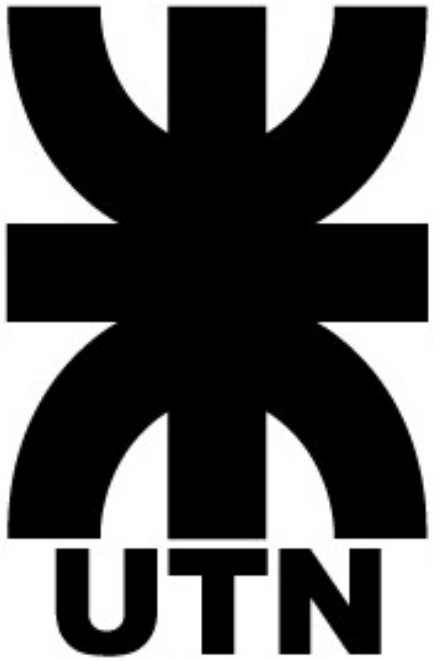


Trabajo práctico integrador - Soporte a la Gestión de Datos con Programación

Profesores:
- Juan Ignacio Torres
- Mario Alejandro Castagnino

Grupo N°5

Año de cursado: 2022

Integrantes:
- Fernando Gardeñes
- Brenda Godoy
- Facundo Sosa
- Liza Strappini

Fecha de entrega: 27/02/23

Índice

Narrativa

Nombre del proyecto: ECOMMERCE

Descripción del proyecto: Un ecommerce es un sitio web que permite a los usuarios comprar productos o servicios en línea a través de Internet. Un ecommerce funciona como una plataforma de comercio electrónico que permite a los usuarios ver, comparar y comprar productos o servicios de forma virtual. Los usuarios pueden acceder al sitio web, buscar los productos que necesitan, agregarlos al carrito de compras y realizar el pago a través de una variedad de opciones, como tarjeta de crédito, transferencia bancaria, PayPal, entre otros.

Objetivo del proyecto: Proporcionar una experiencia de compra fácil y cómoda para los usuarios, permitiéndoles hacer sus compras en línea desde la comodidad de su hogar o desde cualquier lugar donde tengan acceso a Internet. Además, un ecommerce permite a los comerciantes llegar a una audiencia más amplia y expandir su mercado de manera global.

Para lograr este objetivo, los ecommerce deben contar con una serie de funcionalidades que permitan a los usuarios navegar por los productos, realizar búsquedas avanzadas, filtrar resultados, seleccionar productos y hacer transacciones seguras. También deben contar con medidas de seguridad que protejan la información personal y financiera de los usuarios, así como un sistema de atención al cliente que permita resolver dudas, problemas y ofrecer asistencia postventa.

Requerimientos funcionales

- La página principal se encarga de mostrar los productos y permitir al usuario agregar los mismos al carrito de compra, también permitirá acceder a su carrito de compra y a su perfil.
- La pantalla del carrito de compras se encarga de mostrar los productos a comprar con su respectivas cantidades y precio, así también el precio total de la compra y la posibilidad de confirmar la misma.
- La pantalla del checkout permite ingresar toda la información relacionada a la compra como: dirección de envío, método de pago, y datos personales del cliente como mail y teléfono para que el mismo pueda hacer un seguimiento de la compra.
- Se permitirá el registro de pedidos de compra con datos obligatorios incompletos, los cuales podrán completarse posteriormente modificando el pedido. Antes de poder aprobarse los datos del pedido deben estar completos.
- A cada orden se le asignará un identificador único, que será utilizado para identificarla en todos los procesos subsecuentes que se realicen sobre esta.
- Un usuario podrá iniciar sesión en el sistema utilizando su nombre de usuario y contraseña.
- El sistema maneja un registro maestro de clientes. Sólo los usuarios autorizados podrán ingresar nuevos clientes, modificar los datos o eliminarlos.
- El sistema permitirá manejar un maestro de materiales y productos, en el cual se registrarán todos los ítems que se pueden vender. Cada ítem debe tener al menos su nombre, y precio.
- Al seleccionar un ítem, se mostrará su descripción y su precio. Se podrán realizar búsquedas por nombre o familia de material. Para finalizar el registro de la línea, es obligatorio especificar la cantidad.
- En todo momento, el pedido se podrá registrar en borrador, pudiendo ser modificado y registrado en definitivo posteriormente.

Requerimientos no funcionales

Portability

- El sistema debe funcionar correctamente en múltiples navegadores (Sólo Web).
- El sistema debe ejecutarse desde un único archivo .py llamado app.py (Sólo Escritorio).

Security

- Todas las contraseñas deben guardarse con encriptado criptográfico (SHA o equivalente).
- Todas los Tokens / API Keys o similares no deben exponerse de manera pública.
- El sistema no revelará a sus operadores otros datos personales de los clientes distintos a nombres y números de referencia.

Maintainability

- El sistema debe diseñarse con la arquitectura en 3 capas. (Ver checklist_capas.md)
- El sistema debe utilizar control de versiones mediante GIT.
- El sistema debe estar programado en Python 3.8 o superior.

Reliability

- El sistema debe responder rápidamente a las solicitudes de los usuarios, especialmente en la carga de las páginas y en la finalización de las transacciones.
- El sistema debe ser capaz de manejar un aumento en el número de usuarios y en la cantidad de transacciones sin afectar la disponibilidad o el tiempo de respuesta.
- El sistema debe ser capaz de recuperarse de fallas o errores, como la pérdida de conexión a la base de datos o de energía, sin afectar la disponibilidad o la integridad de los datos.
- El sistema debe asegurar que los datos almacenados en la base de datos sean precisos y estén protegidos contra errores o manipulación malintencionada.
- El sistema debe ser seguro y protegido contra amenazas externas, como el robo de información de tarjetas de crédito o el acceso no autorizado a la información del usuario.
- El sistema debe ser fácil de mantener y actualizar, con un mínimo tiempo de inactividad para las actualizaciones y correcciones.
- El sistema debe cumplir con las leyes y regulaciones aplicables, como la protección de datos personales y los derechos de autor.

Scalability

- El sistema debe funcionar desde una ventana normal y una de incógnito de manera independiente (Sólo Web).

- El sistema debe mantener un alto rendimiento incluso cuando hay una gran cantidad de usuarios en línea y realizando transacciones.
- El sistema debe ser capaz de gestionar eficientemente los recursos del sistema, como la memoria y la CPU, para optimizar el rendimiento y reducir los tiempos de respuesta.
- El sistema debe ser capaz de integrarse sin problemas con servicios externos, como servicios de pago, servicios de entrega y servicios de logística, para proporcionar una experiencia de compra sin interrupciones para los usuarios.

Performance

- El sistema debe funcionar en un equipo hogareño estándar.
- El procedimiento de desarrollo de software a usar debe estar definido explícitamente (en manuales de procedimientos) y debe cumplir con los estándares ISO 9000.

Reusability

- El sistema debe ser diseñado en módulos reutilizables que puedan ser fácilmente integrados en diferentes partes del sitio web.
- El sistema debe ser capaz de adaptarse a cambios y evoluciones futuras, sin afectar al resto del sistema.
- El sistema debe seguir estándares reconocidos en la industria para permitir la interoperabilidad y la reutilización de componentes de terceros.
- El sistema debe contar con una documentación clara y detallada de los componentes reutilizables, para facilitar su integración en otras partes del sistema.
- El sistema debe seguir un diseño orientado a objetos para facilitar la reutilización de componentes y la extensión del sistema.
- El sistema debe abstraer los datos del sistema en una capa separada, para permitir su reutilización en diferentes partes del sitio web.
- El sistema debe contar con API claras y bien documentadas, para permitir que otros sistemas se integren con ella y reutilicen sus servicios.
- El sistema debe contar con pruebas rigurosas de calidad para garantizar la calidad y la compatibilidad de los componentes reutilizables en diferentes partes del sistema.

Flexibility

- El sistema debe utilizar una base de datos SQL o NoSQL
- El sistema debe permitir a los usuarios personalizar su experiencia de compra y la presentación de la información de acuerdo a sus necesidades y preferencias.
- El sistema debe ser capaz de adaptarse a diferentes dispositivos, navegadores y resoluciones de pantalla, para proporcionar una experiencia de compra óptima en cualquier dispositivo.
- El sistema debe ser configurable de manera fácil y rápida para agregar o modificar productos, servicios, promociones y opciones de pago.
- El sistema debe ser capaz de crecer y evolucionar para adaptarse a las necesidades del negocio y a los cambios en el mercado.

- El sistema debe ser capaz de integrarse con otros sistemas y plataformas externas, como sistemas de pago y de logística, para mejorar la eficiencia y la experiencia de compra para los usuarios.
- El sistema debe estar diseñado en módulos reutilizables y fáciles de integrar, para permitir la evolución y adaptación del sistema.
- El sistema debe permitir a los usuarios seleccionar su idioma preferido, para proporcionar una experiencia de compra personalizada y global.

Stack tecnológico

El proceso de desarrollo se gestionará por medio de la herramienta Django.

Capa de Datos

La base de datos a utilizar es SQLite, y la ORM que provee Django.

Capa de Negocio

La capa de negocio será encargada de manejar toda la lógica de negocio relacionada con la venta y el procesamiento de pedidos de ropa en línea. Algunas de las tareas

Gestión de catálogo: Encargada de manejar la información y la lógica relacionada con el catálogo de productos. Esto incluye el almacenamiento y la recuperación de información sobre los productos, como su descripción, precios, talles, colores, sexo y demas. Los métodos que nos permiten hacer este manejo de la información están compuestos por una función que hace un request a la página donde accedemos a nuestra base de datos y traemos lo que necesitamos, mostrando el html que necesitamos por medio del método render.

Gestión de clientes: La capa de negocio podría encargarse de la gestión de clientes, incluyendo el almacenamiento de información de los clientes, el seguimiento de sus pedidos anteriores y el manejo de su historial de compras.

Capa de Presentación

El framework utilizado es Django, por lo que las diferentes capas de presentación se manejan a través de las views de Django que procesan las solicitudes del usuario y renderizan el código HTML que usamos para mostrar la información que queremos traer. Algunas de las que aplicamos para nuestra tienda Online fueron las que nos permitian:

- Agregar, eliminar, modificar cantidad productos del carrito
- CRUD de usuario, mediante nuestros métodos login, register y logout
- Método filter para que los usuarios puedan buscar los productos que quieren a través de los filtros que les proporcionamos
- En página principal, cargar la información de los productos junto con su imagen

Reglas de negocio

- **Registro de usuarios:** Se deben definir las condiciones para que los usuarios puedan registrarse, tales como requisitos de información personal y de contacto, y la validación de su cuenta de usuario.

- **Catálogo de productos:** Se deben establecer las categorías y subcategorías de productos que se ofrecen en la tienda, así como las características y especificaciones de cada uno de ellos.

- **Proceso de compra:** Se debe definir el proceso de compra de principio a fin, desde la selección de los productos hasta la confirmación de la compra, pasando por la forma de pago y los plazos de entrega.

- **Política de envío y devoluciones:** Es importante establecer las condiciones y los plazos de envío de los productos, así como los procedimientos para gestionar devoluciones y reembolsos.

- **Precios y promociones:** Se deben definir los precios de los productos y las promociones que se ofrecen a los usuarios, incluyendo descuentos y cupones.

- **Seguridad de la información:** Es esencial garantizar la seguridad de la información de los usuarios, incluyendo la protección de datos personales y de los métodos de pago.

- **Servicio al cliente:** Se debe establecer una política de servicio al cliente que permita atender las consultas y reclamos de los usuarios de manera efectiva y en un plazo razonable.

- **Administración de inventario:** Se debe establecer un sistema de administración de inventario que permita llevar un registro actualizado de los productos disponibles y gestionar las existencias de manera eficiente.

Casos de uso principales

- **Registro de usuario:** El usuario debe tener la posibilidad de registrarse en la plataforma ingresando sus datos personales y de contacto.

- **Búsqueda de productos:** Los usuarios deben poder buscar los productos que desean adquirir a través de diferentes criterios, como nombre, categoría, marca, etc.

- **Selección de productos:** El usuario debe poder seleccionar los productos que desea adquirir y agregarlos a su carrito de compras.

- **Proceso de compra:** El usuario debe poder realizar la compra de los productos seleccionados a través de diferentes métodos de pago, como tarjeta de crédito, transferencia bancaria, etc.

- **Seguimiento de pedidos:** El usuario debe tener la posibilidad de realizar un seguimiento del estado de sus pedidos, incluyendo información sobre el envío y la entrega.

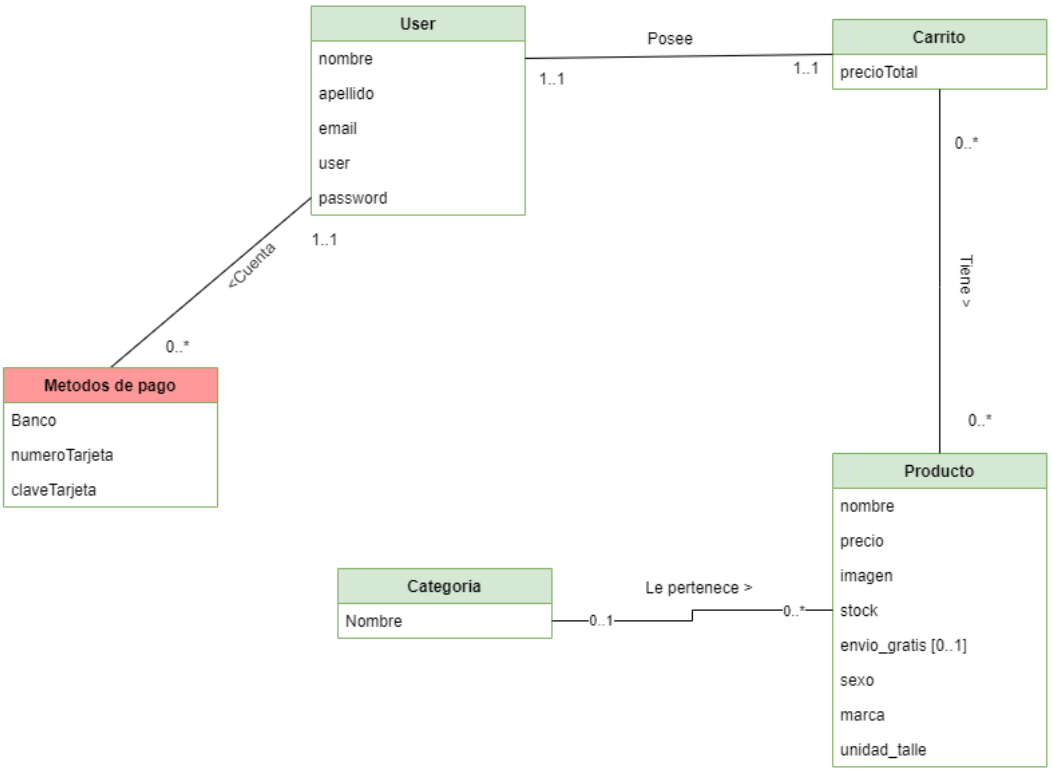
- **Devoluciones y reembolsos:** El usuario debe poder solicitar devoluciones y reembolsos en caso de que los productos no cumplan con sus expectativas o lleguen en mal estado.

- **Servicio al cliente:** El usuario debe tener la posibilidad de contactar al servicio al cliente de la plataforma para resolver dudas, hacer consultas o reportar problemas.

- **Administración de inventario:** Los administradores de la plataforma deben tener la capacidad de administrar el inventario de productos, añadir nuevos productos, actualizar precios y promociones, entre otras funciones.

- **Análisis de ventas:** Los administradores de la plataforma deben tener la capacidad de realizar análisis de ventas y estadísticas de la plataforma para poder tomar decisiones informadas sobre el negocio.

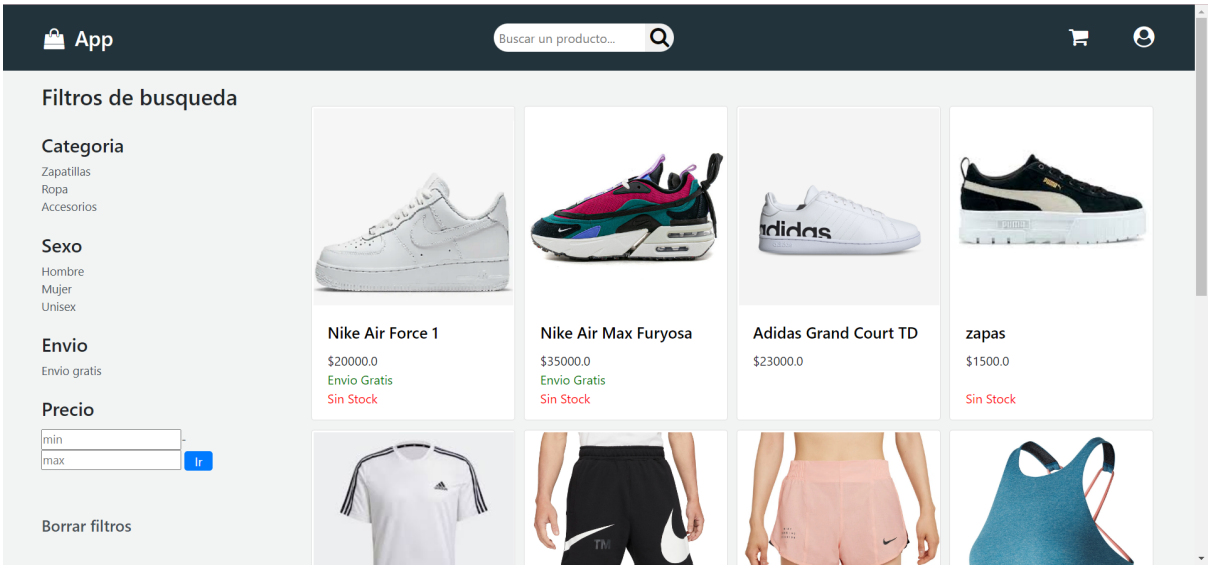
Modelo de dominio



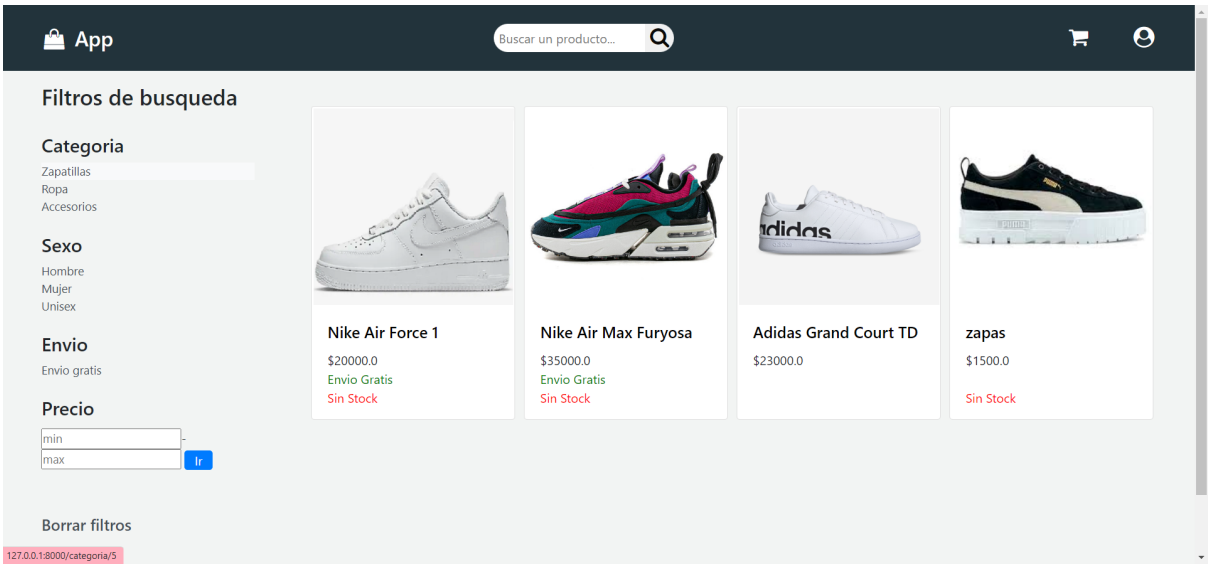
La clase “Metodos de pago” no se implementó por temas de imposibilidad, pero se agregó al MD para que quede completo

Imágenes de las pantallas

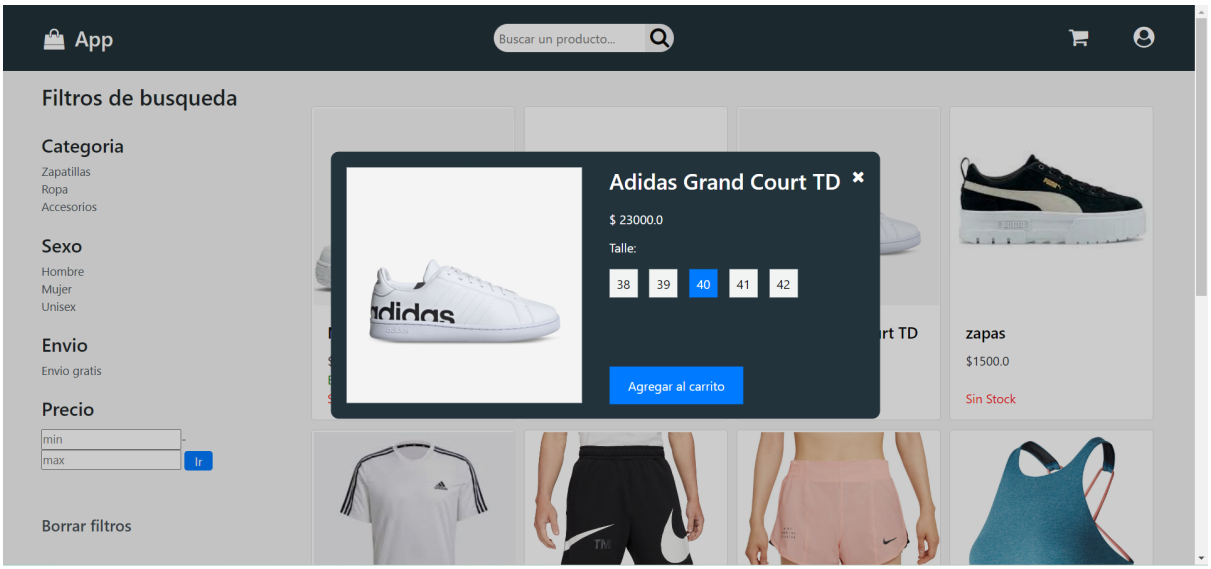
Inicio



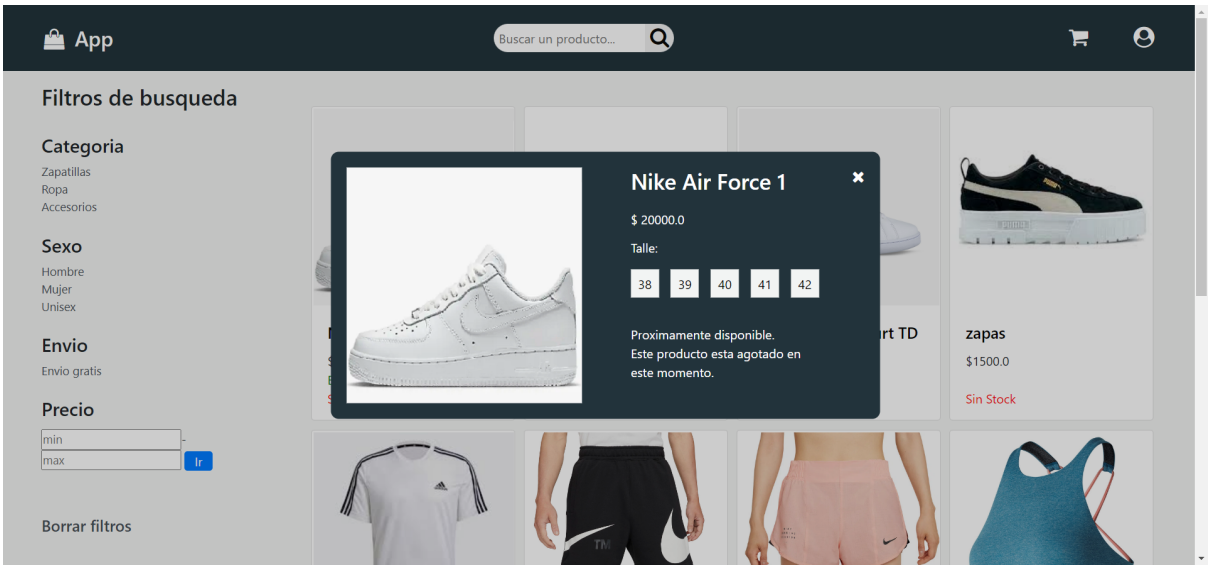
Inicio por categoría (zapatillas)



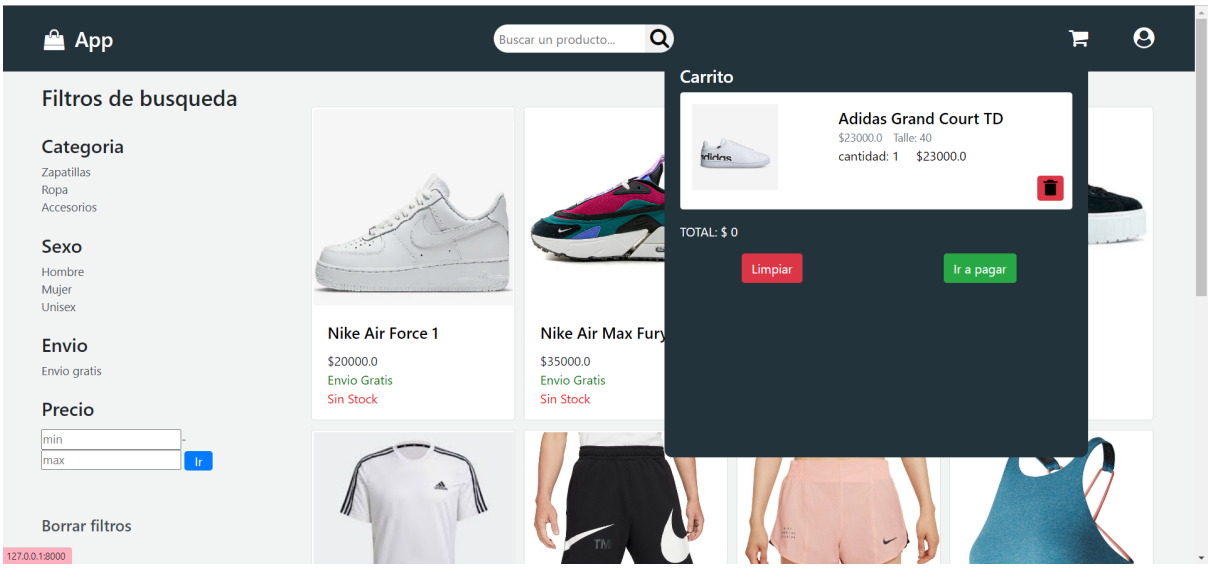
Vista individual del producto (con stock)



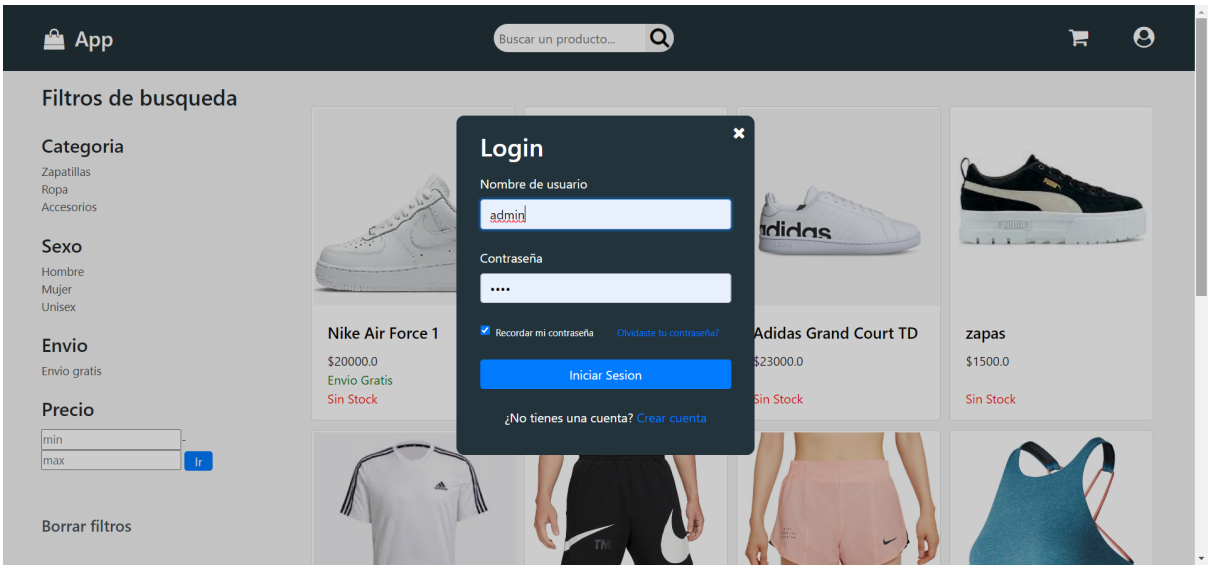
Vista individual del producto (sin stock)



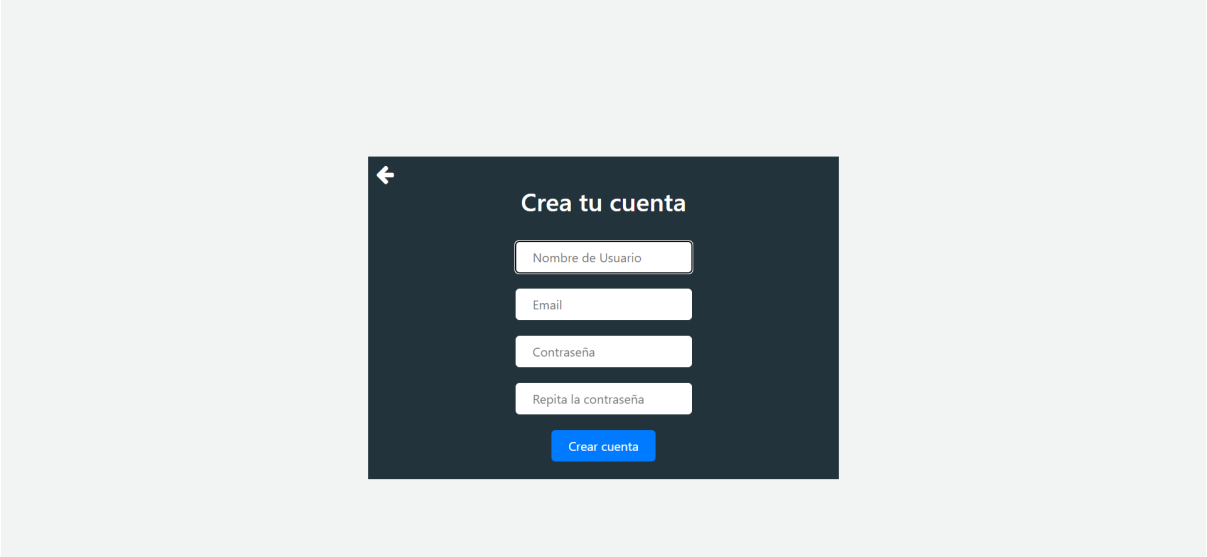
Carrito de compras



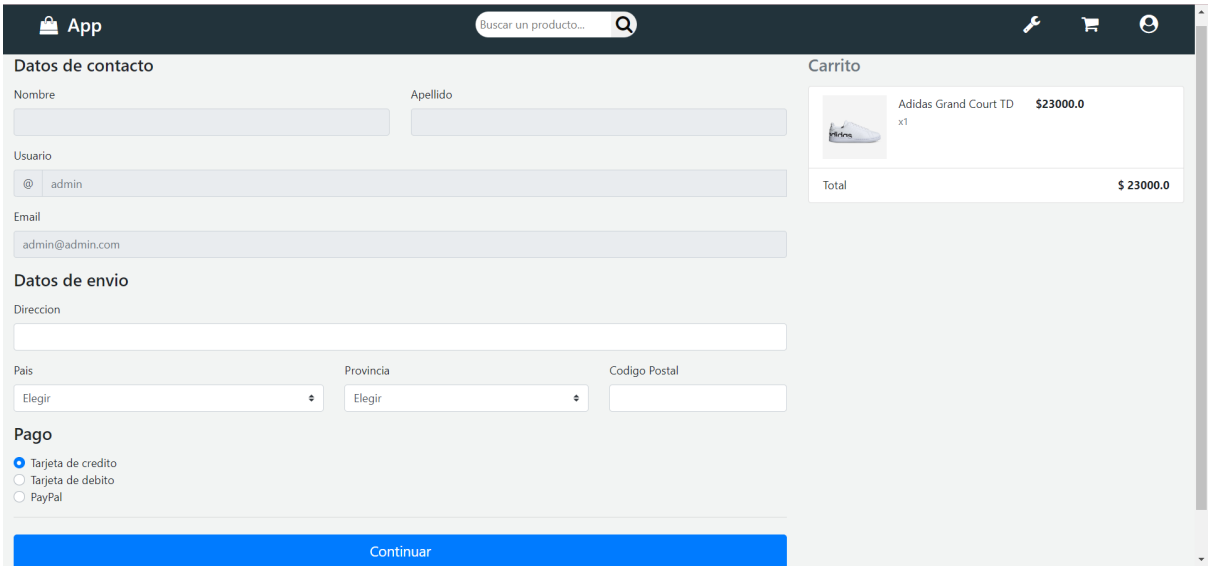
Login



Registrar una nueva cuenta



Checkout



Link a código fuente

https://github.com/facuso162/ecommerce.git

Link a la pagina: http://lzs1999.pythonanywhere.com/

Bibliografía

- https://www.djangoproject.com/ → Pagina oficial de Django
- https://es.stackoverflow.com/ → Utilizada para resolver dudas de codigo