

PYTHON

Nombre y Apellidos

PROYECTO FINAL



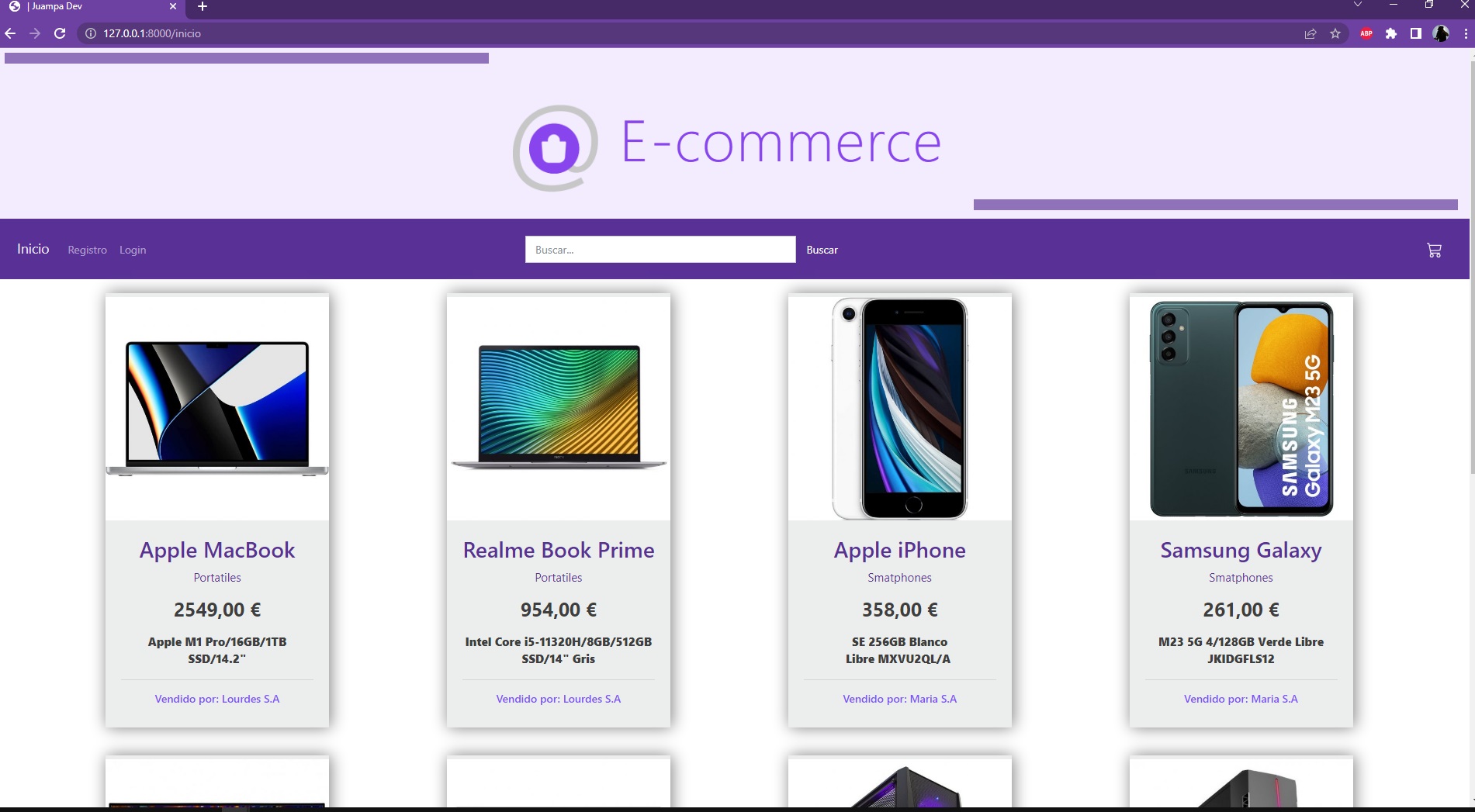
Juan Pablo Oltra Jiménez

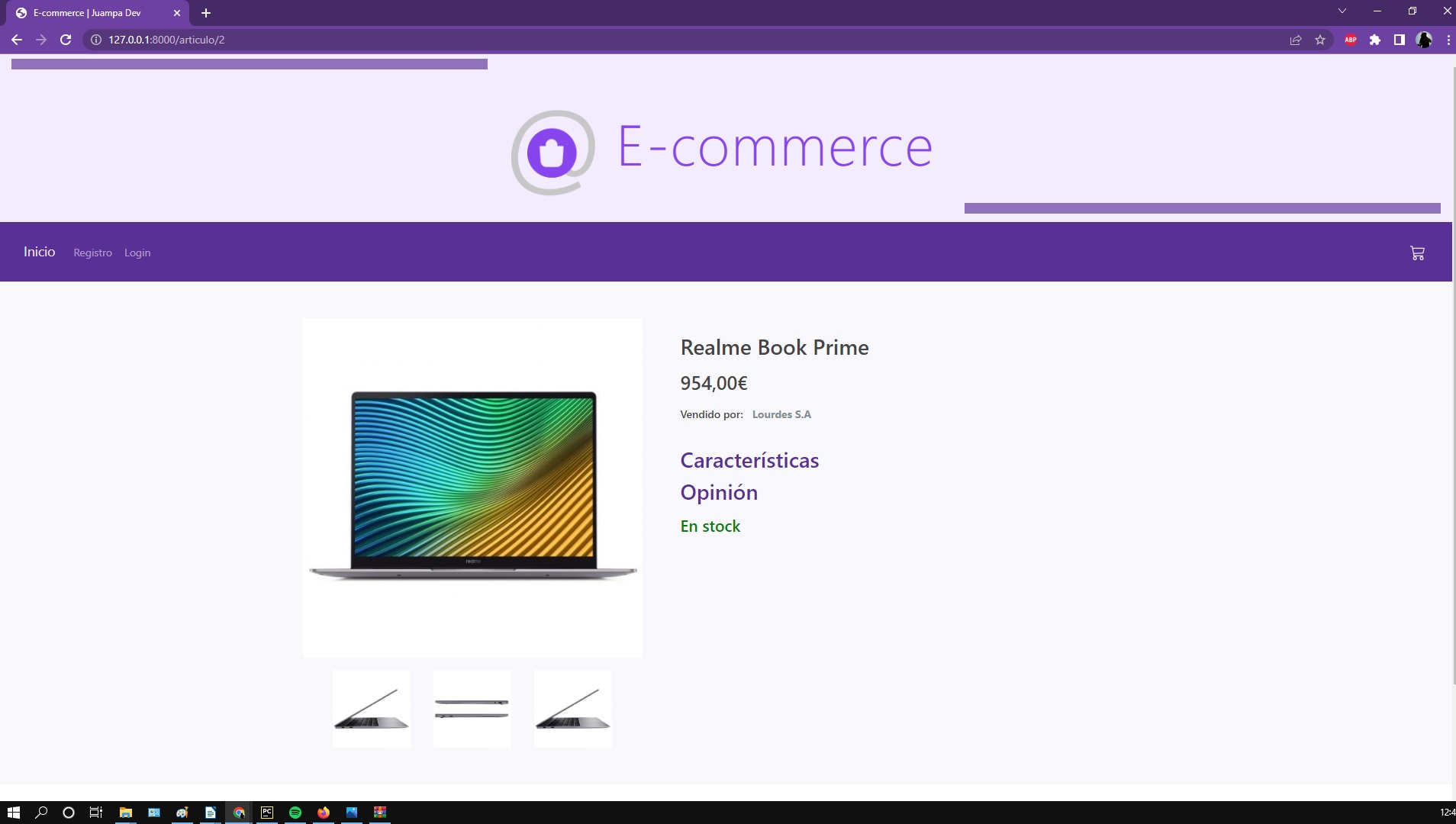
Estudiante de Programación en Python

**Fecha:** 26, 09, 2022

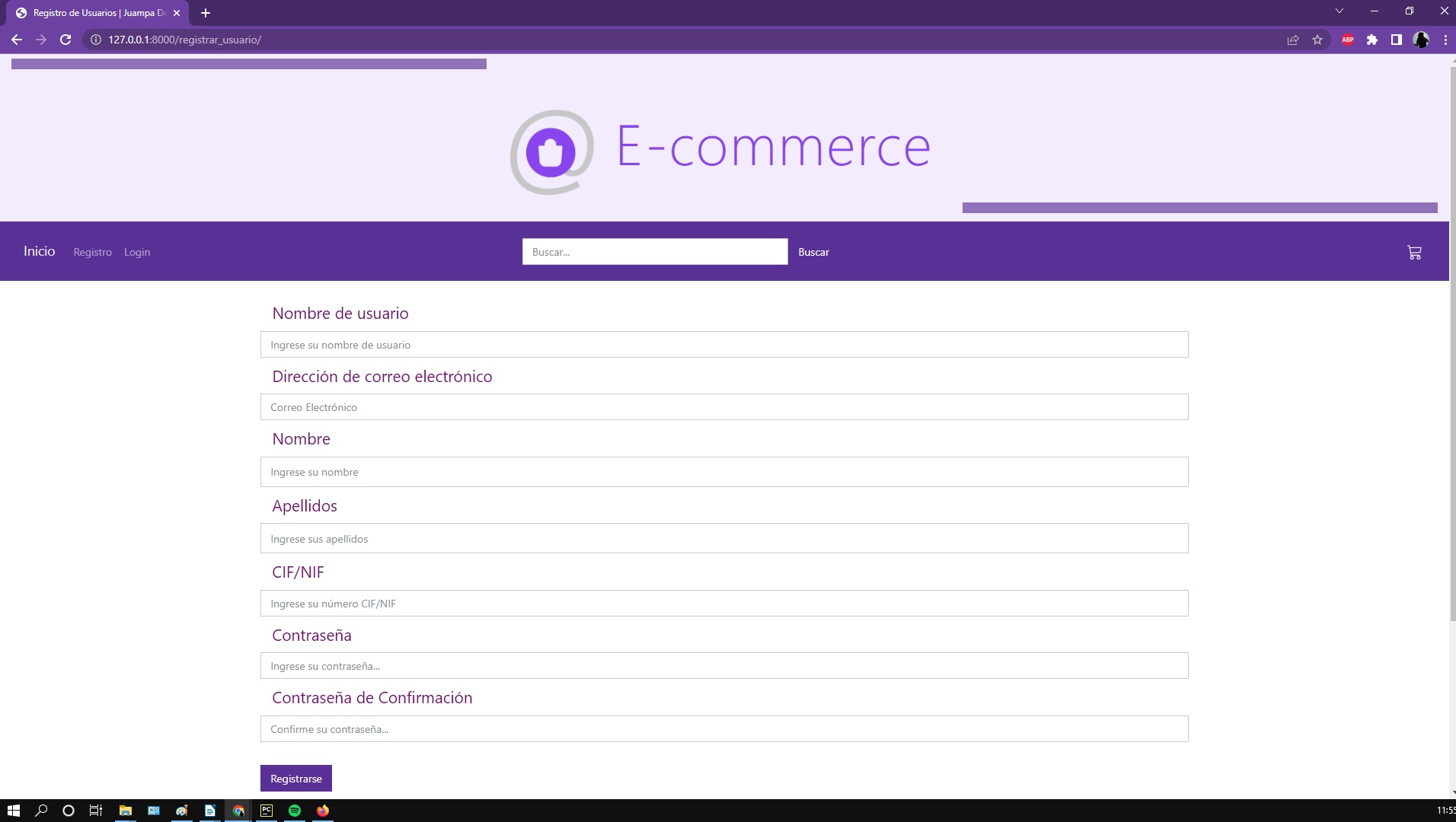
**Aplicación web para una Ecommerce**

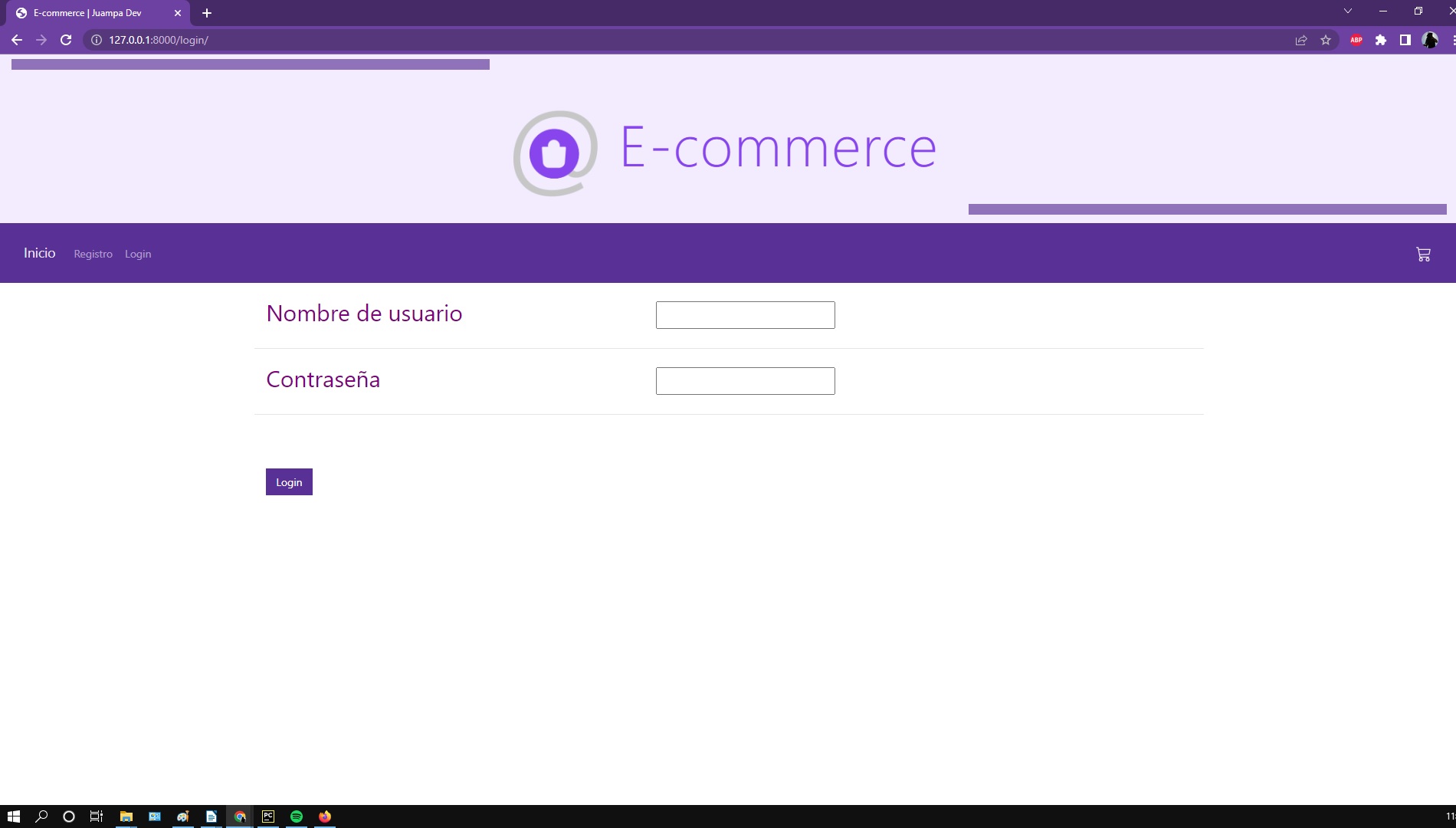
Desarrollo

En la primera imagen podemos ver la página inicial de la web, en esta pagina el usuario esta sin loguear y como vemos sólo tiene acceso a la lista de productos y a una barra de búsqueda con la que puede buscar artículos, no discrimina ni mayusculas ni nombres exactos.El usuario no tiene opción de comprar si no esta autenticado, no le aparece el botón de comprar como vemos en la siguiente imagen.

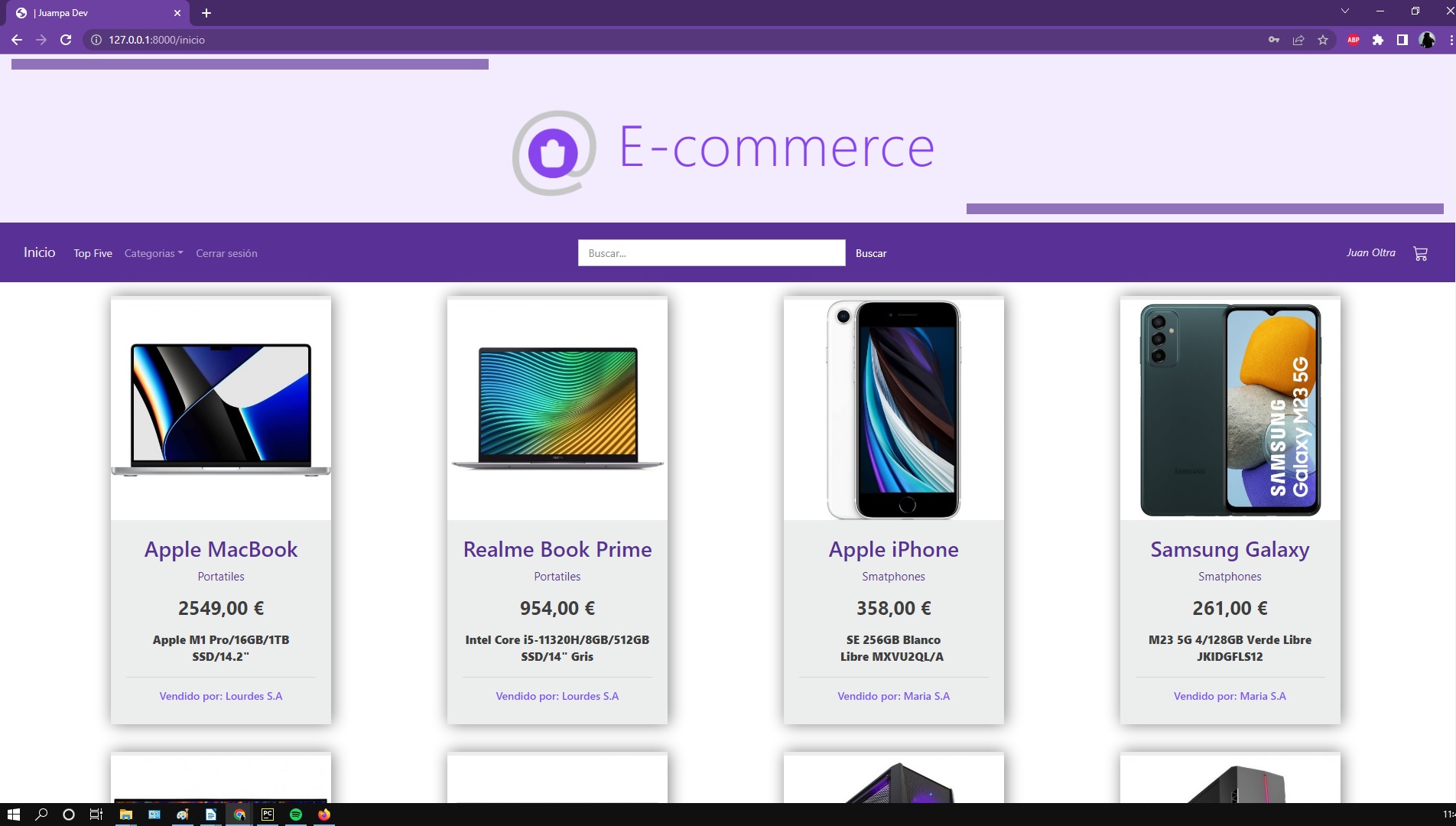


Página de registro.

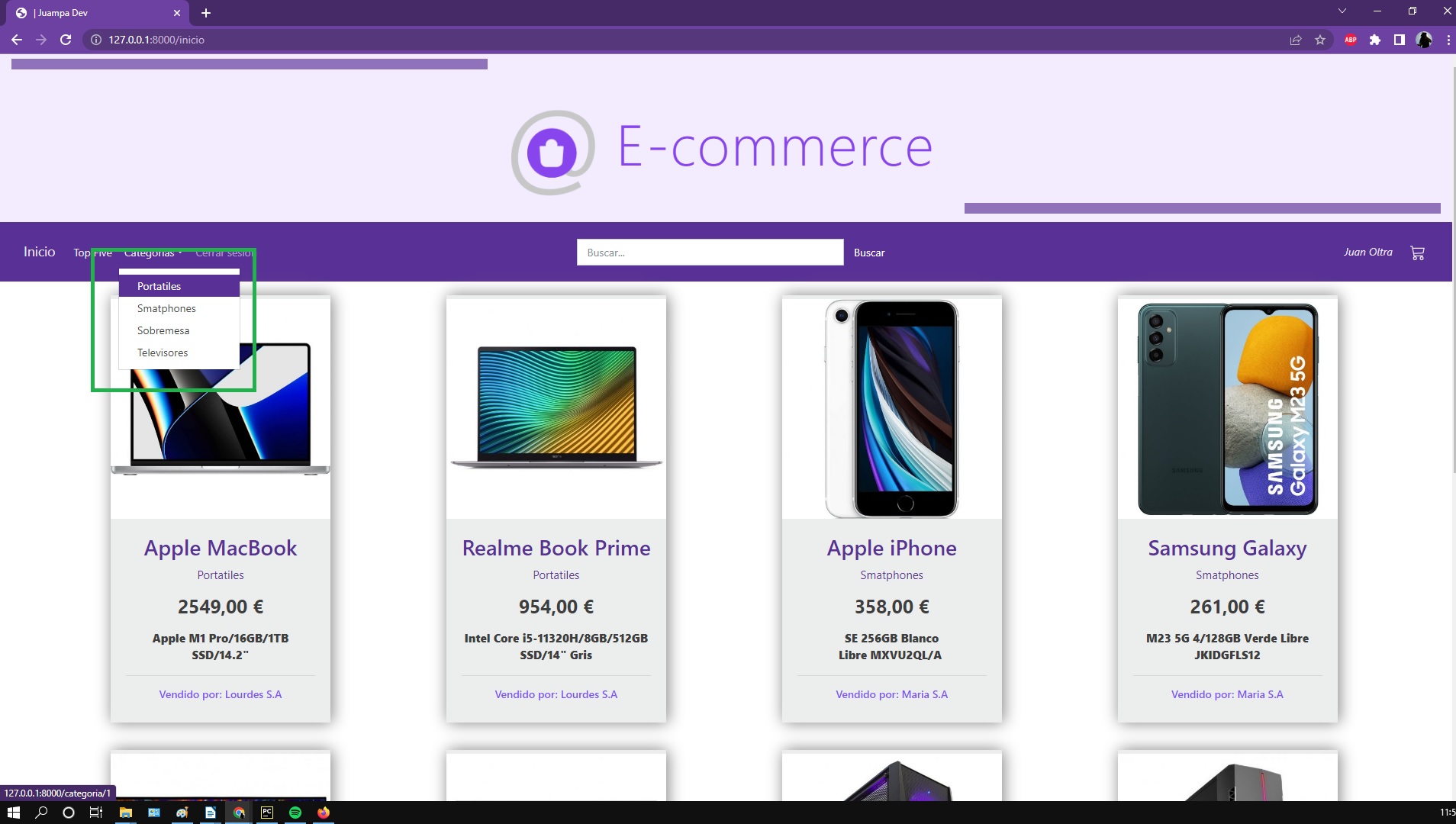


 Página de login.

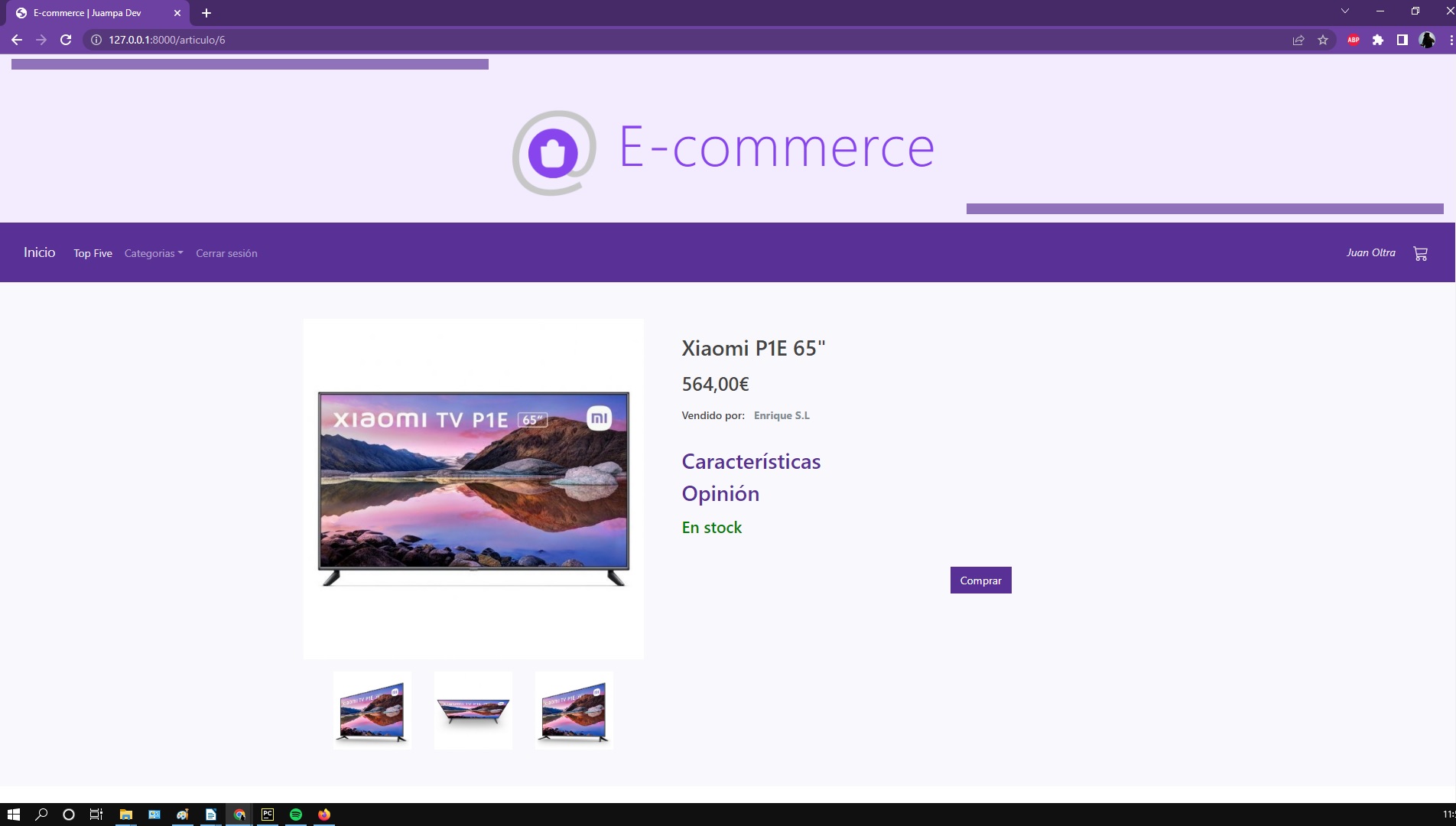
Una vez logueados ya tenemos acceso al resto de botones de la página.



Tenemos un botón, dropdown, donde podemos seleccionar los artículos por sus categorías como vemos marcado con un recuadro verde.



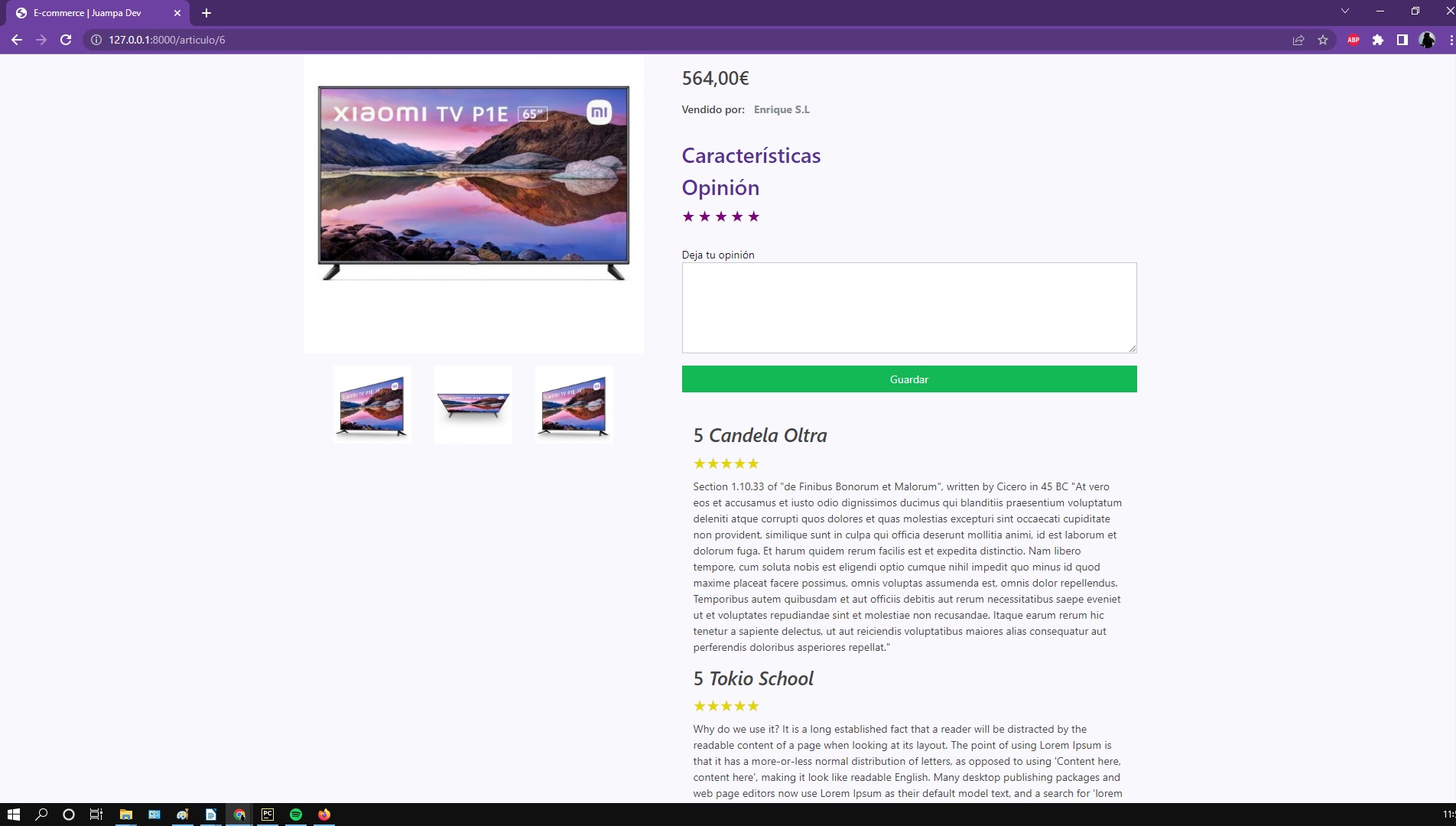
Vemos la página de detalle del artículo donde podemos encontrar la información del mismo. Encontramos el nombre, precio, el proveedor que lo vende y dos desplegables, uno donde puede verse las características del producto y otro desplegable donde podemos ver las opiniones sobre el artículo y dejar la nuestra.



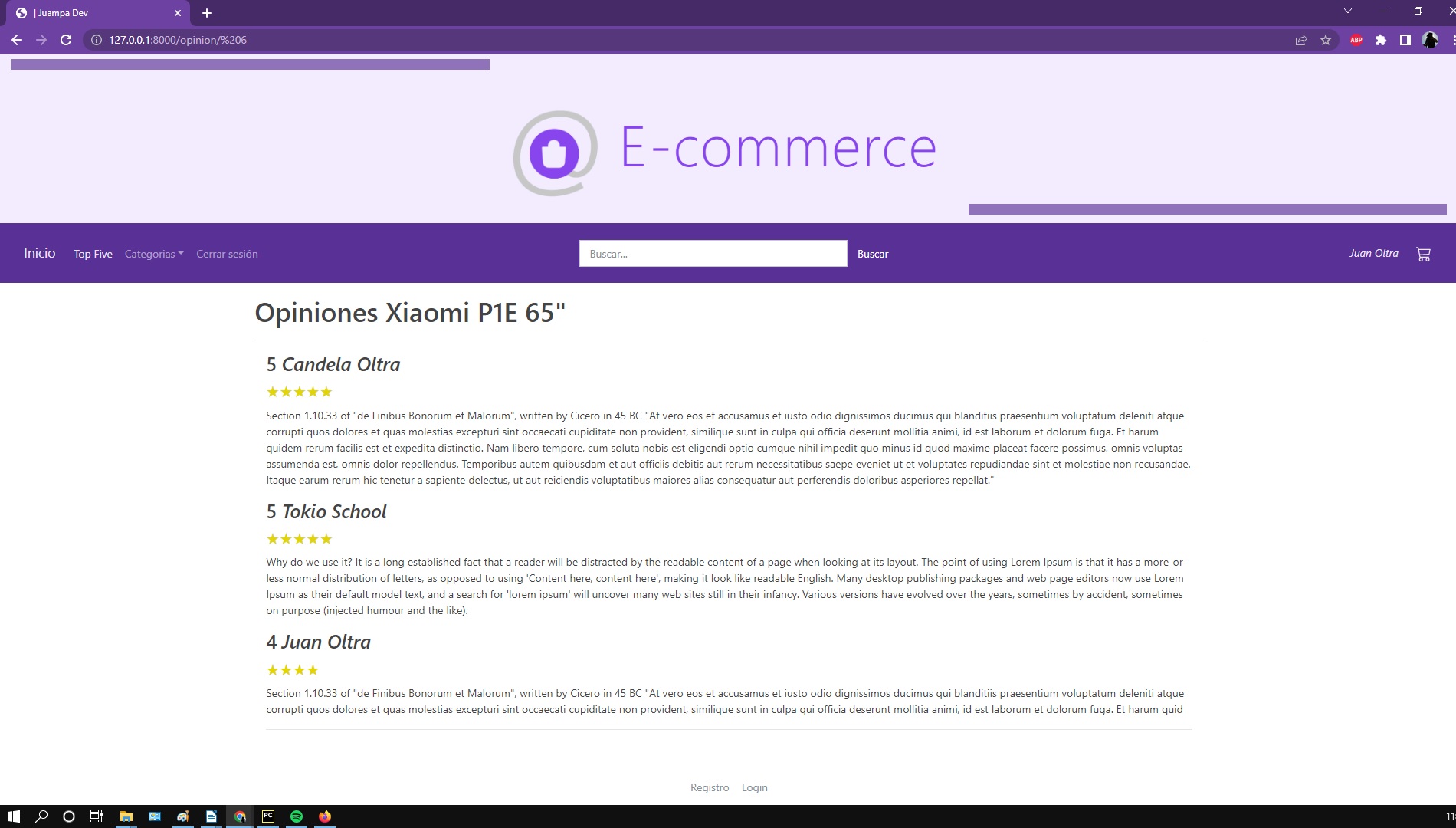
Desplegamos la caja de características del articulo. En el modelo hemos usado un editor de texto externo llamado Ckeditor.



Desplegamos la caja de opiniones del articulo.



Una vez que el usuario ha puesto su opinión la web nos redirige a una página donde podemos leer todas las opiniones de dicho artículo.

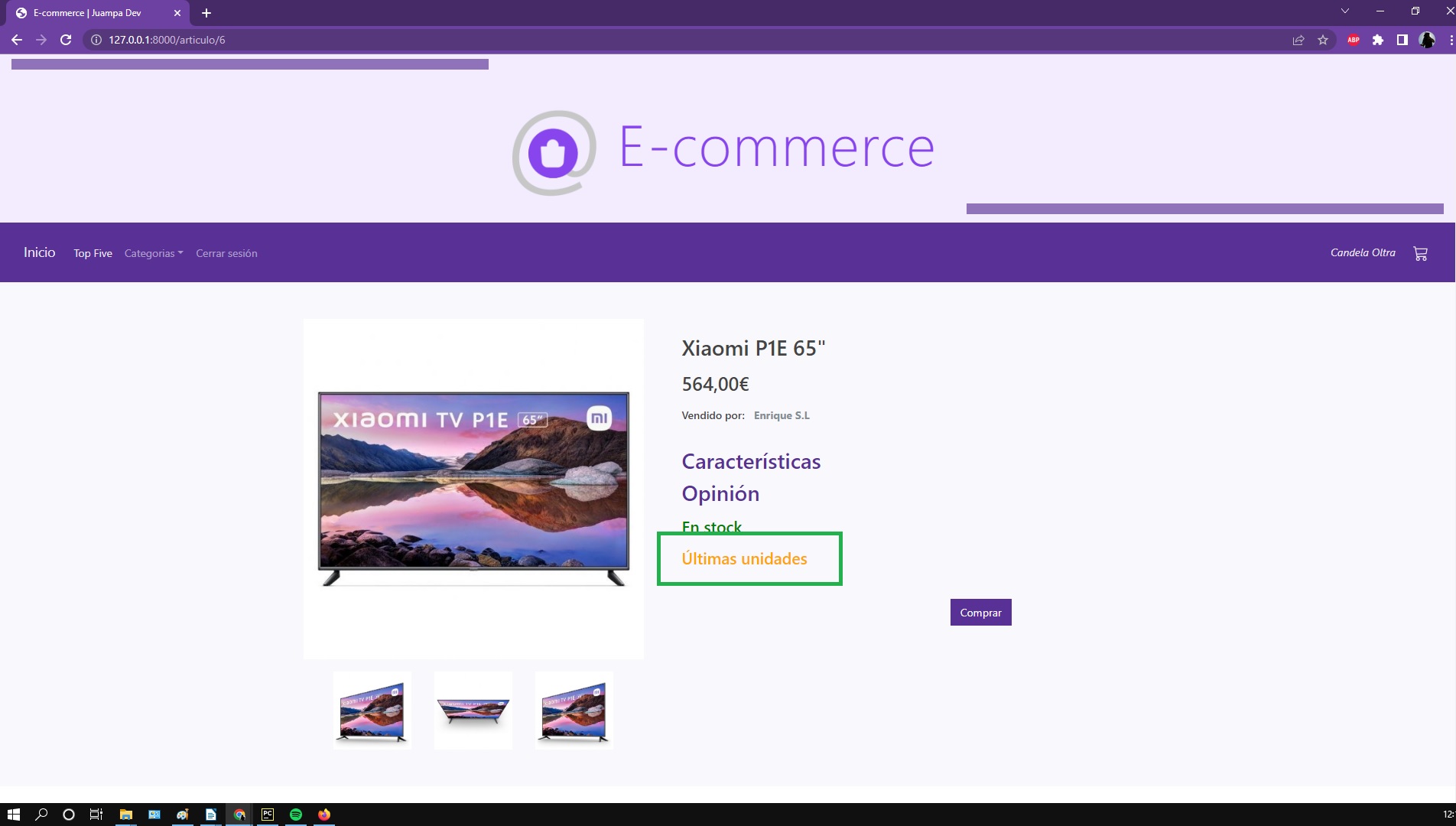


También contamos con detalles como; si el articulo esta disponible o si hay pocas unidades. Este último parámetro lo establece el proveedor al poner un stock mínimo, cuando este es alcanzado la app mandará un correo electrónico al proveedor .

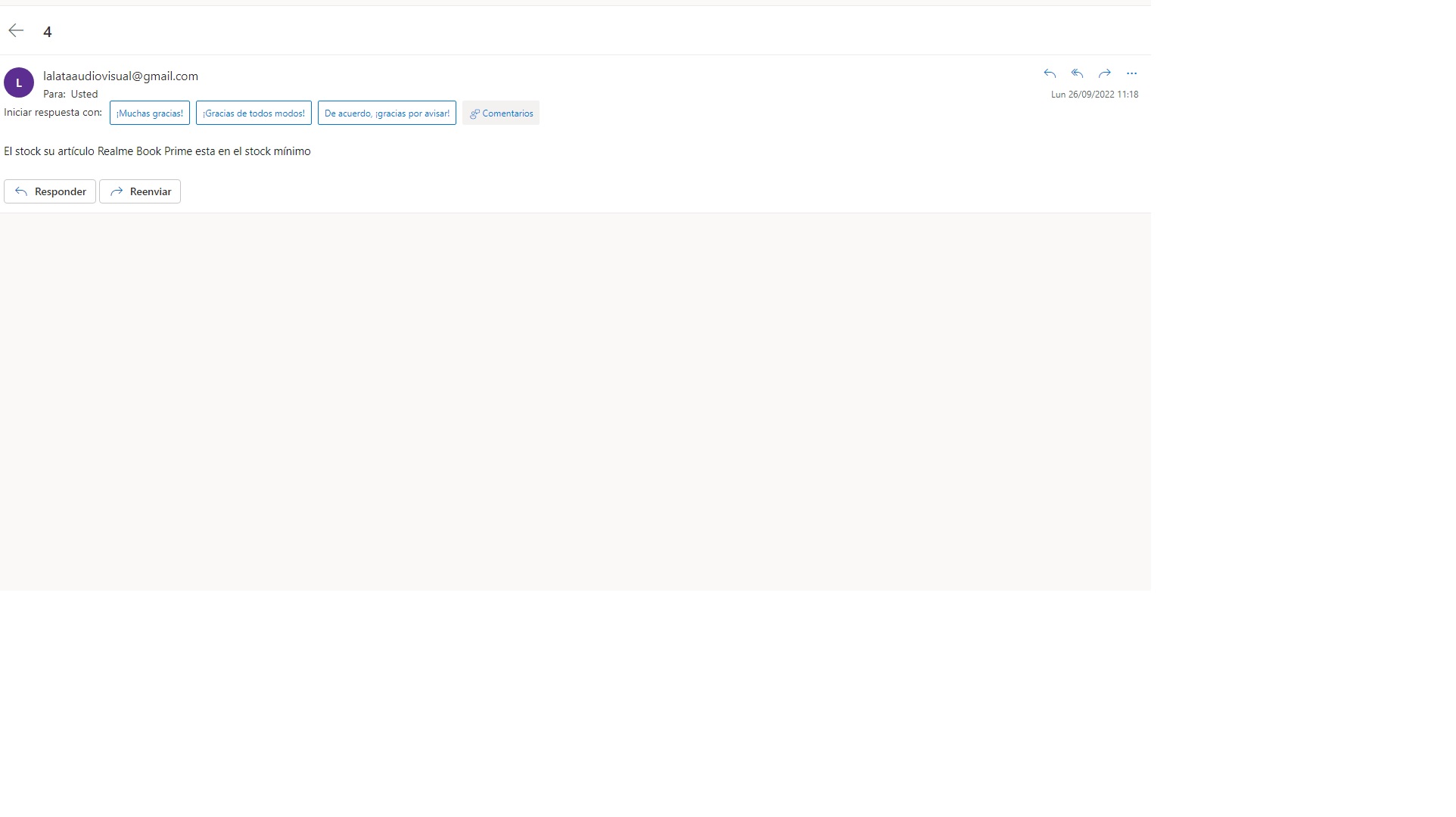
Vemos que si el stock es 0 nos aparece un mensaje en el detalle del artículo que nos dice que no esta disponible.



Si el stock es igual o menor que el stock mínimo nos avisa de que quedan pocas unidades al mismo tiempo que manda un correo al proveedor.



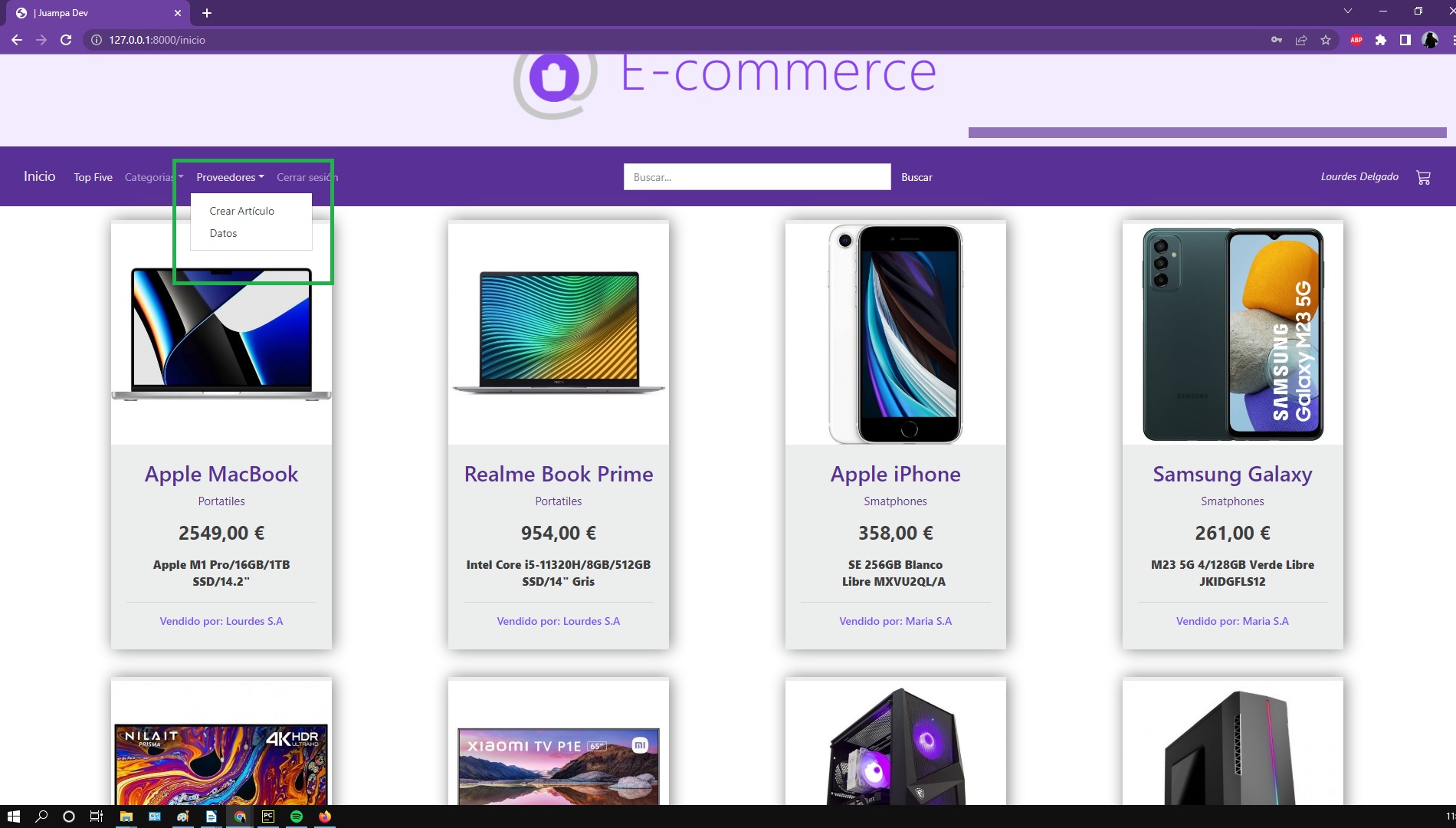
Correo que recibe un proveedor cuando se llega al stock mínimo.



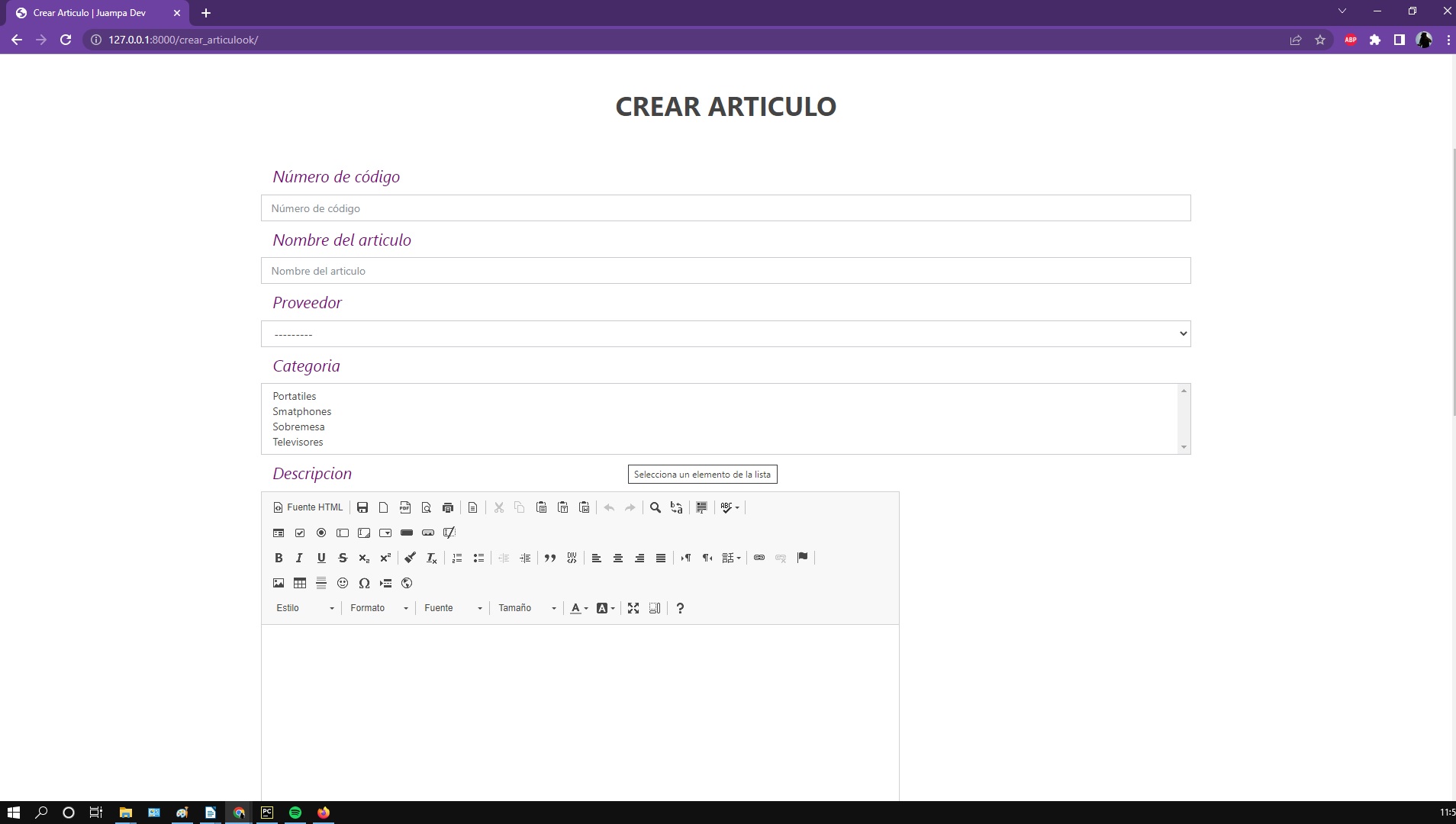
En la web tenemos 2 tipos de acceso sin contar el superusuario. Tenemos el acceso cliente y el acceso proveedor. Los proveedores tienen acceso al admin django en el cual sólo tienen permiso para añadir y ver artículos y categorías. He ampliado el modelo User django y le he añadido el campo cif e is\_proveedor. El atributo is\_proveedor al igual que is\_staff lo da el superusuario pero puedo vinvular un proveedor con el usuario a través del cif que es único con este condicional {% if articulo.proveedor.cif == request.user.cif %} me reconoce al proveedor y me deja ver ciertos elementos de la aplicación.

El proveedor, tiene un botón donde puede crear un artículo y ver datos de sus ventas. Podrá ver una tabla y comprobar el estado de los productos que tiene a la venta en la web. En la tabla se puede observar el stock que tiene disponible el número de ventas del producto, el precio y la facturación total con dicho producto. También el proveedor cuenta con una gráfica disponible donde puede ver el porcentaje de ventas de cada producto y la cantidad vendida.

Vemos el botón de proveedores con: Crear articulo y Datos. Marcado en la imagen.



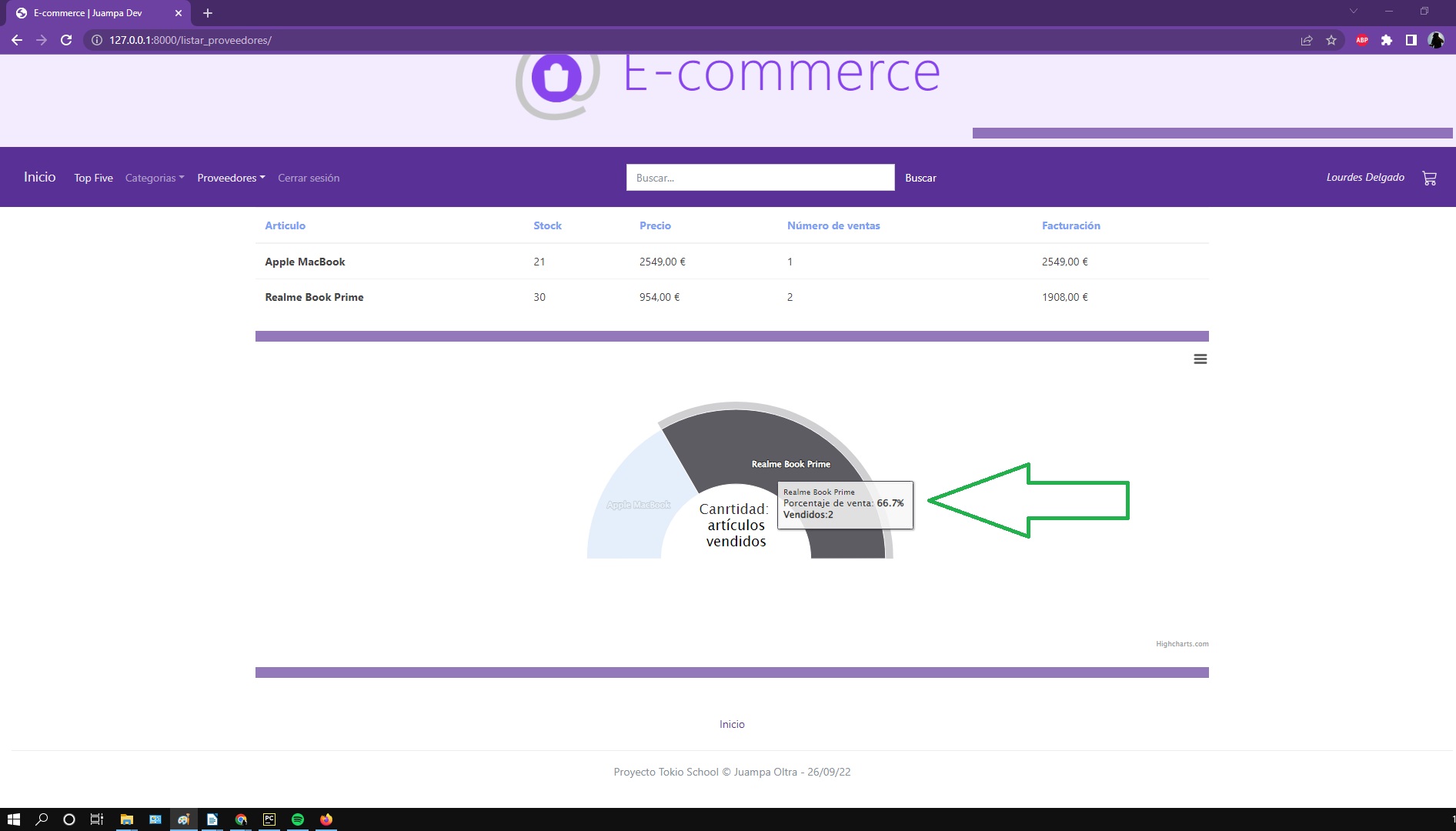
Página para crear artículo.

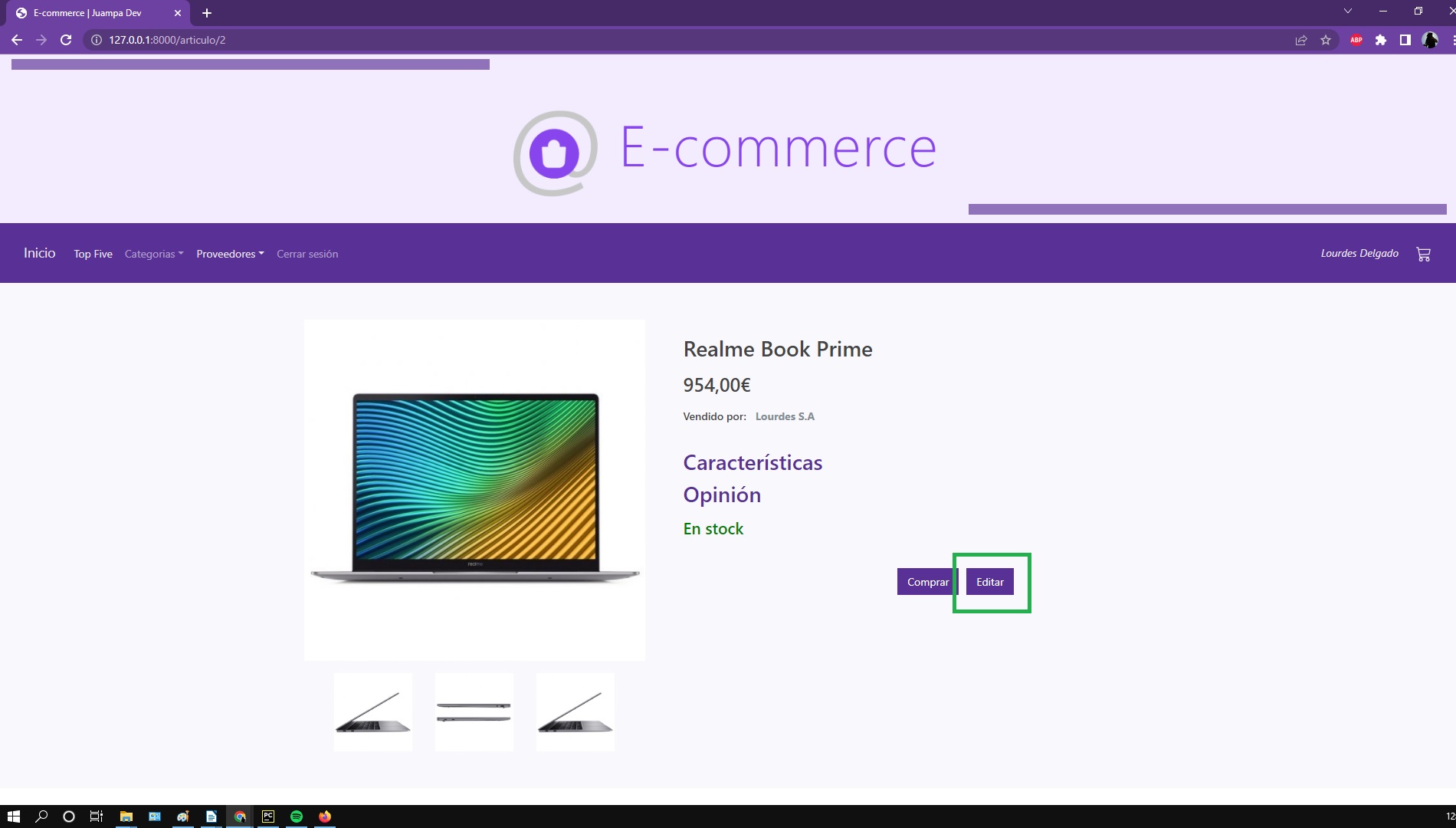


Página de los datos del proveedor.

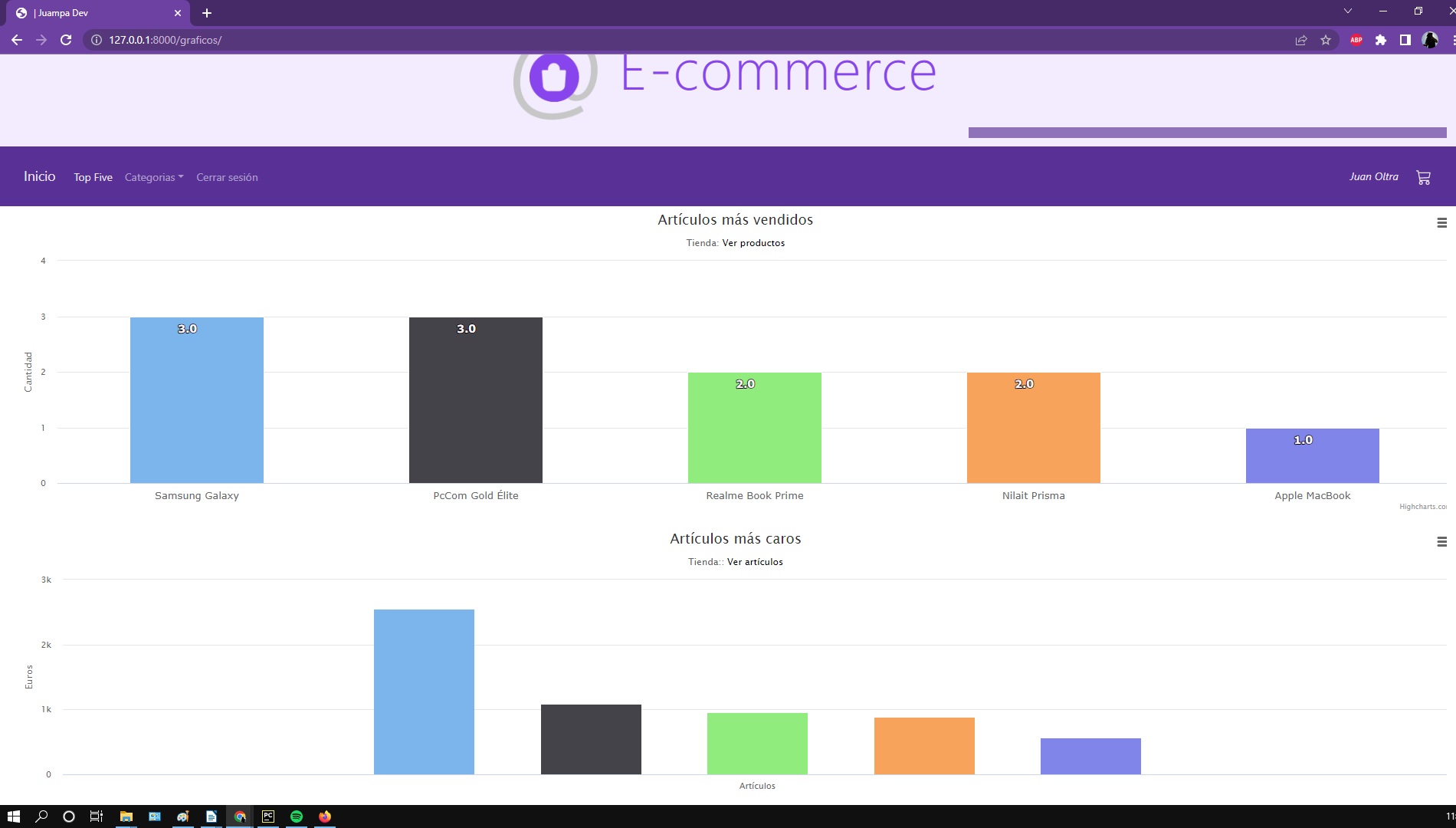
Vemos marcada la tabla y debajo el gráfico. En la imagen posterior podemos ver que el gráfico tiene un tooltip donde aparecen los datos del producto que se seleccione.



 También cada proveedor tiene acceso a un botón de editar producto. Este botón sólo aparece en los productos que cada proveedor a puesto a la venta.

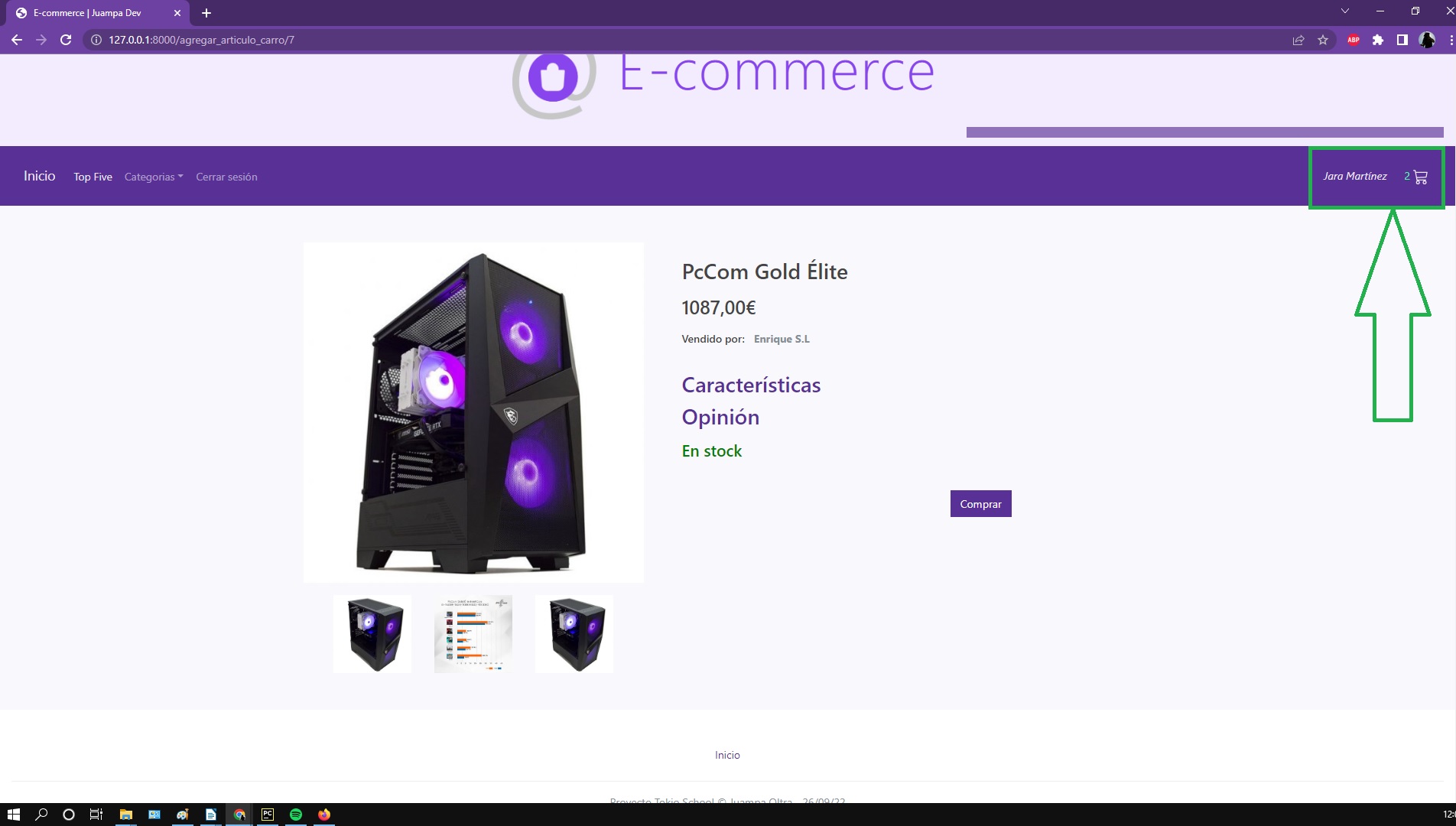


Existe también un botón ‘Top Five’ en el cual todos los usuarios pueden ver gráficos con los 5 productos más vendidos y los 5 productos más caros de la web.

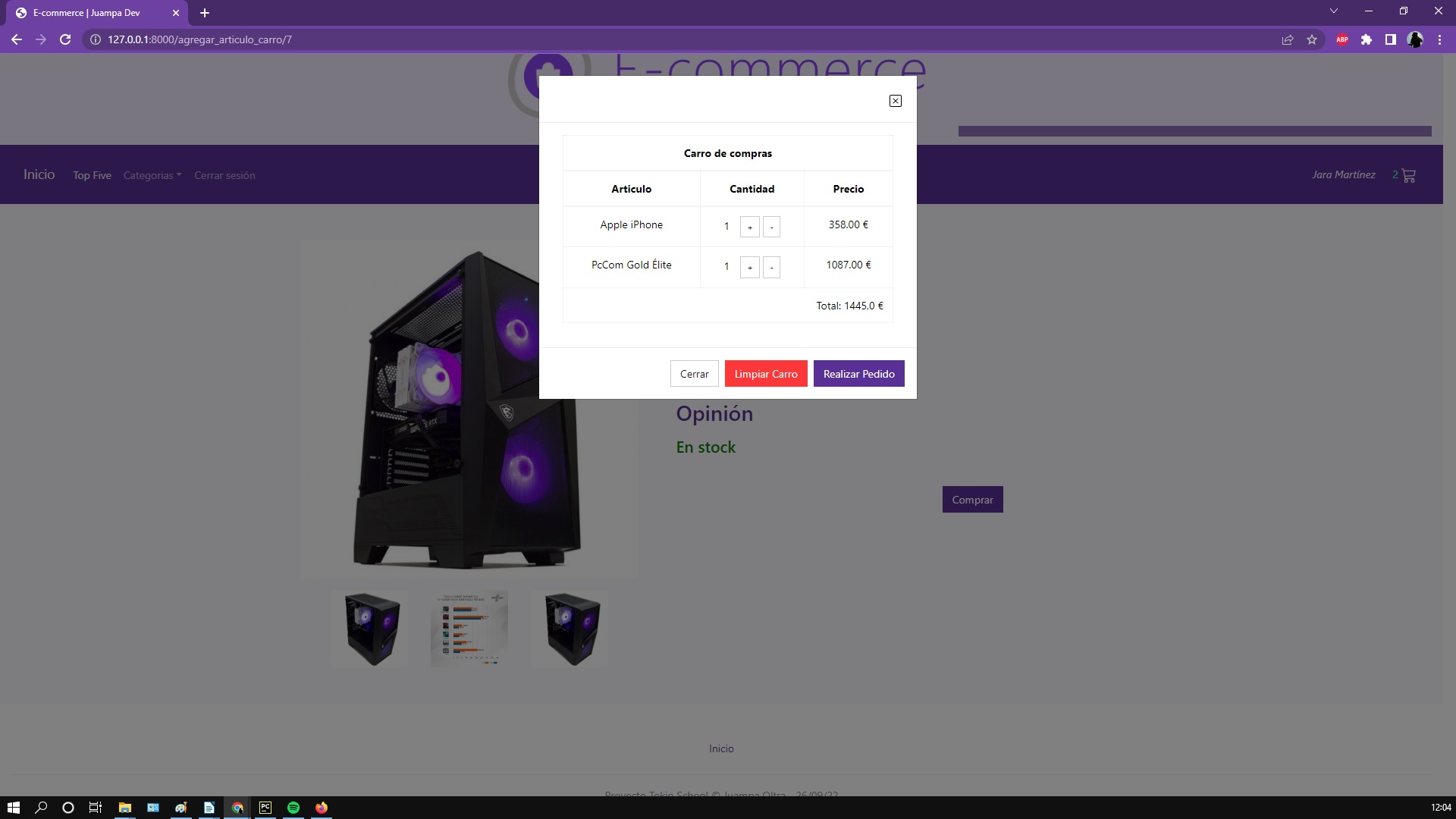


He creado una app de carrito de compras. Con la cual el usuario puede ver en todo momento lo que ha añadido al carrito los productos que lleva agregados y el total en de la compra. La app también manda un correo al usuario al realizar un pedido.

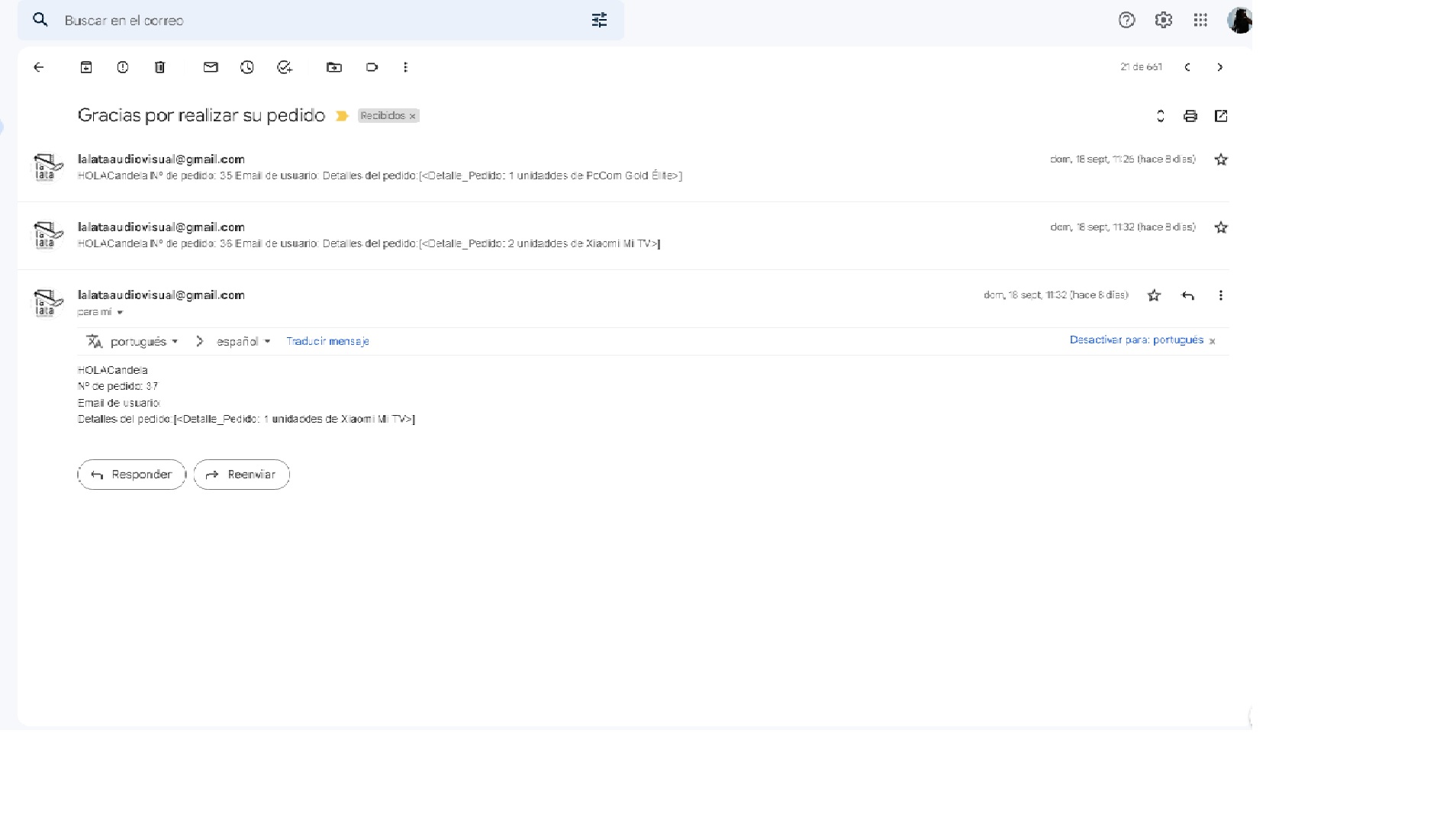
En esta imagen vemos los productos que el usuario lleva en el carro.



En la siguiente imagen vemos un modal con el carrito desglosado.

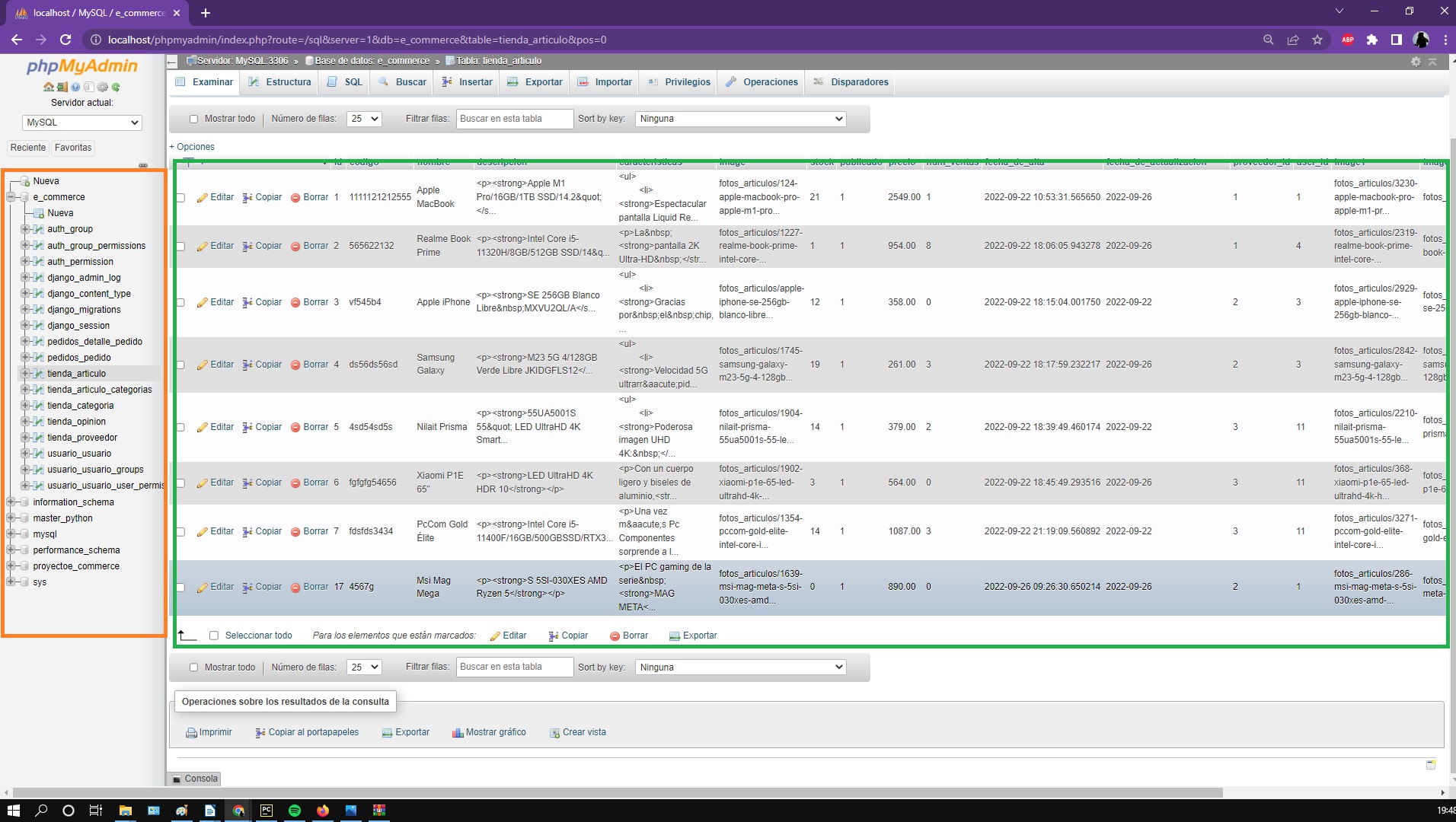


Correos que recibe un usuario al realizar el pedido.

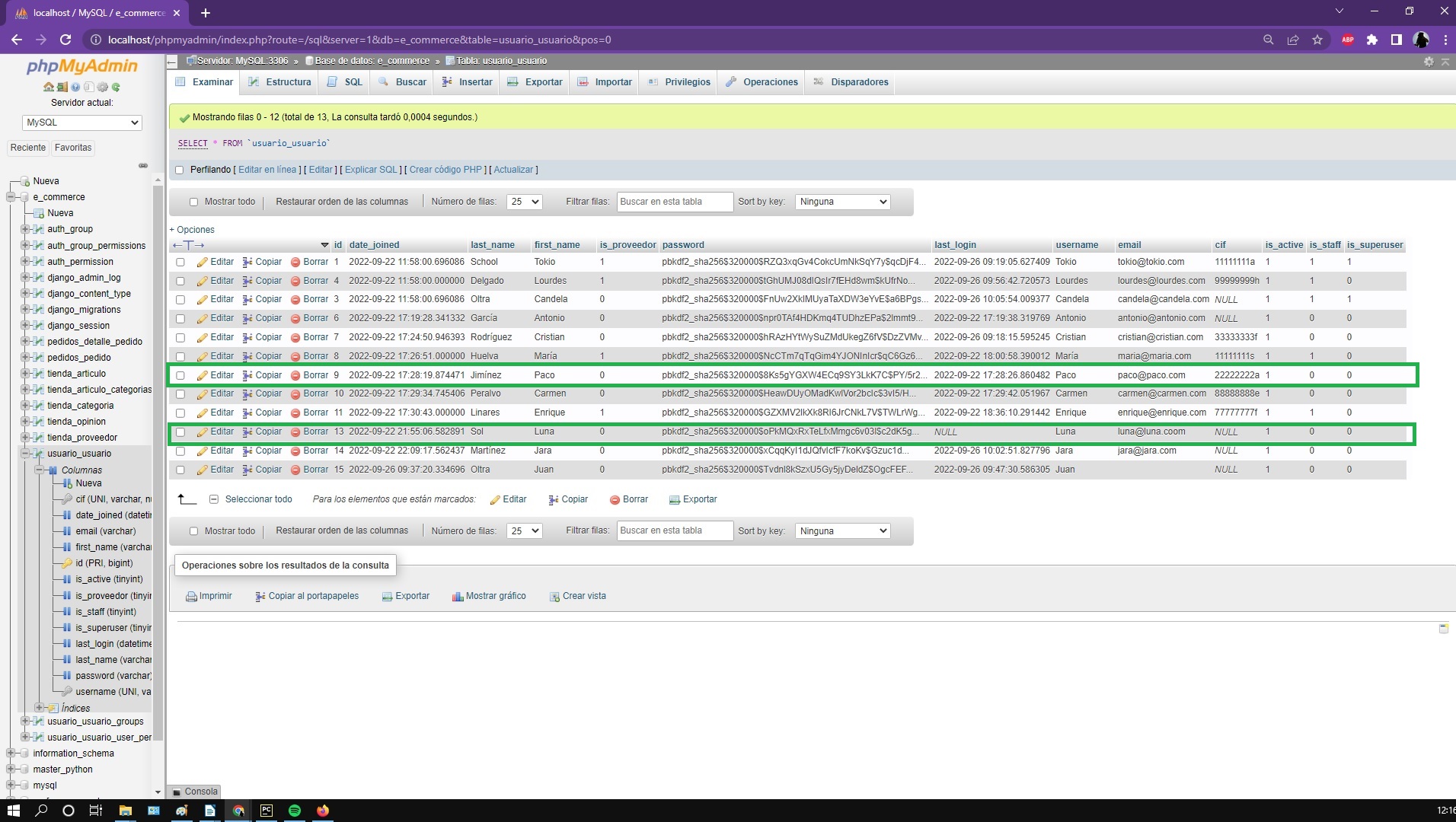


Que mejor fin que con un pedido realizado y enviado.

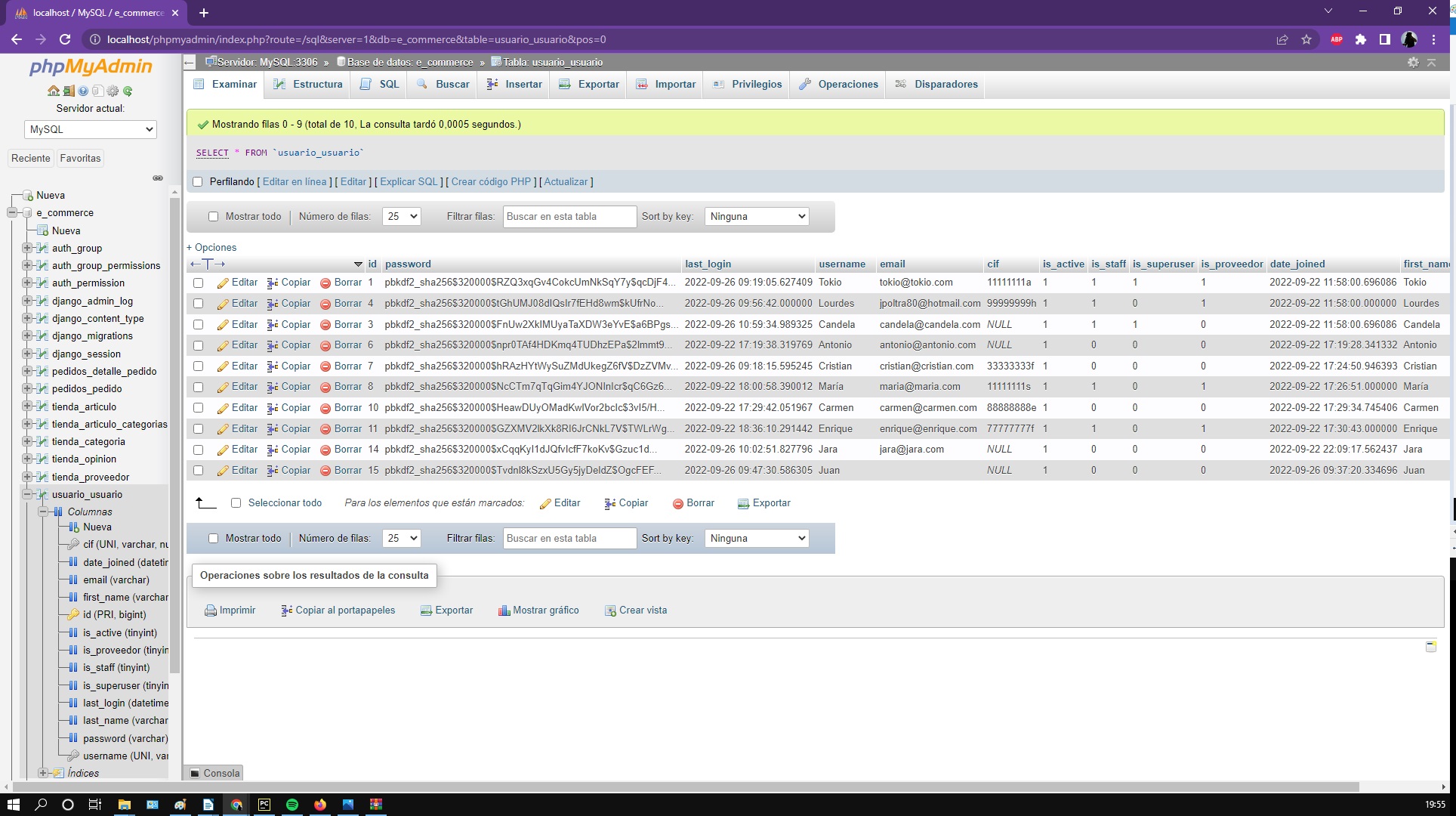
Adjunto capturas de la base de datos con las columnas creadas en naranja los articulos con recuadro en verde.



En las imagenes vemos como desde la base de datos se pueden eliminar usuarios.



Aquí vemos como han desaparecido los dos marcados con anterioridad.



**CONCLUSIONES**

El objetivo del proyecto era hacer una aplicación web para una ecommerce donde se venderían productos electrónicos. La realización de este proyecto conllevaba dos retos.

El primero era poner en práctica todo lo aprendido durante este curso de Programación con Python.

El segundo reto que me propuse era hacerlo con un framework que no habíamos visto en clase, Django, animado también por el especialista en Python(Cristian), el cual nos animaba a explorar otros tecnologías y frameworks, Aparte de utilizar Django también he utilizado HTML, CSS, Bootstrap, MySQL, Highcharts, tecnologías que sólo había usado en las anteriores prácticas del curso. Mi gran reto era enfrentarme solo a un proyecto de estas dimensiones.

Estos objetivos quedan cumplidos ya que cumple con todas las funciones propias de una aplicación web y podría ser utilizado por cualquier ecommerce.

**Adjunto listado de usuarios ya registrados por si es necesario.**

**SUPER:**

Tokio : tokio

Candela : Contraseña

**PROVEEDOR:**

Lourdes : Contraseña

Maria : Contraseña

Enrique **:** Contraseña

**CLIENTES:**

Antonio :Contraseña

Cristian : Contraseña

Carmen : Contraseña

**Acceso a la base de datos:**

usuario: root

contraseña:’ ‘