

INGENIERIA WEB

PA\$AR€LA



Kiril Ventsislavov Gaydarov

Adrián Montoya Ros

Wing Kei Liang Tan

Antonio Rubio Martínez

Juan Carlos Maestre Martínez

Índice

1. Descripción general
 - a. Descripción técnica del sistema
2. Wireframes y mockups
 - a. Patrones de diseño
3. Diagramas
 - a. De datos
 - b. Casos de uso
 - c. De clases
4. Metodología y planificación
5. Descripción de la implementación
6. Problemas encontrados y solución
7. Mejoras y ampliaciones
8. Referencias

1. Descripción general

Nuestro proyecto se basa en el diseño de una plataforma de pagos (TPV) donde diferenciamos tres tipos de usuarios: los comercios, los técnicos y los administradores.

- Los comercios podrán usar nuestra web para que sus clientes realicen los pagos a través de él. También pueden ver las transacciones que sus clientes han realizado a través de nuestra plataforma de pagos.
- Si ocurriese algún problema, los comercios podrán pedirnos soporte a través de la web y los técnicos serán los encargados de resolver dichos problemas.
- El administrador será el encargado de dar de alta a los comercios para que puedan usar la plataforma de pagos.

a) Descripción técnica del sistema

Hemos utiliza laravel para la implementación de la plataforma de pagos ya que es un framework de creación de aplicaciones web open-source. El patrón de arquitectura aplicado es el de Modelo-Vista-Controlador. Utilizamos MySQL como sistema de gestión de bases de datos relacional para almacenar toda la información necesaria para el correcto funcionamiento de la web.

Para el cifrado de la información hemos empleado AES en modo CBC para cifrado simétrico, compartiendo la misma clave entre comercio y servidor, siendo esta introducida por el primero a la hora del registro. También se ha empleado SHA256 para el resumen de determinadas cadenas de texto, a modo de comprobación de que esa información no ha sido modificada.

La creación del proyecto se ha hecho usando Composer que es un gestor de dependencias en proyectos, para programación en PHP. Nos permite gestionar (declarar, descargar y mantener actualizados) los paquetes de software en los que se basa nuestro proyecto PHP.

Los comercios puedan emplear nuestra web debido a que nuestro sistema ha sido desplegado como servidor público. Disponemos de ip estática y dominio asociado, por lo que los comercios no se deben de preocupar en ningún momento por encontrarnos.

Debido a las características de Laravel, hemos detectado un problema en el intercambio de mensajes, ya que este realiza una validación CSRF que hemos tenido que desactivar nosotros en el startpoint y los comercios en el endpoint.

2. Wireframe y Mockups

Wireframe

Vista Invitado (#Guest)

- Home / Propaganda
- Inicio de sesión
 - o Salto a la vista del ClienteComercio
 - o Salto a la vista del TecnicoSoporte
 - o Salto a la vista del ADMIN

Vista Administrador (#ADMIN)

- Dashboard (Home)
- Cuentas (incluye Comercio y Técnico)
 - o Listado
 - o Buscador
 - o Detalle 'X' Cuenta
 - o Generar cuenta
- Tickets
 - o Listado
 - o Buscador
 - o Detalle 'X' Ticket
- Valoraciones
 - o Listado
 - o Detalle 'X' Valoración

Vista Comercio (#ClienteComercio)

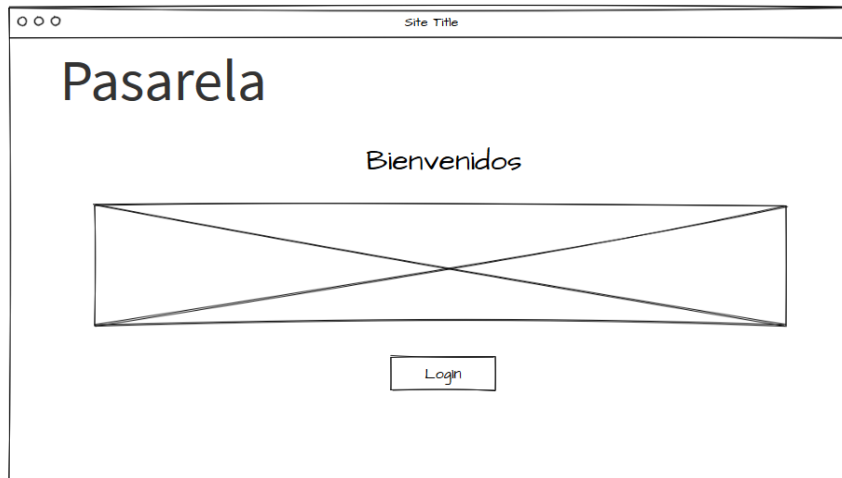
- Dashboard (Home)
- BackOffice
 - o General
 - o Buscador
 - o Detalle 'X' pedido
- Perfil del Comercio
- Tickets
 - o Listado
 - o Buscador
 - o Detalle 'X' Ticket
 - o Mis Valoraciones
 - > Perfil del Técnico de Soporte
 - > Valorar al Técnico de Soporte
- Wiki

Vista Soporte (#TecnicoSoporte)

- Dashboard (Home)
- Perfil del Soporte
- Tickets
 - o Listado
 - o Buscador
 - o Detalle 'X' Ticket
- Valoraciones sobre mi

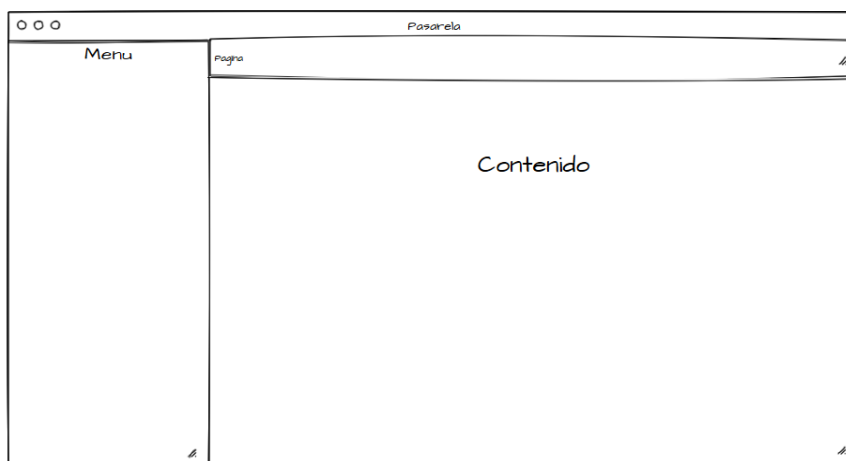
Mockups

- Página principal



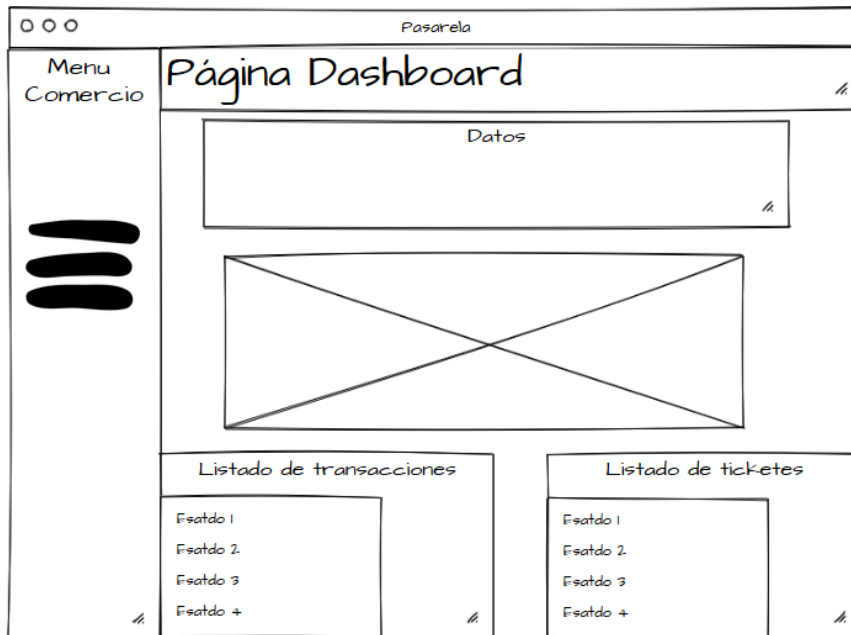
- Estructura de las vistas comercio, técnico y administrador

Todas las vistas tendrán la siguiente estructura. Un menú a la izquierda, el título de la página que estamos en la parte superior y debajo el contenido.

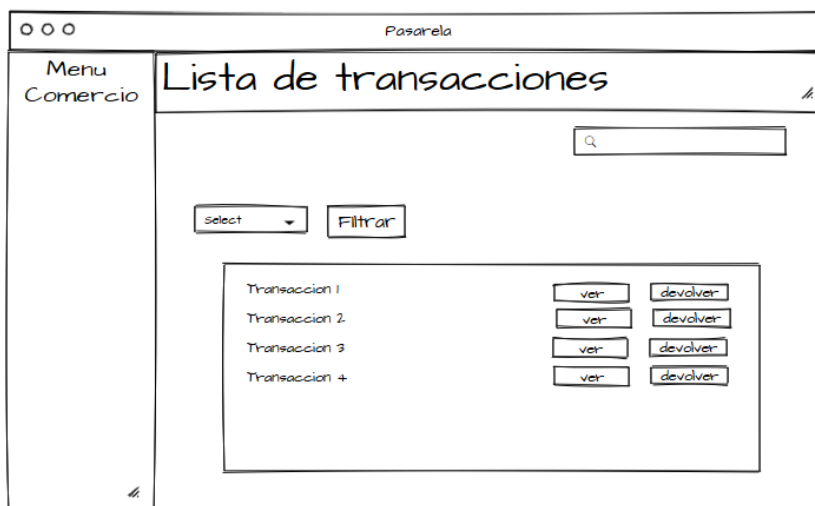


- Vistas de un comercio

El usuario que sea un comercio tendrá una vista general (dashboard) que mostrará un análisis de las transacciones que sus clientes han realizado y el número total de tickets que ha realizado. En el centro de la página estará situada una gráfica con el número de transacciones generadas en un determinado tiempo en concreto.



En dicha vista se mostrará todas las transacciones generadas con sus opciones de filtrado y búsqueda. El comercio también tendrá la posibilidad de hacer la devolución de una transacción



Además, cada comercio tendrá una lista de los tickets que haya abierto para ponerse en contacto con el servicio técnico.

ooo

Menú Tickets

Pasarela

Listado Tickets

Q Buscar...

ID	Información
1	...
2	...
3	...
...	...

←

1

2

3

4

5

6

7

8

9

→

También podrá ver los detalles de cada ticket así como enviar mensajes al técnico asignado a dicho ticket para que arregle su problema. Una vez se resuelva el ticket, el comercio podrá valorar la calidad del servicio técnico.

ooo

Menú Tickets

Pasarela

Detalles Ticket

ID	Información
1	...

Comercio: Juan

Técnico: Angel

Hola, queria saber si mi pago se ha realizado correctamente, ya que no me llega confirmacion del pedido

...

...

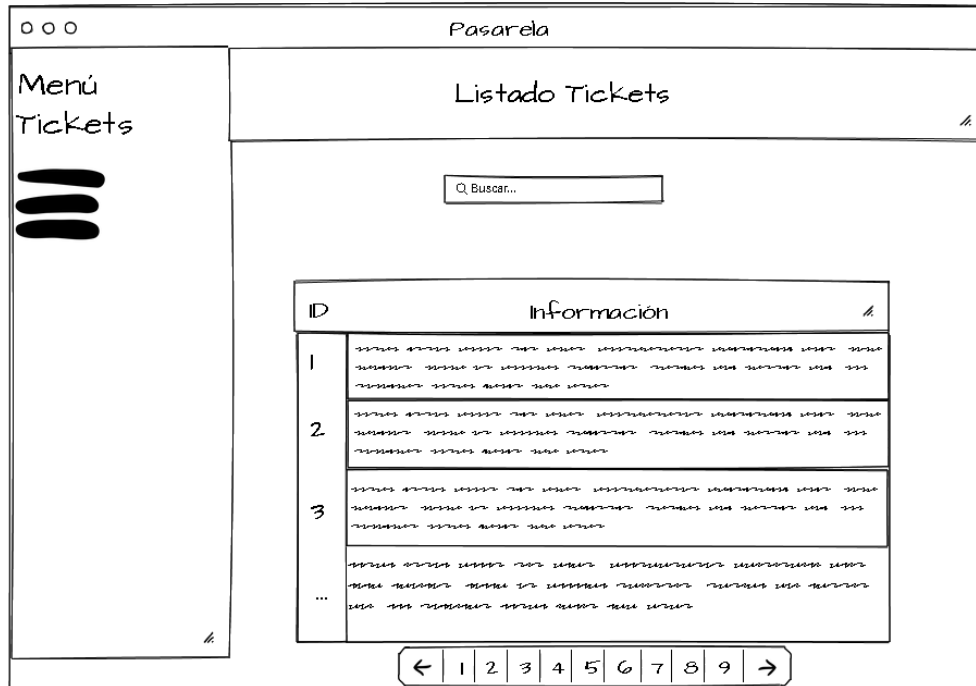
...

Como comercio podrá valorar a los técnicos que han atendido los tickets resueltos por incidencias, para poder almacenar la satisfacción del comercio, tiene la posibilidad de escribir una valoración hacia el técnico, añadiendo para este un comentario y una puntuación de 0 a 5. Las valoraciones se almacenarán y se podrán volver a consultar con el botón ver “valoraciones”, donde aparecerá la lista de detalles que hemos realizado a un determinado técnico con todos los detalles que se tiene de esta valoración.

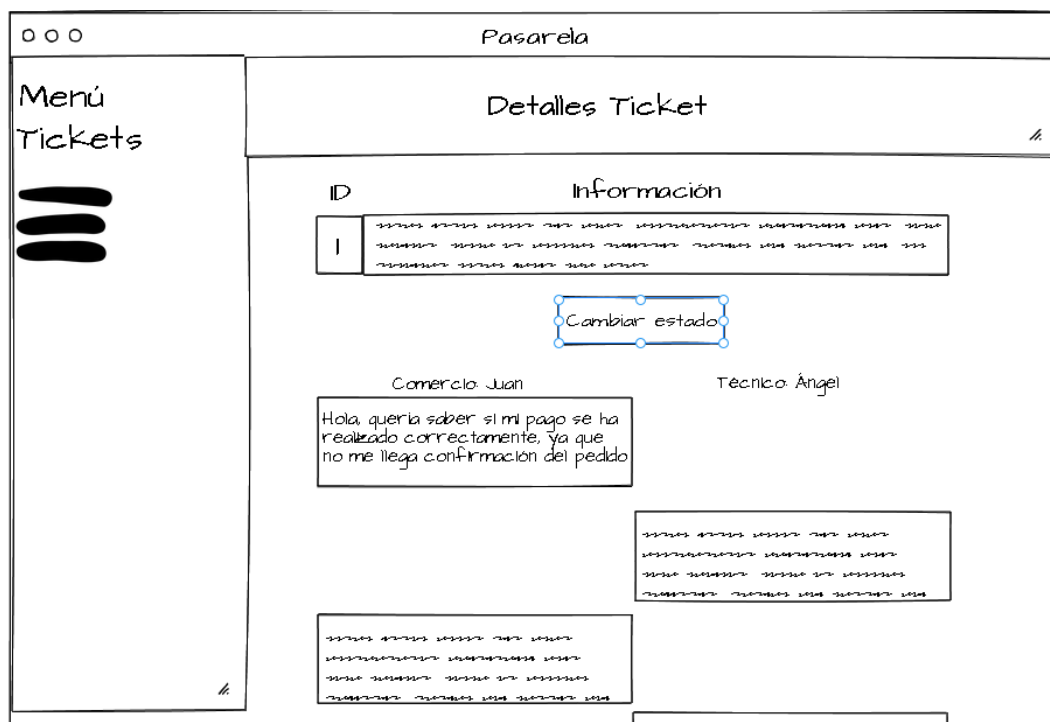
Pasarela	
Menú comercio	Listado Valoraciones
	<div><div>Nombre técnico: Juan Comentario: Trabaja bien ☆☆☆☆☆ <div>Valorar</div><div>Ver valoraciones</div></div><div>Nombre técnico: Juan Comentario: Trabaja bien ☆☆☆☆☆ <div>Valorar</div><div>Ver valoraciones</div></div></div>

- Vistas de un técnico

Igual que los comercios, podrá ver un listado de todos los tickets que deba atender.



En los detalles, además de enviar mensajes como los comercios, podrán cambiar el estado del ticket para notificar si ha sido aceptado o rechazado.

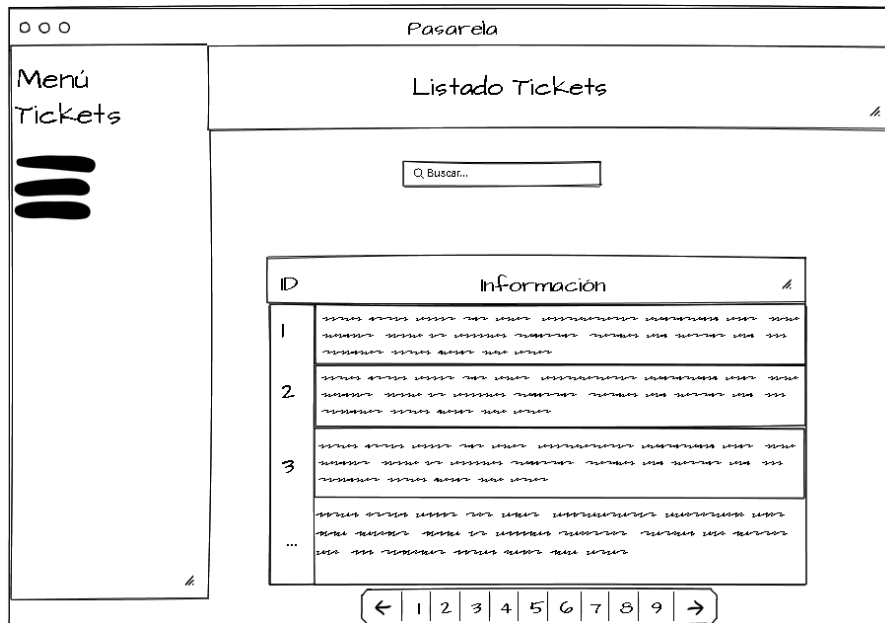


Desde el perfil de técnico podremos ver las valoraciones que nos han realizado los diferentes comercios, además de mostrarnos quien nos ha hecho la valoración con su comentario y su puntuación desde el botón de “ver detalles”.

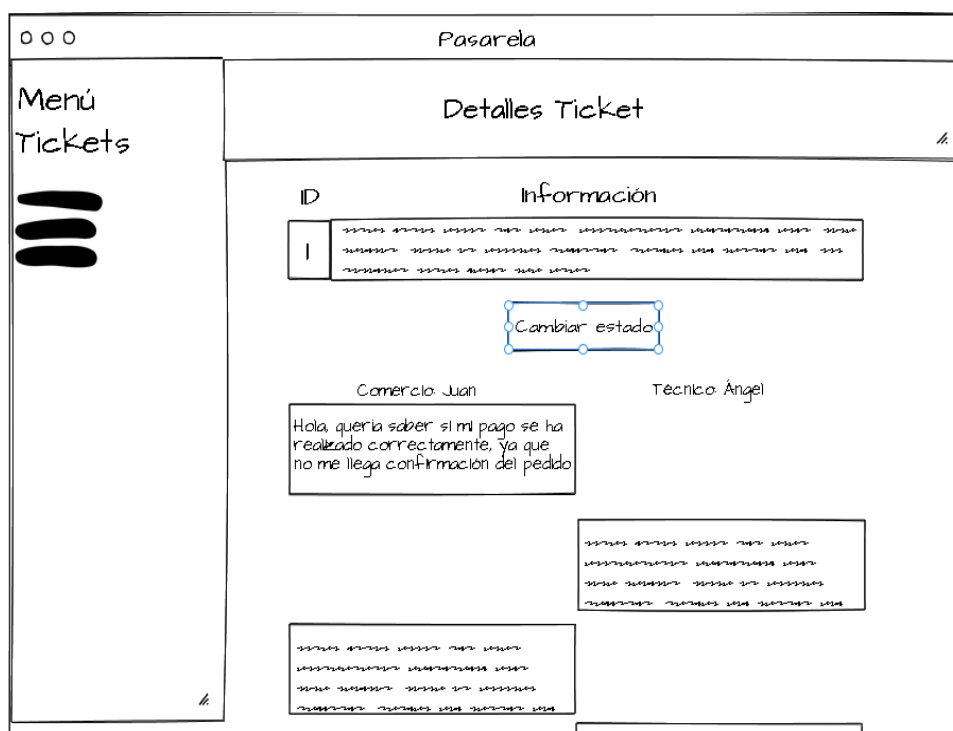
<div>ooo</div> <div>Pasarela</div>	
<div>Menú técnico</div>	<div>Listado Valoraciones</div>
	<div><div><div>Nombre técnico: Juan</div><div>Comentario: Trabaja bien</div><div>☆☆☆☆☆</div><div>Ver detalles</div></div><div><div>Nombre técnico: Juan</div><div>Comentario: Trabaja bien</div><div>☆☆☆☆☆</div><div>Ver detalles</div></div></div>

- Vistas de un administrador

Tendrá acceso a un listado de todos los tickets, independientemente de su estado.



En los detalles, no podrá enviar mensajes, pero sí observarlos, y podrá modificar el estado del ticket.



En la vista de de las valoraciones vista desde un menú administrador podremos encontrar la lista de comentarios que cualquier comercio hacia realizado a los diferentes técnicos que hay, en cada una de las valoraciones podremos tanto borrar la valoración como ver los detalles de esta.

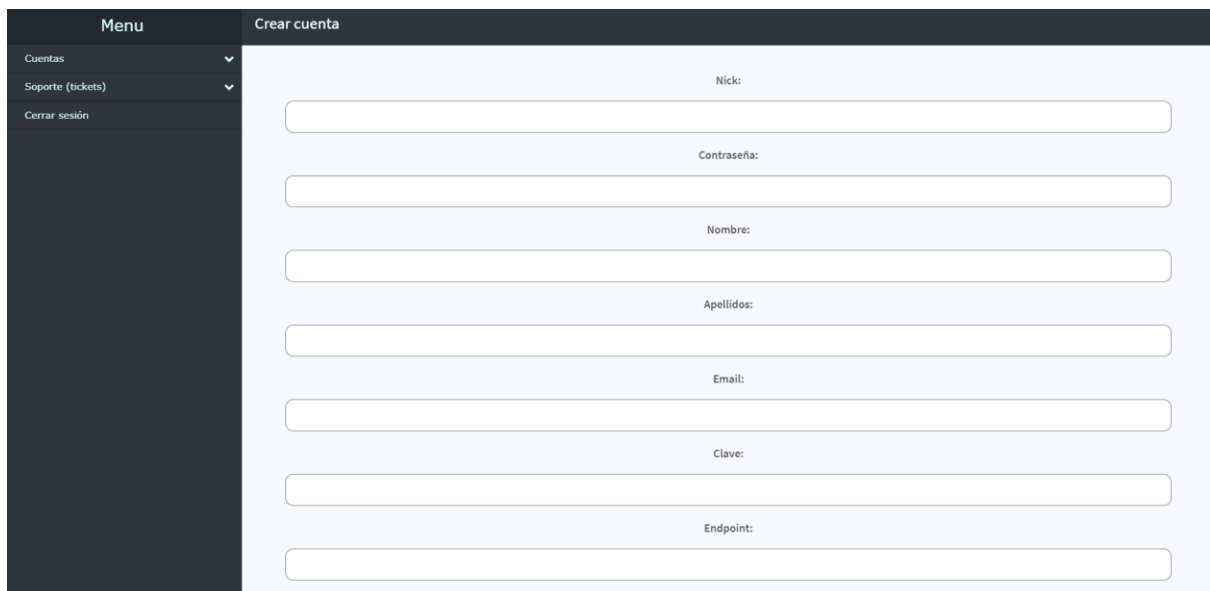


2. a Patrones de diseño

En este apartado describiremos los diferentes patrones de diseño aplicados en nuestra página web con una pequeña descripción de cada uno de ellos además de describir el tipo de patrón del que se trata.

-Patrón de diseño de identificación de usuarios: Registro de usuarios

Aunque el registro de usuarios se realiza unicamente por parte del usuario administrador, es decir, es este perfil el que se encarga de dar de alta los nuevos usuarios ya sean de comercio, técnicos o nuevos administradores.



The image shows a web interface for creating a new account. On the left is a dark sidebar menu with the title 'Menu' and three items: 'Cuentas', 'Soporte (tickets)', and 'Cerrar sesión'. The main area is titled 'Crear cuenta' and contains a form with the following fields: Nick, Contraseña, Nombre, Apellidos, Email, Clave, and Endpoint. Each field is represented by a text label above a white input box with a thin border.

Menu	Crear cuenta
Cuentas	Nick:
Soporte (tickets)	Contraseña:
Cerrar sesión	Nombre:
	Apellidos:
	Email:
	Clave:
	Endpoint:

-Patrón de diseño de identificación de usuarios: Log in

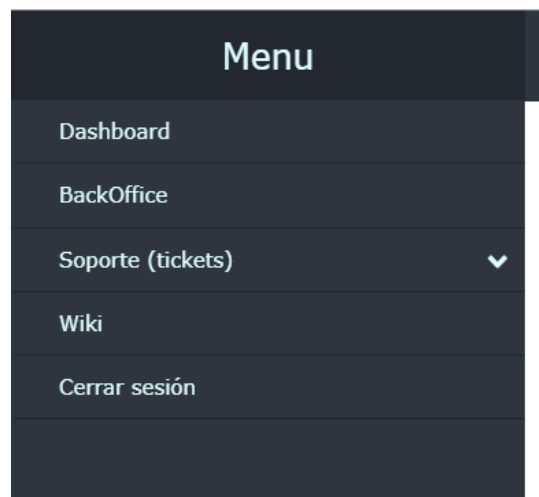
Hemos implementado este patrón para proporcionar un mecanismo de identificación para los usuarios, para que estos puedan acceder a zonas privadas o personalizadas. Este también ofrece la posibilidad de recordar las credenciales mediante el uso de correo además de la utilización de la opción no cerrar sesión que evitará el log out en caso de cerrar la ventana.



The image shows a login form for a website called "Pa\$arola". The logo is at the top left, with the letters "Pa" in green, "\$" in yellow, "ar" in green, and "ola" in yellow. To the right of the logo is a link that says "Iniciar sesión". Below the logo is a dark blue rectangular area containing a white login form. The form has a title "Iniciar sesión" at the top. It contains two input fields: "Correo electrónico" and "Contraseña". Below the "Contraseña" field is a checkbox labeled "No cerrar sesión". At the bottom of the form is a blue button labeled "Entrar". Below the button is a link that says "Has olvidado tu contraseña?".

-Patrón de diseño de identificación de usuarios: Log Out

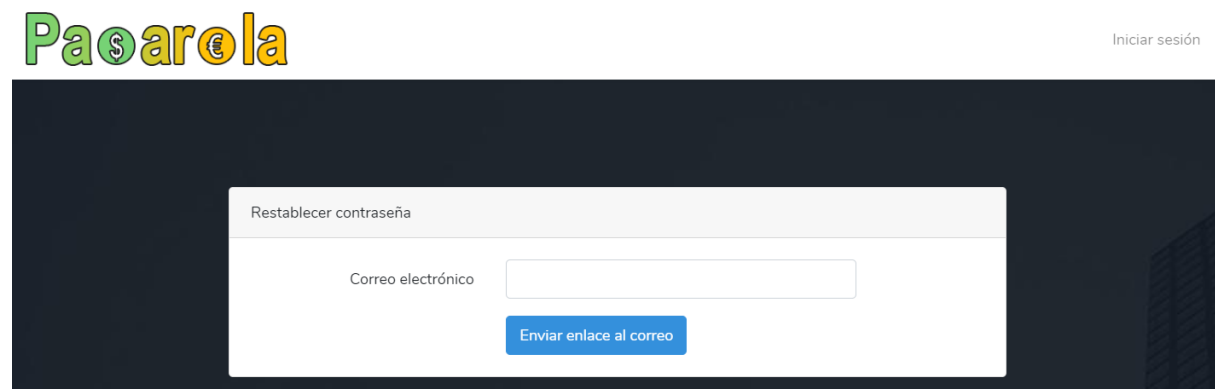
Este patrón necesario lo hemos incorporado al menú que nos seguirá en todo momento de nuestra sesión, por lo tanto, en cualquier instante podremos cerrar la sesión presionando dicho botón.



The image shows a dark blue menu with a white title "Menu" at the top. Below the title are five menu items: "Dashboard", "BackOffice", "Soporte (tickets)", "Wiki", and "Cerrar sesión". The "Soporte (tickets)" item has a small white downward arrow next to it. The "Cerrar sesión" item is the last item in the menu.

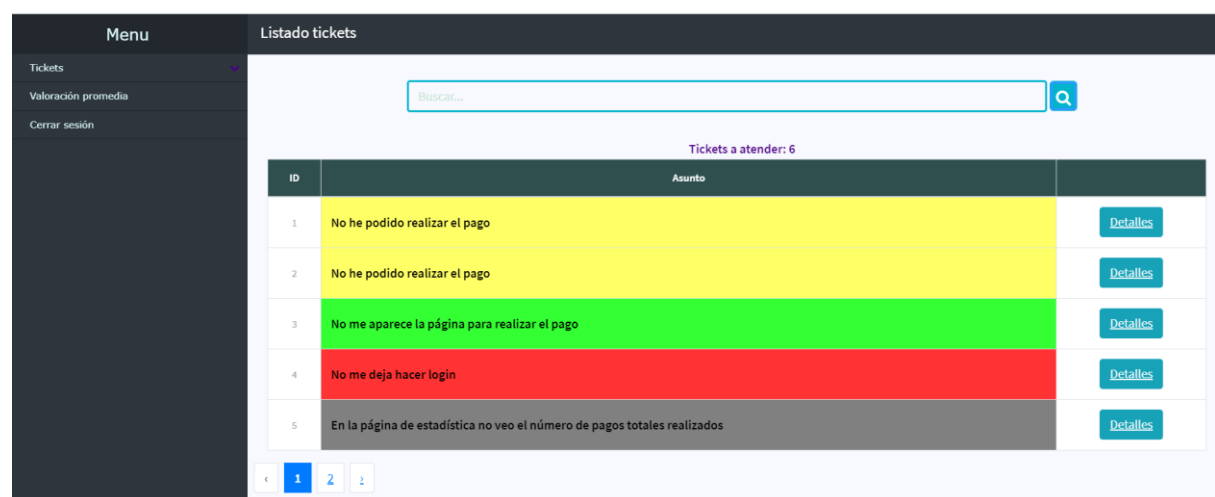
-Patrón de diseño de identificación de usuarios: Pérdida de contraseña o de cuenta

Este patrón es muy útil para evitar que el usuario administrador reciba innumerables peticiones de nuevas contraseñas o nuevos perfiles por la pérdida de credenciales, para ello disponemos en el login una opción de un enlace “¿Has olvidado la contraseña?” que nos redirige a una pantalla en la que podemos recuperar la contraseña haciendo uso del correo electrónico.



-Patrón de página principal: Bandeja de entrada

Cada usuario tiene su propia bandeja de entrada con la que se encontrará nada más loguearse, en el caso del técnico podemos ver como en su bandeja de entrada puede encontrar los tickets que debe consultar para solucionar y ponerse en contacto con los comercios.



-Patrón de página principal: Panel de control

El usuario comercio tiene la posibilidad de utilizar un backoffice donde podrá devolver o consultar los pagos realizados o incluso buscarlos mediante filtros.

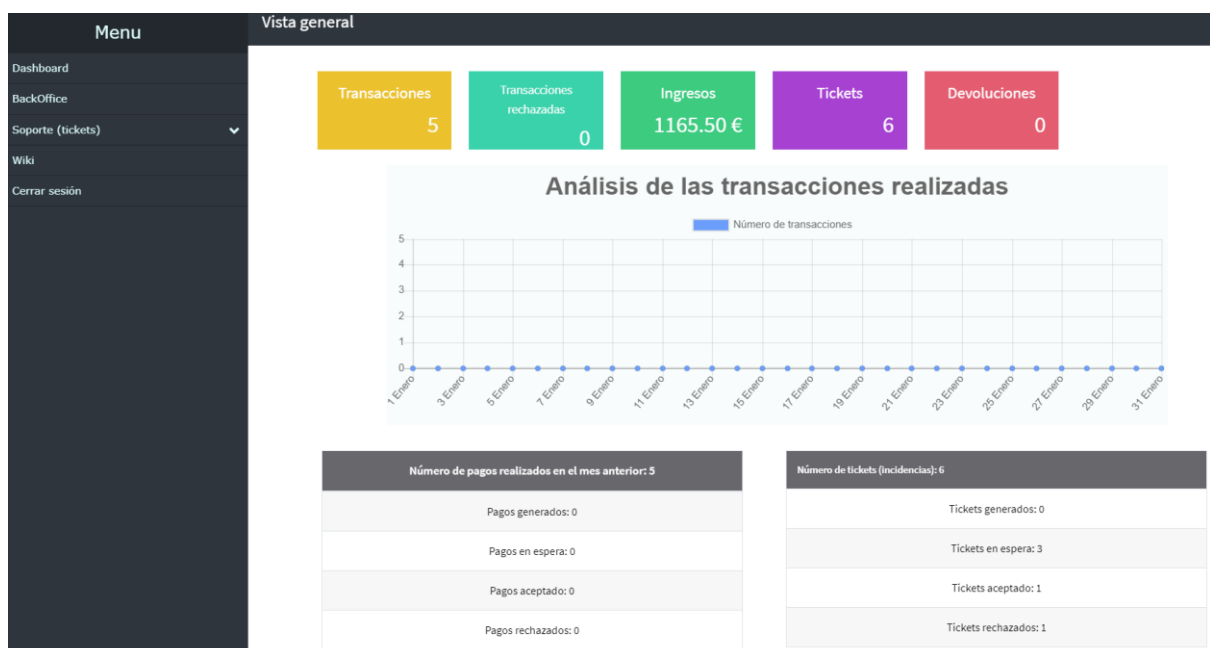
Menu		Pagos				Search	SEARCH
Dashboard							
BackOffice							
Soporte (tickets)							
Wiki							
Cerrar sesión							

Todos los estados	Todos los importes	FILTRAR
-------------------	--------------------	---------

ID	Importe	Estado	Tarjeta	Comentario	Accion	
1	520.50 €	generado	**** * 1240	Venta de un portátil hp	Detalles	Devolver
2	22.50 €	generado	**** * 4871	Venta de unos auriculares	Detalles	Devolver
3	22.50 €	espera	**** * 4871	Venta de una funda para ipad	Detalles	Devolver
4	500.00 €	espera	**** * 4440	Venta de un iphone 6	Detalles	Devolver
5	100.00 €	generado			Detalles	Devolver

-Patrón de página principal: Dashboard

La implementación de este patrón en nuestro sitio web proporciona al usuario comercio una vista global de las transacciones, tickets, ingresos y devoluciones que se han recopilado en su cuenta junto con gráficas que muestran un estudio de las transacciones realizadas.



-Patrón de página principal: Personalización

Como ya hemos mostrado en varias capturas anteriores cada usuario puede pertenecer a un perfil: comercio, técnico o administrador. Cada uno de estos únicamente comparte la vista de login y el estilo de la web, todas las demás pantallas incluso cuando muestran los mismos objetos lo hacen de forma diferente adaptada a las necesidades y funciones de cada perfil. En la siguiente captura mostramos una nueva bandeja de entrada, con la que el administrador podrá hacer operaciones con los usuarios.

Menu

Cuentas

Soporte (tickets)

Cerrar sesión

Listado cuentas

Buscar...

q

ID	Comercio	Nombre	Apellidos	Correo electrónico	Tipo de usuario	Acciones
1	Antoni	Antonio	García Martínez	antonio1@gmail.com	administrador	<div>Editar</div> <div>Borrar</div>
2	Jose1	Jose	Martínez Lopez	jose1@gmail.com	comercio	<div>Editar</div> <div>Borrar</div>
3	Fran	Francisco	Lopez Sanchez	francisco1@gmail.com	técnico	<div>Editar</div> <div>Borrar</div>
4	Juanito el grande 2	Juan chaparro	Gonzalez Sanchez Gonzalez otra vez	juanito12@gmail.com	técnico	<div>Editar</div> <div>Borrar</div>
5	Isa	Isabel	Jimenez Moreno	isabella@gmail.es	comercio	<div>Editar</div> <div>Borrar</div>

<

1

2

3

4

-Patrón de formularios: Formularios breves y Agrupar campos

Poseemos distintos formularios para crear tickets, añadir valoraciones o crear usuarios, entre ellos el más largo es el de la creación de usuarios ya que se necesitan más datos para almacenar, pero en relación a todos los demás hemos utilizado formularios cortos y concretos haciendo uso de listas desplegables o guardando las entradas escogidas para que el usuario tenga que teclear lo mínimo posible. En lo referente al patrón de agrupar campos, en cada página de formularios tenemos una pantalla limpia con únicamente el contenido necesario y correspondiente a cada atributo de la clase que vamos a crear y almacenar.

Crear ticket

Asunto...

Transacción (seleccione una):

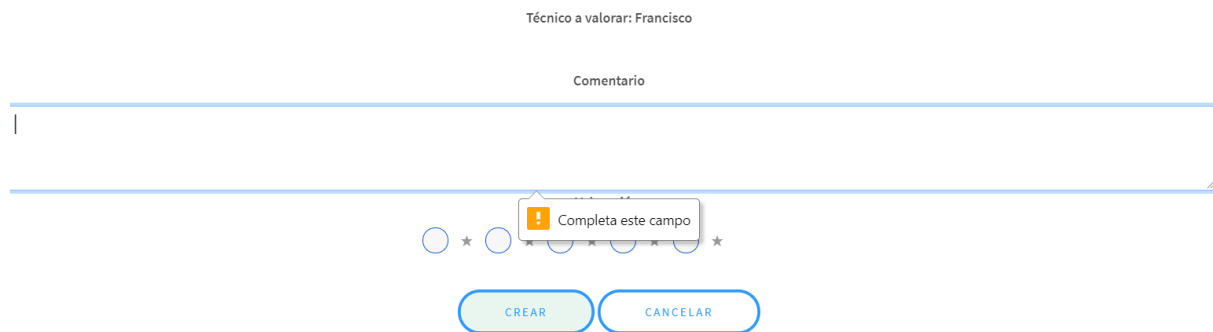
Selecciona una opción

Escribe la incidencia...

ENVIAR

-Patrón de formularios: Alineado de las etiquetas y los campos

A pesar de los problemas obtenidos por la combinación de los archivos css hemos podido diseñar unos alineamientos correctos para la organización y la buena visibilidad de los campos, utilizaremos la imagen de abajo para los siguientes patrones usados.



Técnico a valorar: Francisco

Comentario

|

Completar este campo

CREAR CANCELAR

Detailed description: This image shows a web form with a title 'Técnico a valorar: Francisco'. Below it is a label 'Comentario' followed by a large text input field. A cursor is visible at the start of the input field. Below the input field is a row of seven circular progress indicators, each followed by a star icon. The third indicator from the left is highlighted with a tooltip that says 'Completar este campo'. At the bottom of the form are two buttons: 'CREAR' (green) and 'CANCELAR' (blue).

-Patrón de formularios: Señalizar campos obligatorios

Como podemos apreciar en la imagen de arriba, en caso de que no se complete los campos principales del formulario aparecerá un mensaje indicando la obligación de rellenar los campos indicados.

-Patrón de formularios: Valores por defecto inteligentes

En la imagen superior también podemos apreciar después de presionar el botón de “añadir comentario” que nos lleva a la página de la captura donde se muestra ya el técnico a valorar, para que no tenga que volver a seleccionarlo el usuario comercio.

-Patrón de formularios: Formato de los datos

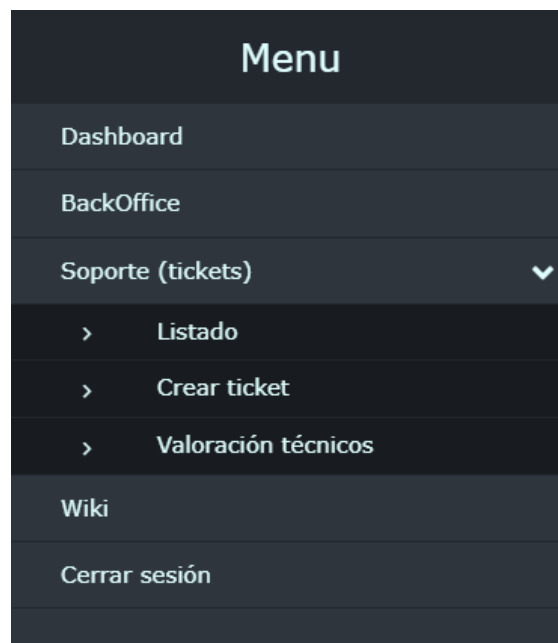
Para atender correctamente cualquier petición en la captura anterior y en la mayoría de nuestros formularios hemos permitido la entrada de datos de cualquier tipo para que el usuario pueda expresar el problema o cualquier situación con todos los detalles que pueda proporcionar.

-Patrón de formularios: Navegación con el teclado

Con la finalidad de poder hacer cómoda la tarea de rellenar formularios, hemos probado que puedan ser rellenados haciendo uso del teclado incluso cuando nos encontramos con campos ocultos.

-Patrón de navegación: Menú de primer, segundo nivel y rastro de migas de pan

En el menú con el que podremos interactuar en cualquier momento con la situación de del usuario podemos encontrar este patrón de navegación que incluye un menú con submenús para que se pueda tener localizado en todo momento el lugar y las opciones de navegación que tiene un usuario, con esta funcionalidad incorporamos el patrón de rastro de migas de pan, patrón utilizado para rastrear en todo momento la parte de la página en la que nos encontramos y cómo volver atrás.



-Patrón de formularios: Botones de comando

Para que no haya pérdida del usuario entre los distintos campos, hemos incorporado grandes botones con colores señalizados para que no haya ningún problema a la hora de localizarlos cuando se quiera realizar alguna operación.

Asunto...

Transacción (seleccione una):

Selecciona una opción

Escribe la incidencia...

ENVIAR

-Patrón de formularios: Mensajes de error

Como en el formulario de valoración, también en el formulario de creación de ticket podemos encontrar con mensajes de error a la hora de enviar campos necesarios sin completar.

Asunto...

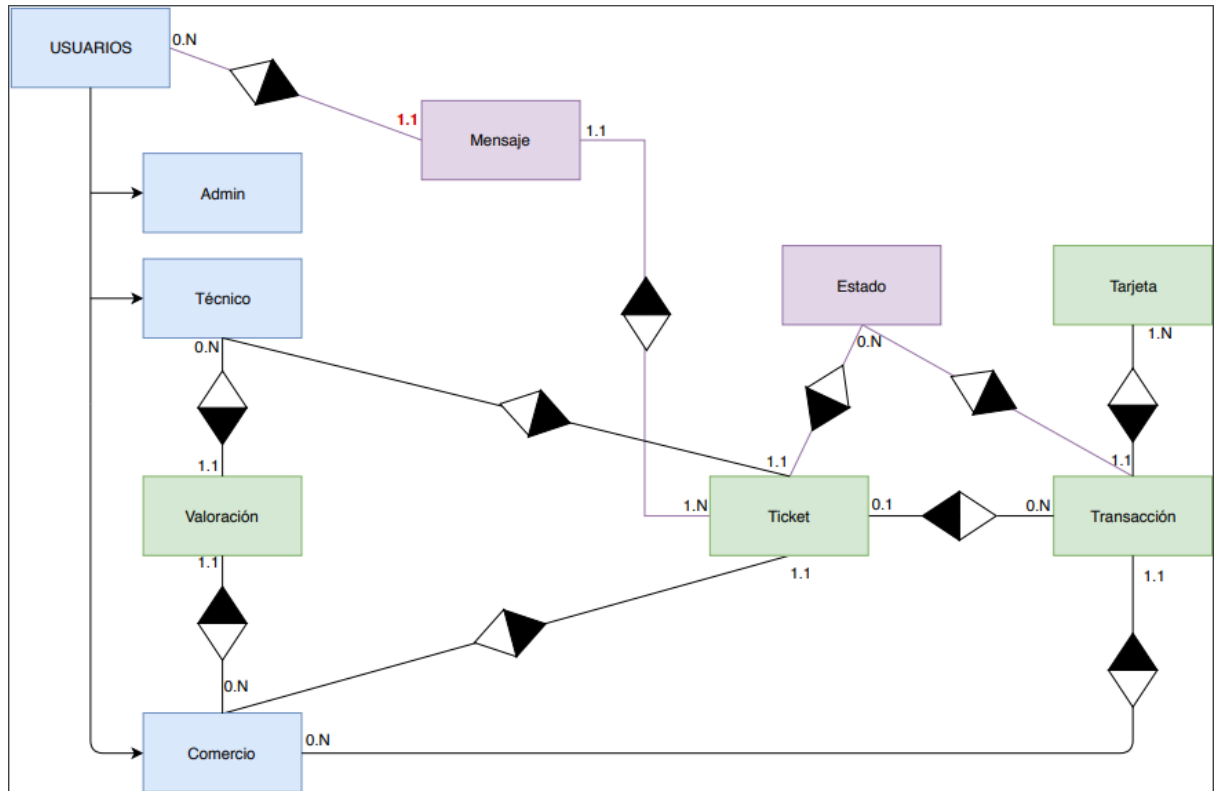
Completa este campo

Selecciona una opción

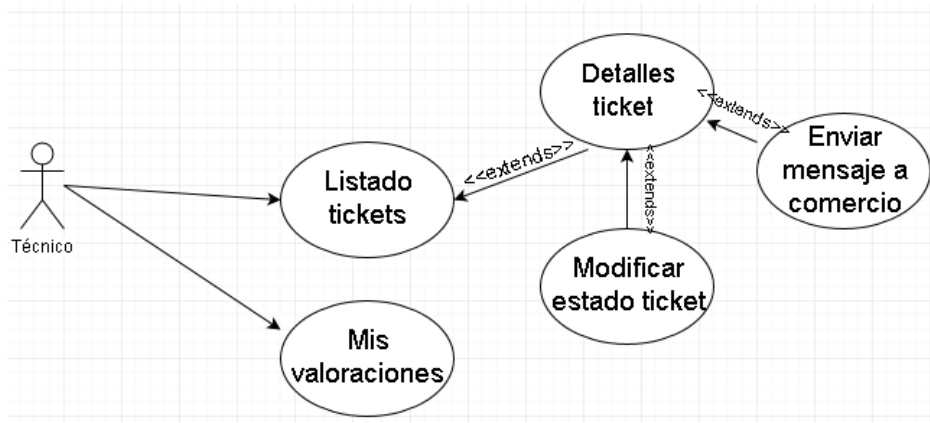
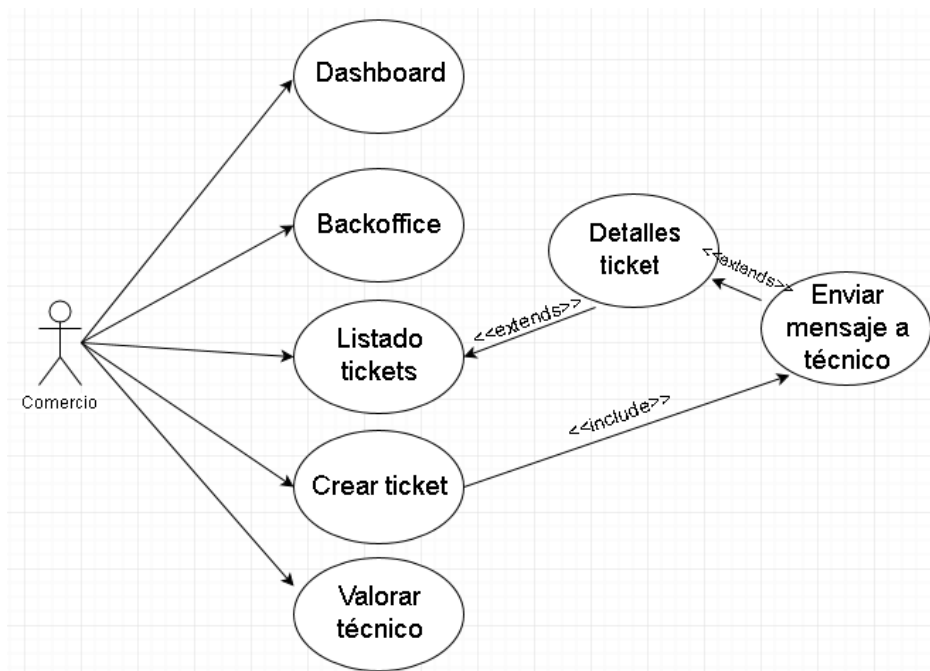
Escribe la incidencia...

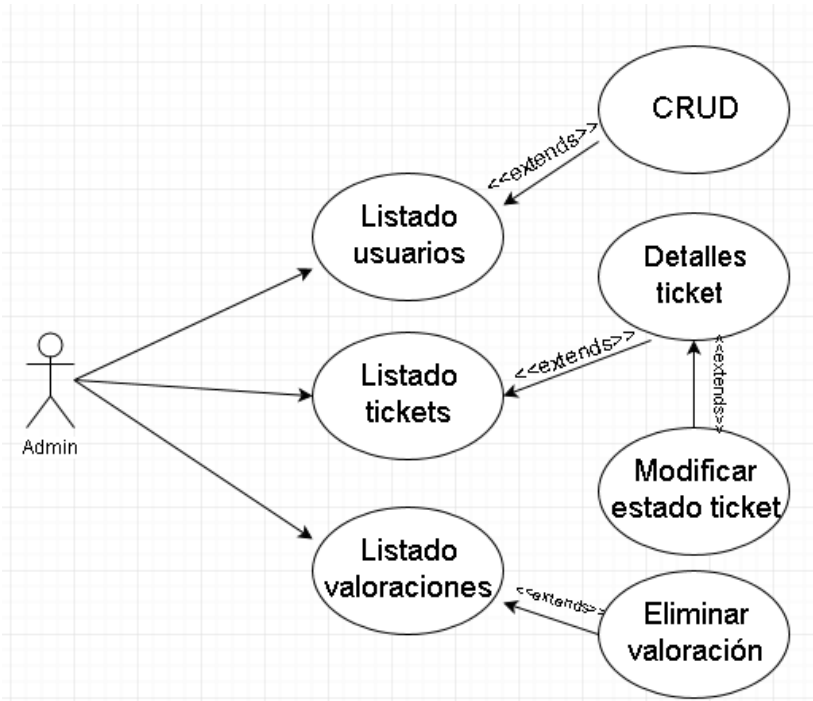
ENVIAR

- Diagrama de clases de la plataforma de pagos



- Diagrama de casos de uso:





4. Metodología y planificación

Para desarrollar la aplicación hemos utilizado una variedad de software. A continuación lo detallamos:

Software empleado:

- **Telegram:** hemos empleado esta herramienta como principal canal de comunicación entre los miembros del equipo. La ventaja de utilizar Telegram reside en que es una app multiplataforma y esto permite tenerla en cualquier dispositivo, sea móvil, portátil, ordenador sobremesa o tablet. De esta forma podemos estar comunicando entre nosotros desde cualquier sitio y en cualquier momento.
- **Discord:** complementa a Telegram para añadir funcionalidades que la primera no tiene. Permite conversaciones en directo entre todos los miembros del equipo con una calidad excelente. Además, permite compartir las pantallas de cada uno de los usuarios. Esta funcionalidad es la más valiosa, ya que permite resolver dudas, comentar nuevas funciones y tomar decisiones sobre características del proyecto.
- **Visual Studio Code:** editor de texto enriquecido de Microsoft. Se emplea para escribir todo el código que genera la página web.
- **NeoRouter:** crea una red VPN entre todos los miembros del equipo. Permite compartir la carpeta del proyecto y que todos tengan acceso a ella desde cualquier ordenador distinto del host. Permite el control remoto de los equipos si se asignan los permisos pertinentes y tener actualizado el deploy con los últimos cambios.
- **XAMPP:** conjunto de herramientas en un único paquete. Dispone de servicios MySQL, PHP, FTP, Apache entre otras. Compatible con la mayoría de sistemas operativos actuales. De todo el conjunto utilizaremos únicamente Apache para realizar el despliegue de la aplicación.
- **MySQL Server 5.7:** servidor de bases de datos MySQL que utilizamos para almacenar todos los datos de la aplicación. Datos como las transacciones, los tickets, las valoraciones, los usuarios, etc. Se emplea la versión 5.7 y no la última, que es la 8, porque ésta es la más compatible con Laravel.
- **MySQL Workbench:** cliente MySQL que se emplea para conectar con el servidor de la base de datos tanto de forma remota como de forma local. Permite gestionar

la base de datos entera, asignar permisos a los usuarios, crear usuarios, borrar usuarios, crear bases de datos, editar las tablas de las bases de datos y un sin fin de opciones más.

- **Team Viewer:** permite control remoto desde cualquier dispositivo y sitio. Se utiliza para controlar los servidores web y de bases de datos de la aplicación.
- **GitHub Desktop:** permite sincronizar el trabajo realizado con el sistema de control de versionado Git.
- **Trello:** Gestor de tareas para realizar una división lo más equitativa posible, además de ofrecer información útil para el proyecto, como por ejemplo ver el avance actual del proyecto mediante las tareas realizadas y pendientes.
- **Balsamiq:** herramienta que sirve para crear mockups y wireframes para el proyecto. El resultado de este programa son imágenes escaparate de las páginas web que están incluidas en esta memoria.
- **Google Docs:** para la redacción de la presente memoria entre todos los miembros del grupo.

El trabajo se ha distribuido al comienzo de la práctica entre todos los miembros del grupo, de forma razonada, en función de sus conocimientos y preferencias.

Cada uno ha ido haciendo su parte, coordinándose con sus compañeros para que el producto final fuese más coherente.

Desde el principio se ha dispuesto de un servidor dedicado que aloja la base de datos a la cual pueden acceder sólo los desarrolladores de la práctica. La alta disponibilidad del equipo favorece el desarrollo del proyecto.

Para gestionar el código fuente del proyecto hemos utilizado el SCV Git mediante la plataforma GitHub. Cada miembro del grupo tiene su propia rama dónde va desarrollando las issues que se le han asignado. Cuando se considere como terminado un issue, el propietario de la rama se encarga de avisar al resto de compañeros antes de realizar un merge con la rama principal del proyecto.

Conforme íbamos avanzando con el proyecto nos planteamos buscar alguna forma de desplegarlo para que los demás proyectos de nuestros compañeros puedan conectarse y

consumir nuestros servicios. Por este motivo, lo que hicimos fue alojar la aplicación en otro equipo ubicado en el domicilio de uno de los miembros de nuestro grupo.

Disponemos de dirección IP estática y dos servidores con alta disponibilidad. Uno aloja la BD y el otro se conecta a la BD del primero y aloja la página web.

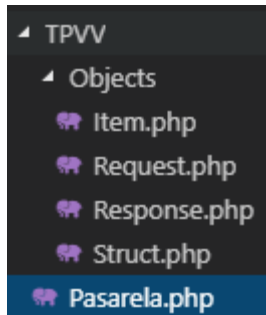
Al disponer de IP estática, podemos compartirla perfectamente con el resto de grupos, pero queda poco profesional tener que aprenderse de memoria una dirección IP.

Por este motivo, hemos optado por registrar un dominio gratuito llamado pasarela.ml para realizar todas las conexiones desde fuera sin necesidad de saber la dirección IP del servidor Web.

5. Descripción de la implementación

PA\$ARELA engloba 2 características principales: la plataforma de pago virtual y la gestión de ella misma.

La **plataforma de pago** permite realizar a los comercios asociados transacciones, mediante el uso de intercambios de información seguros y sencillos.



Se ha diseñado una clase **PASARELA** que implementa todas las funcionalidades necesarias para los clientes. Internamente utiliza otras clases que almacenan la información necesaria que esta necesita para funcionar.

El cliente simplemente deberá de llamar a pocos métodos para construir el pedido y poder generar los mensajes cifrados que nuestro servidor espera.

Nuestro servidor validará todas esas peticiones que le lleguen, dando un visto bueno o excepciones de error.

```
<input type="hidden" name="tpvv_request" value="M0Pe/N1TWzjdM0F/a+12AcpKI+E8x  
Itj5LB8M6v11RidTTSI0JgLN/0KmLRC0UVC1VHstB93dFi0EsscfqFRNqgcZx1DvJunVPYXF+A1FvE  
bN7hDEbqWzU6FRaJDMziSrEA7Hi4WUYiQcgE4x7XXdbPNuUmw2Qvq5jiAEMC5p+x+1TnBJDC1Zriy+  
rY/a7b8KhHsbdo3bKPWylranNcPzUBJ6kTuWKPeb15CUsycTctjPvUBFW6iKdZ1kxMv+vu7HmJ0ufB  
H62ecxNxmG49ALecQr93Wf6GubJpnnItLFCb8+wkVmsVGSSAig8itPXx8Q5EUzJ2rwn9rau0pa9q0  
+gNF5Fw7rUFVEkcxq4lorqHwGP/dz7bFWj/HwJWi2eSSCTk2kZiusHf/ly704rNSbceoGnXaGBVCRy  
AHm5JGoGUfg+XvadW1njApCK0iLbypChDdIkFqZ0vyuWb1a9vAb41mi4WslhyGbAOTCCpVBg6yCCXN  
IhFh3dCBI0BfFgimC3cgpuIF5KuAnlmVEhUF00SAzMU0ra8kaqvA/6XhQDmpXS3aEj6CzPg1gxX4oQ  
8AnLzGNgI++Dme091Cs9G0g11KN0MP1vw2oLNOp24zK611sA3PPjksX77B0tSWWJJJAaCmV/7FnLabq  
9N0k11RVqBwIbL0z1Yj1I1E5je1p6HUUVZxFokYnvXjG5UDMQaFX8Tq1719DxRj0KYtt1kjqQ==">
```

Introducir tarjeta de credito	Lista de productos asociados
	x - x1 d - x2
<input type="text" value="4525 1531 5465 4645"/> <input type="text" value="11 / 2021"/>	Precio 1.00€ PAGAR!
<input type="text" value="Pepe martinez"/> <input type="text" value="132"/>	

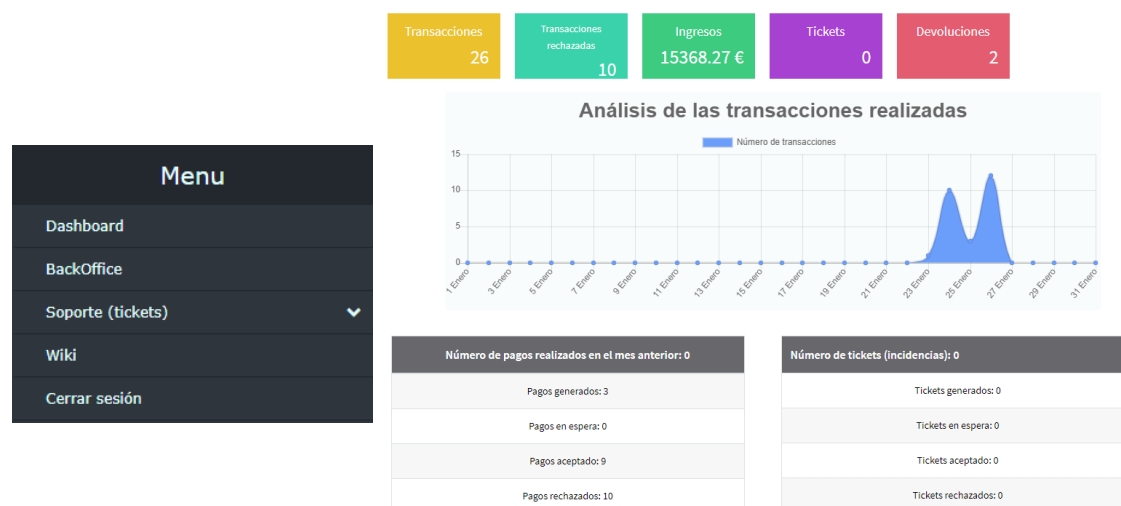
En caso de que todas las validaciones sean correctas, accederemos al formulario de pago. Este formulario muestra toda información acerca del pedido, como puede ser la lista de productos, el precio y los campos a rellenar sobre la tarjeta de crédito.

Tras pagar se realiza una comprobación básica de los datos introducidos y posteriormente una **simulación** de tarjetas. Se ha diseñado que todas las tarjetas que empiecen por 50 y 51 sean automáticamente válidas. Por el contrario, todas las que empiecen por 46 y 52 se tratarán como tarjetas de crédito no válidas que producirán errores diferentes dependiendo de los números introducidos a continuación.

45XX	Visa	TODO OK	<div> <div>Tarjeta desactivada</div> <div>Error autenticación titular</div> <div>Intentos PIN excedidos</div> <div>CVV incorrecto</div> <div>Saldo insuficiente</div> <div>Caducada (check)</div> </div> <div> <div>00-19</div> <div>20-39</div> <div>40-59</div> <div>60-79</div> <div>80-99</div> <div>TODOS</div> </div>
51XX	MM	TODO OK	
46XX	Visa	TABLA	
52XX	MM	TABLA	

Una vez finalizada la simulación, se generará una respuesta con el resultado de la transacción, que será enviada al punto de encuentro que determina el comercio, a este punto le hemos llamado **endpoint**. Llegada la información al cliente, se realizará un proceso de verificación, con la idea de comprobar que el origen del destino somos nosotros. Una vez validado el mensaje, podrá obtener la información que desea, como es el pedido, el resultado de la transacción y la fecha.

Para poder gestionar todo esto hemos habilitado la segunda característica: **el panel de usuario**. Cualquier comercio podrá iniciar sesión en la parte de usuarios y tendrá acceso a diversas características, como son un **dashboard**, **listado de transacciones**, **el soporte técnico** y **un tutorial de como incorporar PA\$ARELA** en su pagina web.



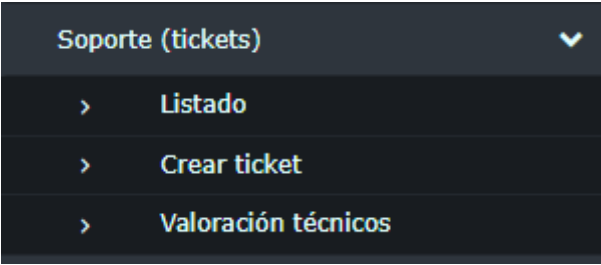
En el **dashboard** podemos ver un gráfico del mes actual con las transacciones realizadas e información asociad, como puede ser los estados o los ingresos.

Todas los estados Todos los importes FILTRAR

ID	Importe	Estado	Tarjeta	Comentario	Accion
12	1.00 €	devolucion	***** 4651	Devolucion	Detalles
13	15.00 €	devolucion	***** 4546	Devolucion	Detalles
14	125.21 €	aceptado		Devolucion	Detalles Devolver
22	2.00 €	aceptado	***** 2312	Devolucion	Detalles Devolver
34	10.00 €	rechazado	***** 0000	Error de sistema	Detalles Devolver
35	10.00 €	rechazado	***** 1100	Intentos PIN excedidos.	Detalles Devolver

En el apartado de Backoffice podemos ver el listado de transacciones. Se puede consultar todos los detalles de dicha transacción o incluso solicitar una devolución.

Si algún surge algún problema con alguna transacción nuestra página web ofrece un servicio técnico mediante el cual podrás chatear con un técnico especializado que hará todo lo posible para solventar el problema. Al cerrar el ticket se podrá valorar al técnico.



Mensajes: 5

Comercio: (Jose1)	Técnicos: (Yo)
Hola, estoy haciendo un pago y me dice que el número de la tarjeta no es válida	
24 Jan 2019 17:01:44	
	Hola, estamos revisando la página. En breve le informaremos.
	24 Jan 2019 17:02:13
Tengo mucha prisa por realizar la compra, pueden darme una respuesta?	
24 Jan 2019 17:27:38	
Lo necesito hoy mismo, no tengo tiempo	
24 Jan 2019 17:28:20	
Ya lo he arreglado	
24 Jan 2019 22:39:16	

Un ejemplo de conversación mediante tickets

Para dar servicio a toda la aplicación se han desarrollado 3 roles de usuarios. El primero de todos y ya explicado sería el rol comercio, que permite generar transacciones. Por otro lado, tenemos el rol técnico que es el que ayuda al comercio en todo lo que puede. Por último, tenemos el rol de administración que es el controla toda la aplicación y da de alta comercios y técnicos.

6. Problemas encontrados y solución

- IP dinámica del router:

El router principal que direcciona todas las peticiones tiene concedida una IP pública, pero ésta cambia cada 2-3 días. Es muy poco profesional ir comunicando la nueva dirección IP al resto de compañeros cada vez que ésta sufra un cambio.

La solución fue contactar con el proveedor del servicio y contratar la IP estática para siempre. Esto supone un coste adicional de 5€ + IVA en la factura de Internet.

- Conflictos al mezclar ramas en GitHub:

El tiempo iba avanzando. En época de fiestas y exámenes nos hemos “distráido” con la práctica. Empezamos trabajando de manera normal con Git y, cada vez que alguien quería hacer un merge, perdíamos bastante tiempo en solventar todos los conflictos producidos por la variación del código.

La solución fue emplear la herramienta Neo Router que nos permitió crear una red virtual entre todos los miembros del grupo. Al tener una red privada virtual entre nosotros, tenemos la posibilidad de compartir carpetas. La carpeta compartida fue la del proyecto entero que todo el mundo podía ver y modificar en tiempo real. Cada cierto tiempo (muy a menudo) se iban haciendo commits para mantener la coherencia de trabajo. Así era posible avanzar más rápido sin perder tiempo valioso en resolver conflictos.

- Fallo verificación CSRF:

Cuando se hicieron las primeras pruebas de interoperabilidad con el resto de grupos, se producía un error 419. Esto se debía a que el token generado por el proyecto Laravel de los otros grupos no podía validarse por nuestro propio Laravel.

La solución fue desactivar la validación de ese token y el problema no volvió a mostrarse otra vez.

- **Fallo NeoRouter:**

Todo iba funcionando perfectamente hasta que TeamViewer dejó de responder en el equipo que hace de servidor. Tras investigar un rato pensamos que era culpa del propio TeamViewer y descartamos que el culpable sea el equipo ya que seguíamos teniendo acceso a la carpeta compartida, a la base de datos y al servidor web. En un momento dado decidimos reiniciar el ordenador “por si acaso”. Desde ese momento, NeoRouter dejó de funcionar. Ningún miembro del equipo era capaz de conectarse a él.

El problema estaba en que el ordenador se había actualizado de forma transparente para el usuario. Cosas de Windows... Tras comprobar el estado del equipo (y varios reinicios), conseguimos arrancarlo y nos dimos cuenta de que el sistema operativo se había actualizado a la versión 1803. Investigando en foros del programa descubrimos que había un fallo en la ejecución del servicio que impedía el correcto funcionamiento del propio servidor de NeoRouter.

Lo solucionamos cambiando una serie de parámetros del adaptador virtual que ha creado NeoRouter y el problema se solventó.

- **El dominio pasarela.ml:**

Esto realmente no se considera un problema, pero sí que merece la pena mencionarlo. El dominio adquirido tiene dos formas posibles de redirigir al servidor web propio:

Mediante un redireccionamiento que, cuando se accede al dominio, redirige directamente hacia la dirección IP del servidor y el usuario la verá en su barra de direcciones. Esto quedará muy feo en un producto final.

La otra opción es cargar todo el contenido devuelto por el servidor Apache en un iframe dentro de la página pasarela.ml. De esta forma se asemeja más a una aplicación SPA cuyo contenido se muestra siempre dentro de la misma URL y no tiene necesidad de navegar a otra para realizar cualquier interacción con el sistema. Hemos optado por esta opción por estos motivos ya que oculta la dirección IP del servidor. Aunque tiene sus desventajas como la pérdida del favicon de la página así como el título de la página que resulta imposible de cambiar ya que no tenemos acceso directo sobre el código HTML de la página maestra que carga el iframe.

7. Mejoras y ampliaciones

- Mejoras

Lo más importante es siempre la comunicación. Haber tenido el grupo más comunicado, en especial en la época de fiestas, habría sido clave para lograr la terminación del proyecto con mayor rotundidad. El tiempo se nos ha echado un poco encima, aunque no hemos tenido grandes problemas para juntar las partes de cada miembro del grupo. No nos hemos entorpecido el código desde la utilización de NeoRouter, lo cual ha mejorado mucho el rendimiento.

- Ampliaciones

Tickets

1. La información del listado es mejorable, como por ejemplo indicar el comercio y técnico asociados para un administrador, con su adecuado diseño.
2. Ofrecer un filtrado de tickets por estado.
3. Tanto para comercio como técnico, indicar si hay un mensaje nuevo por leer.
4. Mejorar el diseño de la mensajería de los tickets.

Global

Trabajar todos bajo una misma filosofía de rutas con las herramientas que el propio Laravel proporciona para facilitar el enrutamiento entre vistas de distintos miembros del grupo.

8. Referencias

Documentación de los talleres de Laravel proporcionados por el profesor.

Documentación de los patrones de diseño de la asignatura.

<https://laravel.com/>

<https://laracasts.com/>

<https://stackoverflow.com/>

<https://getbootstrap.com/>

<https://www.w3schools.com/>

<http://www.neorouter.com/>