TP FINAL

## PYTHON3 NIVEL INCIAL

## “Control de stock para pañalera”



**Profesor:**[Gabriela Verónica Aquino](https://campus.utnba.centrodeelearning.com/user/view.php?id=119125&course=1)

**Profesor:**[juan Barreto](https://campus.utnba.centrodeelearning.com/user/view.php?id=9127&course=1)

**Profesor:** [Brenda Abigail Barreto Aquino](https://campus.utnba.centrodeelearning.com/user/view.php?id=167383&course=1)

**Alumno:** Federico Saraullo

Tabla de contenido

**Introducción3**

**Arquitectura de la aplicación4**

**Modo de uso de la aplicación4**

**Especificaciones del código4**

# introducción

El objetivo de la aplicación es servir como control de stock de pañaleras, o cualquier comercio que requiera una aplicación de este tipo.

La idea general es crear una aplicación sencilla donde en base a una serie de botones se pueda dar de alta, baja, actualizar y consultar los productos. Cada producto contará de cuatro campos, “producto”, “stock”, “costo” y “precioventa”.

La información se guardará en una base de datos SQL y para generar la interfaz gráfica utilizaremos la librería tkinder de Python.

Imagen que contiene Tabla

Descripción generada automáticamente

El uso de la aplicación es muy sencillo, bastará con completar los campos en la parte superior para realizar las acciones mencionadas anteriormente.

Además, se podrá exportar la información a un archivo csv en caso de ser necesario.

# Arquitectura de la aplicación

En esta sección se detallará a alto nivel como fue diseñada la aplicación.

Funcionamiento:

Diagrama

Descripción generada automáticamente

Cómo se puede observar en la imagen previa hay 4 módulos donde cada uno tiene una función específica.

Módulo run.py:

Este es el módulo principal y es donde se inicia el programa mediante el comando **python3 run.py** desde la consola. Este módulo será el encargado de inicializar la base de datos como también construir la GUI mediante la librería de tkinder. Haciendo de uso de la librería tkinder, se generarán eventos donde se darán altas, bajas, updates y consultas de productos, como así también se podrán observar los mismos en la pantalla. Finalmente se podrá exportar la información a un archivo .csv.

Módulo dbservice.py:

Este será el módulo encargado de realizar todas las operaciones en la base de datos.

Módulo tkinderservice.py:

Este módulo es utilizado exclusivamente por dbservice.py. En el caso que se quiere eliminar algún producto inexistente o el ingreso de productos no sea en el formato requerido, entre otras, este módulo abrirá las ventanas emergentes necesarias.

Módulo util.py:

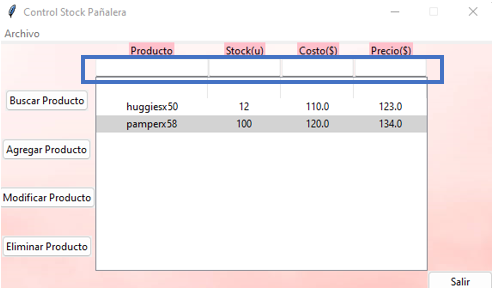
El módulo útil.py (o módulo de utilidades varias) es el encargado de exportar la información de la base de datos hacia un archivo csv.

# Modo de uso de la aplicación

En esta sección se detallará el modo de uso de la aplicación.

**Alta de producto:**

Para realizar el alta de un producto habrá que rellenar los campos dentro del rectángulo azul:



Solamente es obligatorio rellenar el campo producto, siendo los demás opcionales.

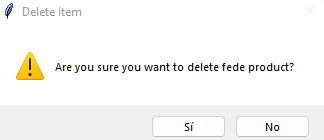
Para el campo producto solo se podrán ingresar letras y números, en stock solo números y en los campos costo y precio números o formato decimal (3.23 por ejemplo).

Una vez llenados los campos, se requiere que se presione el botón *“Agregar Producto”*

**Baja de producto:**

Es similar al alta de un producto, solo es necesario agregar el nombre del producto en el campo producto y luego presionar el botón “*Eliminar Producto*”.

Una ventana emergente preguntará si está seguro y en caso de estarlo presionar “si”.



Si no se recuerda el nombre del producto, presionar el botón “*Buscar producto*” para listar todos los ítems guardados.

**Modificación de producto:**

Para modificar un producto será necesario colocar el nombre del producto y luego modificar los valores que crea conveniente en los demás campos. Finalmente presionar el botón “*Modificar Producto*” para que el cambio tome efecto.

**Consulta de producto:**

Se podrá consultar un producto en particular o la lista entera de los mismos. Si la aplicación detecta algo escrito en el campo Producto y se encuentra en la base de datos, se imprimirá en la pantalla los datos de ese producto en particular. En caso de no colocar nada, se imprimirá la lista entera con todos los productos. Si el producto no es encontrado, solamente se avisará al usuario del mismo.

**Exportar información a archivo:**

Para exportar la información guardada en la aplicación, en el menú superior clickear sobre *Archivo* y luego *Exportar a csv*. Se genera un archivo en llamado con el formato “*export + fecha*”

# Especificaciones del código

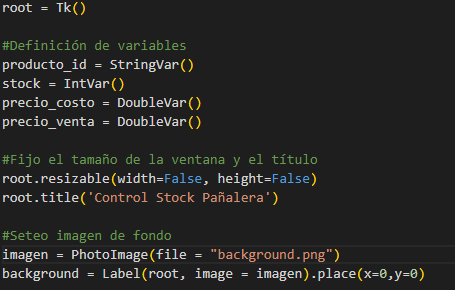
**Modulo run.py**

Inicialización de la base de datos

Texto

Descripción generada automáticamente con confianza media

Creación objeto “tkinder”, variables utilizadas, tamaño de la ventana y fondo utilizado

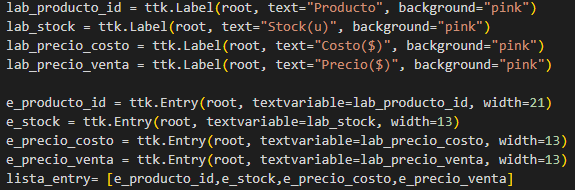


Creación del Menú Bar y las funciones asociadas a los botones del menú

Texto

Descripción generada automáticamente

Creación de las etiquetas y labels:



Creación de los botones y las funciones asociadas

Escala de tiempo

Descripción generada automáticamente

Configuración del treeview:

Texto

Descripción generada automáticamente

Ubicación de todos los elementos creados anteriormente en la ventana

Texto

Descripción generada automáticamente

**dbservice.py**

El módulo dbservice está compuesto por diversas funciones las cuales se detallarán a continuación

* conexión()

Utilizado para conectarse con la base de datos.

Texto

Descripción generada automáticamente

* actualizar\_tree()

Pasarle la información actualizada de la base de datos al modulo tkinderservice.py para actualizar el treeview

Texto

Descripción generada automáticamente

* crear\_tabla()

Crea la tabla en la base de datos, pero chequea previamente si la misma existe.

Texto

Descripción generada automáticamente

* alta()

Utilizada cuando el botón “Agregar Producto” fue presionado. Utiliza tres sentencias Regex para verificar la entrada de datos:

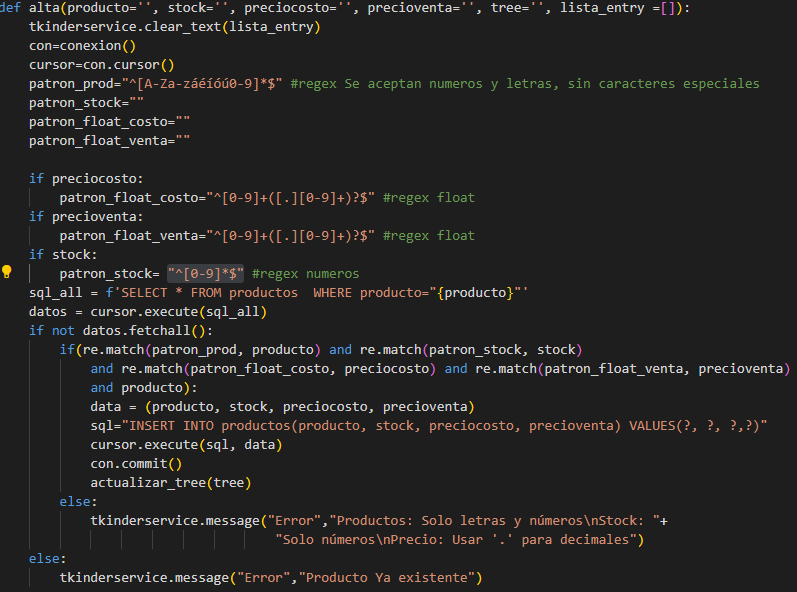
patron\_prod = "^[A-Za-záéíóú0-9]\*$"

patron\_float\_costo = "^[0-9]+([.][0-9]+)?$"

patron\_float\_venta = "^[0-9]+([.][0-9]+)?$"

patron\_stock ="^[0-9]\*$"

Una vez que se verifica que los datos de entrada son correctos, la función se encargará de guardar los datos en la base de datos. En caso de que el producto ya se encuentre en la misma, una ventana emergente dará aviso de lo mismo y no tendremos productos duplicados.

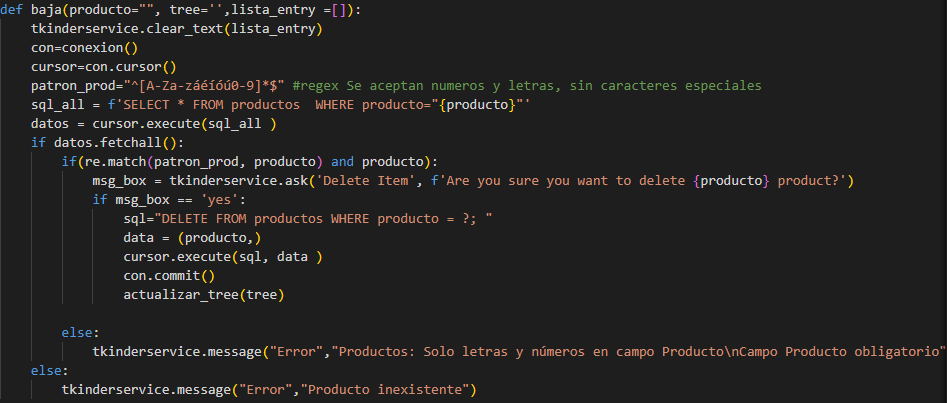


* baja()

Utilizada cuando el botón “Eliminar Producto” fue presionado. Utiliza solo una sentencia Regex porque solo verificará el campo “Producto”:

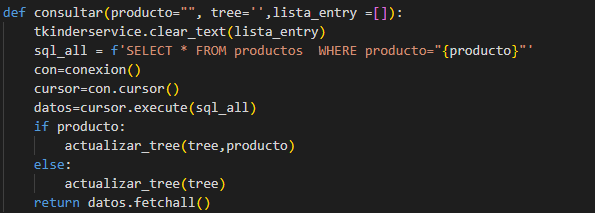
patron\_prod = "^[A-Za-záéíóú0-9]\*$"

En caso de querer borrar un producto inexistente, una ventana avisará sobre la situación



* consultar()

Utilizada cuando el botón “Buscar Producto” fue presionado. Solo verificará el campo “Producto” y mediante una Query en la DB verificará su existencia. En caso de no colocar nada en el campo “Producto” el resultado será la lista completa de los productos.



* modificar()

Utilizada cuando el botón “Modificar Producto” fue presionado. Utiliza tres sentencias Regex para verificar la entrada de datos:

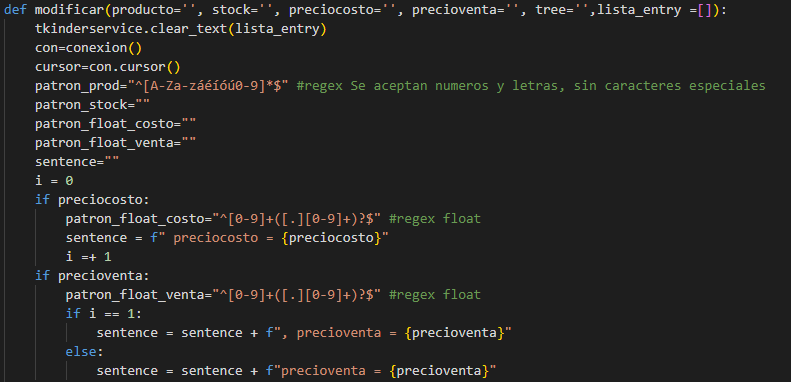
patron\_prod = "^[A-Za-záéíóú0-9]\*$"

patron\_float\_costo = "^[0-9]+([.][0-9]+)?$"

patron\_float\_venta = "^[0-9]+([.][0-9]+)?$"

patron\_stock ="^[0-9]\*$"

Una vez verificado el formato de los valores de entrada, la función modificará los campos necesarios del producto.



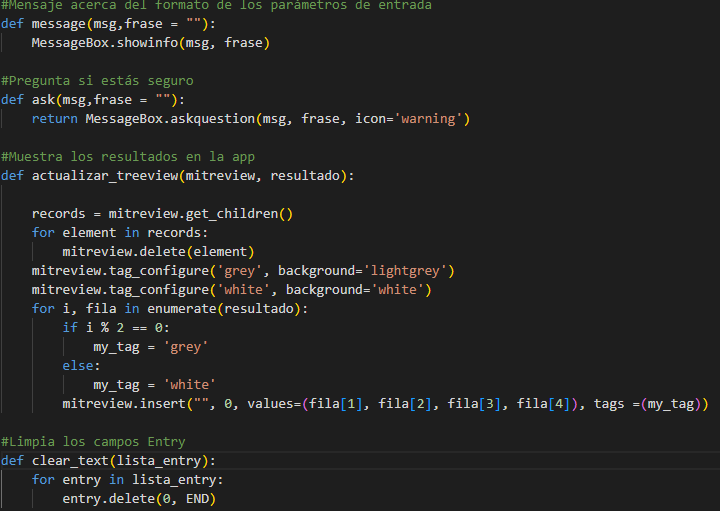
Texto

Descripción generada automáticamente

**tkinderservice.py**

Este servicio/módulo tiene varias funciones que usan la librería de tkinder. El objetivo de este módulo es que las funciones relaciones con tkinder estén separadas del módulo de base de datos.

Las funciones son las siguientes:



ask(): Utilizado para preguntar si el usuario está seguro de borrar un producto

actualizar\_treeview(): Cada vez que se actualiza la base de datos esta función es llamada para actualizar el treeview en la ventana de la aplicación.

message(): Utilizada para mensajes de aviso.

clear\_Text(): Utilizado para borrar lo que hay en los box para el ingreso de datos.

**utils.py**

Este módulo solo tiene una función, que exporta la información de la base de datos a un archivo .csv

