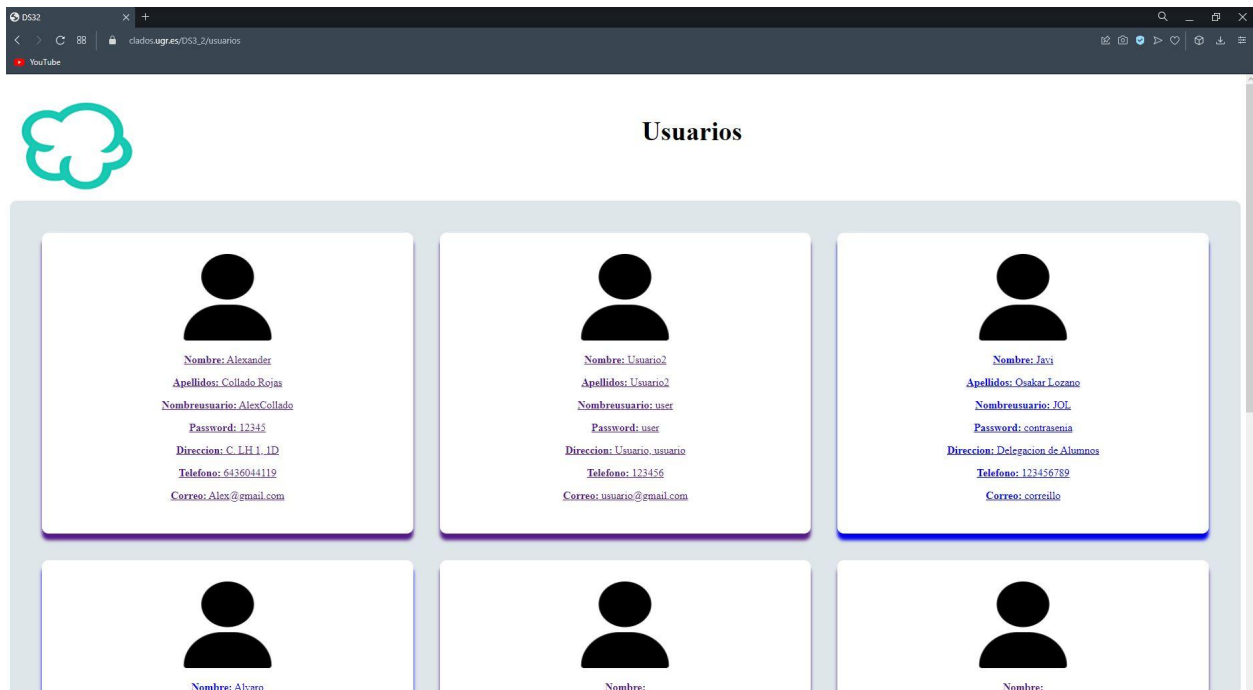
Finished in 2.357458s, 2.9693 runs/s, 3.8177 assertions/s.
7 runs, 9 assertions, 0 failures, 0 errors, 0 skips
javier@DELL-JOL:~/DS/DS3_2$
```

9. Capturas de Funcionamiento de la Web

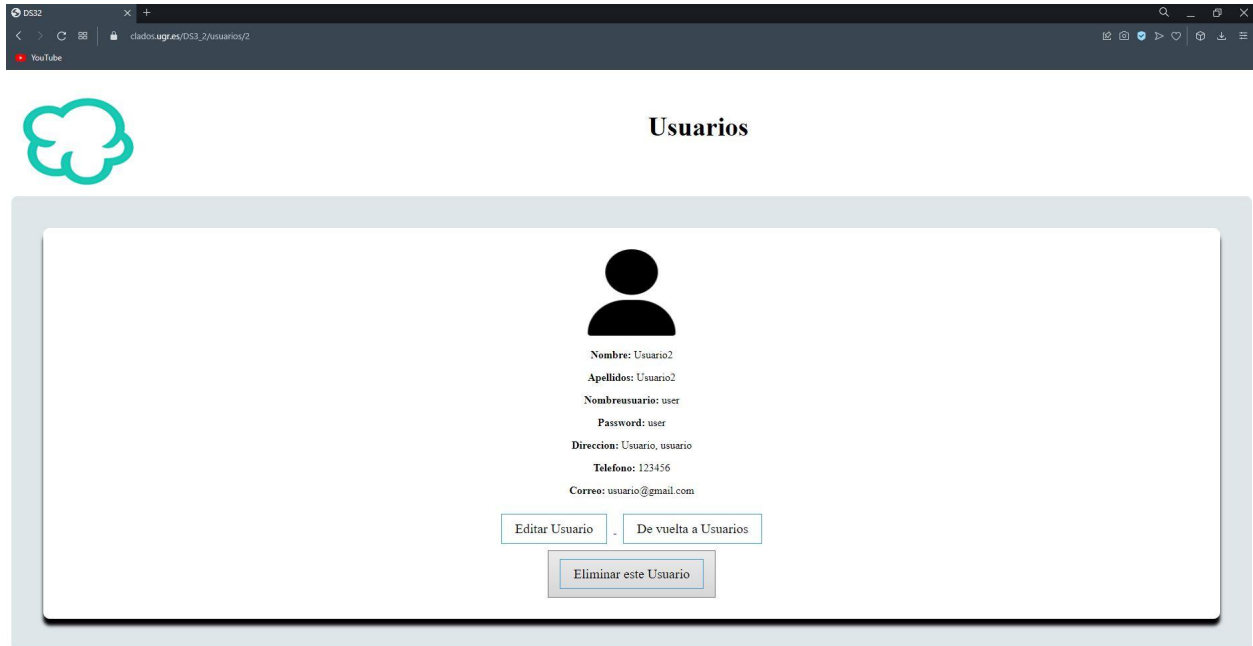
https://clados.ugr.es/DS3_2/



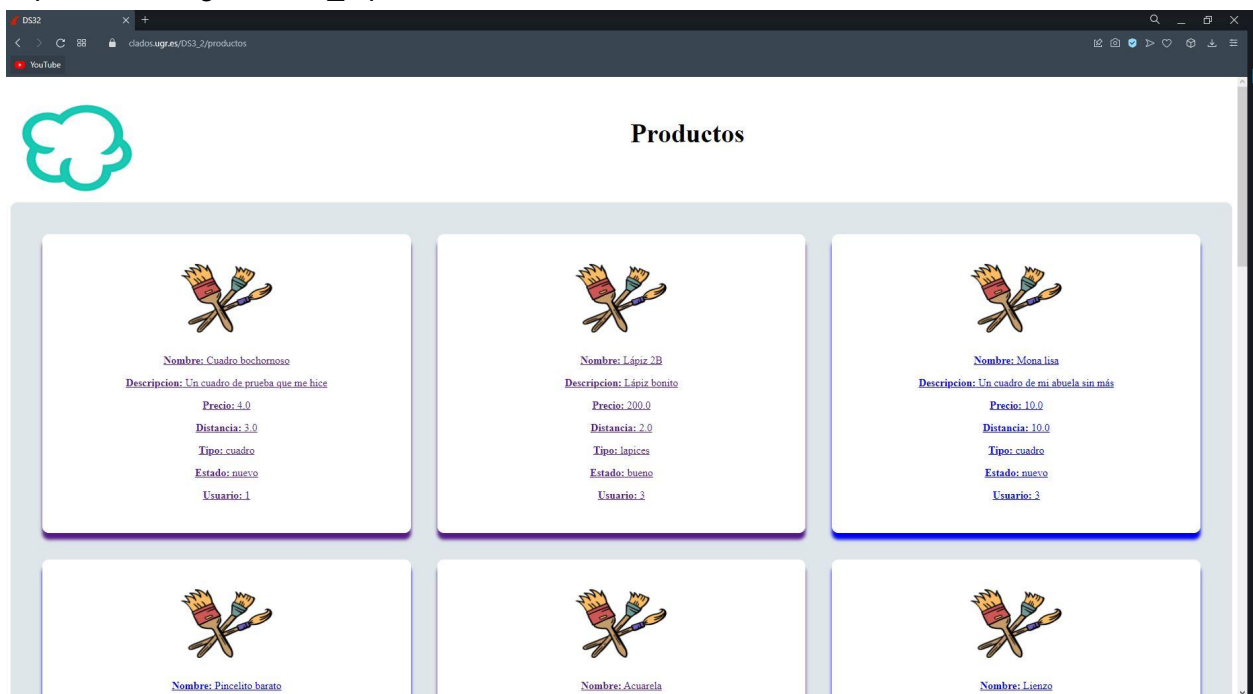
https://clados.ugr.es/DS3_2/usuarios



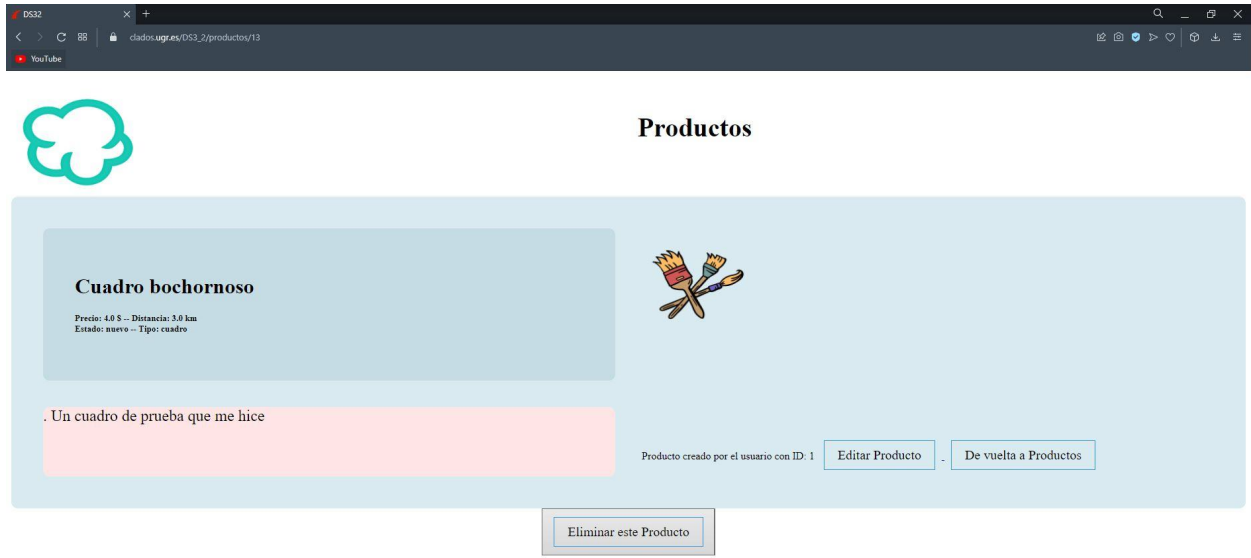
https://clados.ugr.es/DS3_2/usuarios/id



https://clados.ugr.es/DS3_2/productos



https://clados.ugr.es/DS3_2/productos/id



10. Posibles Ampliaciones/Mejoras

Debido a la falta de tiempo hay muchas funcionalidades que hubiera sido bueno mejorar en la parte de Ruby On Rails. Estas funcionalidades son:

- Mejorar de alguna forma los test para ver exactamente cual es el problema con la clave externa de **usuario_id**.
- Añadir un estilo CSS para los formularios.
- Todos los CSS se añaden al archivo de stylesheets.
- La funcionalidad del LogIn debe mejorar, ya que actualmente siempre manda el error de fallo en la autenticación. Esto en un proyecto local propio que tenemos si funcionaba pero al trasladarlo a clados hay un error.
- Faltaría la opción de **cerrar sesión**, está puesta como una ruta en la cual **logout** redirigirá al controlador de la sesión, específicamente al método **destroy** pero no se incluyó esto por problemas del inicio de sesión.

