

Proyecto Programado 2

TEC Shop

1. Descripción

Las empresas que deseen tener una presencia en internet más efectiva no se pueden conformar con una página corporativa que tan sólo proporcione información. Actualmente, el comercio electrónico se ha revelado como una gran vía de negocio para aquellos que sepan aprovechar sus posibilidades.

El comercio electrónico o e-commerce consiste fundamentalmente en el desarrollo de acciones de mercado, ventas, servicio al cliente, gestión cartera, gestión logística y en general, todo evento comercial e intercambio de información llevado a cabo por medio de internet.

Los consumidores ya están totalmente acostumbrados a realizar sus compras online y los procedimientos de pago son cada vez más diversos y seguros. Esto acerca mucho más al consumidor a la experiencia de compra en internet, con más confianza y satisfacción.

Una tienda que pone al servicio de sus clientes las ventas en línea se asegura las siguientes ventajas: creación de oportunidades de negocio y nuevas formas de distribución de sus productos y servicios; acceso a clientes de cualquier zona geográfica, control de pedidos y clientes, entre otros.

Se ha decidido implementar una pequeña parte de un sistema de comercio electrónico, llamado TEC Shop y a usted le corresponde realizar dicha implementación.

2. ¿Qué se busca con esta tarea?

- Practicar las habilidades de resolución de problemas.
- Aumentar el conocimiento del estudiante sobre el lenguaje de programación Python.
- Practicar la experimentación y la resolución de problemas (divide y vencerás).
- Ejercitar la toma de decisiones.
- Fomentar la **investigación** por parte del estudiante sobre como:

- Interfaz gráfica en Python.
- Utilizar interfaces de programación de aplicaciones para agregar funcionalidades al sistema a desarrollar .
- Utilizar la estructura de datos Lista (gestión de los elementos de una lista).
- Implementar estructuras de control de flujo básicas.
- Utilizar mecanismos para solicitar datos al usuario.
- Fomentar el trabajo cooperativo y colaborativo.

3. Por hacer

Se deberá implementar en Python un programa para gestionar productos y realizar compras.

El sistema contará con dos vistas que tendrán las funcionalidades que se detallan a continuación:

3.1 Vista de administrador

Esta vista permite que el dueño de la tienda gestione toda la información relacionada con los artículos que estarán a la venta, también por medio de esta vista el usuario podrá consultar información de las ventas realizadas.

Las funcionalidades que se deben implementar en esta vista son:

3.1.1. *Gestión de categorías:*

Agregar, modificar, eliminar y consultar categorías. Algunos ejemplos de categorías son: electrónicos, ropa, zapatos, juguetes, entre otros.

3.1.2. *Manejo de información de artículos:*

Se deberá manejar para cada artículo el identificador del producto, el nombre, la categoría, la descripción, el precio unitario sin impuestos y una foto del artículo.

Cuando el programa se inicia deberá leer la información antes mencionada de forma automática desde un archivo de texto. El usuario podrá agregar nuevos artículos, modificar o eliminar artículos que estén registrados, cada una de estas operaciones deberán actualizar el archivo de texto en el que se almacena la información de los artículos registrados en el sistema.

3.1.3. Búsqueda avanzada:

El usuario administrador puede visualizar la información de todos los artículos registrados en el sistema, la búsqueda la puede realizar por los siguientes filtros:

- Por palabra clave: esto realiza la búsqueda de todos los artículos que contengan dicha palabra ya sea en la categoría o en el nombre.
- Por precio: esto realiza la búsqueda por un rango de precios determinado.

Las búsquedas se pueden realizar por uno o ambos filtros a la vez. Si el usuario no indica ninguno de los filtros, el sistema debe mostrar la información de todos los artículos registrados en el sistema.

A continuación se muestra un ejemplo de cómo debe visualizarse el resultado de la búsqueda en el sistema:

The screenshot shows a search results interface. On the left, there are two filter sections. The first section, 'Show results for', includes a category list under 'Cell Phones & Accessories' with items like 'Carrier Cell Phones (51)', 'Cell Phones (254)', 'Unlocked Cell Phones (203)', 'Accessories (191)', 'Cell Phone Cases (2,480)', 'Car Cradles & Mounts (23)', 'Charging Stations (7)', 'Cases, Holsters & Clips (3,105)', 'SIM Cards & Prepaid Minutes (1)', and a 'See Less' link. The second section, 'Price', shows price ranges from 'Under \$10 (791)' to '\$100 & Above (161)', with input fields for a custom range and a 'Go' button. On the right, two product listings are shown. The first is an 'LG Tribute HD Prepaid Carrier Locked - Retail Packaging (Boost Mobile)' priced at '\$68⁹⁹' with a 'Show description' button. The second is a 'Samsung Galaxy J7 No Contract Phone, 16GB ROM / 2GB RAM, 5.5 Inch - White - (Boost Mobile)' priced at '\$169⁹⁵' with a 'Show description' button. Below the screenshot, two boxes with arrows point to the filter and product sections. The left box is labeled 'Sección de filtros' and the right box is labeled 'Listado de productos'.

En el listado de productos se debe mostrar la foto, el nombre del producto y el precio. Si el usuario desea ver todo el detalle del producto debe brindarse la opción de hacer la consulta y mostrarse algo similar a lo siguiente:



En esta pantalla debe mostrarse toda la información registrada para el artículo seleccionado, además se debe indicar el precio sin impuestos y el precio con impuestos (para ese caso se va a asumir que al artículo solo se le agrega el 13% de impuesto de ventas).

En la vista avanzada el sistema también debe permitir que el administrador consulte la información de los artículos adquiridos por un usuario, para realizar esta consulta el usuario debe indicar el correo electrónico del cliente y el sistema le mostrará la lista de los artículos comprados por ese cliente.

3.2 Vista de comprador

Esta vista permite que un cliente pueda consultar todos los productos disponibles en la tienda y pueda agregar alguno ya sea a la lista de deseos o a un carrito de compras. En esta vista el comprador también podrá realizar búsquedas avanzadas y consultar su historial de compras.

Las funcionalidades que se deben implementar en esta vista son las siguientes:

3.2.1. *Visualización de artículos*: el sistema debe permitir que el comprador consulte todos los artículos que están disponibles en la tienda, esta función debe mostrar una pantalla similar a la descrita en la sección 3.1 de este documento.

3.2.2. *Gestión de listas*: esta funcionalidad debe permitir que el comprador gestione dos listas diferentes: la lista de deseos y el carrito de compras.

El comprador podrá seleccionar un artículo y agregarlo en una lista de deseos, además puede eliminar artículos de la lista de deseos o consultar todos los artículos que tiene actualmente en dicha lista.

En caso de que el usuario ya esté decidido a comprar el artículo, el sistema le debe permitir que agregue este artículo a un carrito de compra e indicar las unidades que desea de dicho artículo. El usuario puede eliminar artículos del carrito de compra o consultar los artículos que se han agregado.

Para finalizar la compra el sistema le solicitará al usuario un correo electrónico, se enviará al correo suministrado una confirmación con el detalle de todos los artículos que estaban en el carrito. En este correo se debe indicar el nombre del artículo, la categoría, descripción del artículo, la cantidad de unidades adquiridas de ese artículo, el monto unitario sin impuestos, el monto unitario con impuestos. Por último, en el correo se debe indicar el monto total a pagar y el número de cuenta bancaria a la que se debe aplicar el débito para hacer la venta.

Esta venta deberá registrarse en el historial del cliente.

3.2.3. Historial

El sistema debe mostrarle al cliente el historial de todas las compras registradas en el sistema, para consultar el historial se debe indicar el correo electrónico del usuario y esto mostrará el detalle de las compras realizadas.


Este detalle se debe visualizar de forma cronológica descendente, es decir primero se visualizan las compras más recientes y por último las compras más antiguas.

A continuación se muestra una pantalla de ejemplo de cómo debe visualizarse esta funcionalidad:

ORDER PLACED
December 31, 2016

TOTAL
\$8.99

Delivered Jan 26, 2017
Your package was delivered.



Deyoung 2nd HDD SSD Hard Drive Caddy Adapter for Toshiba Satellite C655 C655D C660 C660D

Sold by: HDD/SSDCaddy


\$8.99

Buy it again

ORDER PLACED
December 31, 2016

TOTAL
\$7.77

Delivered Jan 11, 2017
Your package was delivered.



Genuine OEM Google Nexus 4 E960 2100mAh Battery BL-T5 LG Optimus G E970 E973 LS970

Sold by: iDeal Xpert

\$7.77

Buy it again

3.2.4. *Búsqueda avanzada:*

El usuario comprador puede visualizar la información de todos los artículos registrados en el sistema, la búsqueda la puede realizar por los siguientes filtros:

- a. Por palabra clave: esto realiza la búsqueda de todos los artículos que contengan dicha palabra ya sea en la categoría o en el nombre.
- b. Por precio: esto realiza la búsqueda por un rango de precios determinado.

Las búsquedas se pueden realizar por uno o ambos filtros a la vez. Si el usuario no indica ninguno de los filtros, el sistema debe mostrar la información de todos los artículos registrados en el sistema.

El funcionamiento de esta búsqueda es igual al descrito en la sección 3.1.

Funcionalidad que otorga puntos extra

Si se incorpora seguridad al sistema se otorgarán 5 puntos extra. La seguridad consiste en validar el usuario que entra al sistema por medio de su correo electrónico y contraseña, además se debe validar el tipo de usuario, para determinar si es un usuario administrador o comprador y mostrarle solo la vista correspondiente.

4. Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación Python. En caso de requerir librerías externas adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación. En la documentación se debe incluir la línea de comandos para compilar y ejecutar el código. En la revisión, se debe compilar y ejecutar el proyecto por línea de comandos, de lo contrario no se revisará el proyecto.

5. Puntos a ser evaluados

1. Correctitud de la solución computacional.
2. Robustez y resiliencia de la solución computacional.
3. Apego de la solución a los principios del programador.
 - a. El principio del carácter personal.
 - b. El principio de la estética.
 - c. El principio de la claridad.
 - d. El principio de la distribución.

- e. El principio de lo explícito.
 - f. El principio del código auto documentado.
 - g. El principio de los comentarios.
 - h. El principio de volver atrás.
 - i. El principio de El TIEMPO y EL DINERO de otros.
4. Evitar los síntomas de un diseño pobre “olores del software”.
- a. Rigidez.
 - b. Fragilidad.
 - c. Inmovilidad.
 - d. Viscosidad.
 - e. Complejidad innecesaria.
 - f. Repetición innecesaria.
 - g. Opacidad.

6. Evaluación

- 1. Documentación externa 30%
- 2. Pantalla inicial 5%
- 3. Vista de administrador: 25%
 - a. Gestión de categorías 5%
 - b. Manejo de información de artículos 6%
 - c. Persistencias de datos 7%
 - d. Búsqueda avanzada del administrador 7%
- 4. Vista del comprador: 35%
 - a. Visualización de la tienda 7%
 - b. Gestión de lista de deseos 3%
 - c. Carrito de compras 7%
 - d. Envío de correo 6%
 - e. Historial 6%
 - f. Búsqueda avanzada 6%
- 5. Puntos extra: Gestión de usuarios 5%

6. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, y restricciones.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: instrucciones de compilación, ejecución y uso.
 - Debe indicar como correr el programa paso a paso (el archivo y como correrlo), cuales funciones están disponibles y cuál es su funcionamiento. Debe incluir los requisitos para correr el programa, tales como: versión de Python, librerías externas, etc. Debe guiar al usuario para poder utilizar el programa exitosamente. Hacer uso de **screenshots** y comentarios.
3. Pruebas de funcionalidad.
 - Debe demostrar evidencia de todas las funcionalidades implementadas en la tarea programada. Para esto realice la ejecución de su tarea programada y tome **screenshots** dejando evidencia del resultado de la ejecución, según los requerimientos esperados. Debe mostrar evidencia también de la robustez de la aplicación (por ejemplo, qué pasa cuando el usuario ingresa un dato quivocado).
4. Descripción del problema: Qué se tiene que resolver y cómo se va hacer.
5. Bitácora y lecciones aprendidas.
 - Debe crear una bitácora (del programador) de forma individual (en **www.blogger.com**) donde quede registro **cronológico** de la toma de decisiones, nuevos aprendizajes producto de las investigaciones, puntos importantes, y otros elementos que el programador considere relevantes de incluir en su bitácora personal del programador.
En el documento deberán incluir dos vínculos que permitan acceder las bitácoras personales. Asegúrese que los vínculos permitan acceder a su bitácora directamente sin necesidad que el/la profesor/a deba registrarse o autenticarse en la herramienta que usted decidió utilizar para crear la bitácora.
6. Principios.
 - Demostrar evidencia que cumplió con cada uno de los principios del programador. La evidencia se debe sustentar con segmentos de código de la tarea programada (nuevamente tome **screenshots**). Además debe sustentar sus screenshots con comentarios relevantes para cada caso.
7. Olores del software.
 - Debe indicar cuál o cuáles estrategias se usaron para apartarse de cada uno de los olores del software vistos en clase. Las estrategias deben sustentarse con

segmentos de código de la tarea programada (tome **screenshots**) y con documentación de las decisiones que fueron tomadas durante el diseño y la implementación.

8. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).

7. Aspectos administrativos

1. Debe crear un archivo .zip (“TP2.zip”) que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **solucion_computacional**, en la primera deberá incluir el documento de *word* o *open office* (no pdf) solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea. El archivo **info.txt** debe contener la siguiente información (calidades):
 - a. Nombre del curso
 - b. Número de semestre y año lectivo
 - c. Nombre del Estudiante x2
 - d. Número de carnet x2
 - e. Número de tarea programada
 - f. Fecha de entrega
 - g. Estatus de la entrega (definido por el responsable de la implementación de la tarea, debe ser **CONGRUENTE** con la solución entregada):
[Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]
2. Deberá subir el archivo antes mencionado al TEC Digital en el curso de TALLER DE PROGRAMACIÓN GR 02, en la asignación llamada “Tarea Programada 2” debajo del rubro de “Proyectos programados”. Se deberá subir el archivo antes de las 11:50 pm del domingo 14 de mayo del 2017.
3. Dentro de la carpeta de **solucion_computacional**, deberá incluir un archivo .txt (**PrimerNombreMiembro1.PrimerNombreMiembro2..txt**) que contenga todo el texto de la solución del o de los archivos .py presentados (las implementaciones). Este archivo puede ser revisado en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100% original, y no se va a tolerar el plagio.** Este archivo también deberá ser subido al TEC Digital en el curso de TALLER DE PROGRAMACION GR 02, en la asignación llamada “Tarea Programada 2 (archivo TXT)” debajo del rubro de “Proyectos programados”. Se deberá subir el archivo siguiendo los mismos rangos de tiempo que los descritos en el punto anterior.
4. Las tareas deberán ser revisadas con el profesor o el asistente. Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente. La nota de la revisión es individual, el resto de la nota es grupal.

5. La tarea será revisada en la versión de Python 3.5.3 únicamente.
6. La tarea se hará en grupos de 2 solamente.
7. La tarea vale un 20% de la nota del curso.
8. Antes de la entrega del proyecto se pasará en clase una lista donde deben anotar los miembros de su equipo de trabajo. Opcionalmente, pueden mandar un correo electrónico con el asunto “[Taller 117] PP2:GRUPO” e indicar los nombres de las personas que conforman el grupo de trabajo para esta tarea programada. Esta información es necesaria para crear los grupos en el TEC Digital. Si no se anotan en la lista, o no envían el correo, se calificará la tarea de forma individual al miembro del equipo que la suba.