UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y SU NATURALEZA

TEMA: Sistema de inventario de productos farmacéuticos

CÁTEDRA: Algoritmos II.

INTEGRANTES:

* **Stefany Michel Díaz Martínez DM01134627.**

***Flor de María Córdova Elías. CE01134671**

***Marvin Fernando Pérez Ramírez. PR01134638**

CATEDRÁTICO: Lic. Pedro Noble.

SAN SALVADOR, 17 DE ABRIL DE 2018.

INTRODUCCIÓN

En el presente informe mostraremos la importancia de realizar un sistema de inventarios ya que son de gran importancia porque son parte vital de un negocio, puede ser industria, un supermercado, una tienda departamental o cualquier tipo de empresa.

Mostrando los principales aspectos que hay que tomar en cuenta en un sistema de inventario es: cuanto ordenar y también cuanto nos cuesta pedir una orden y cuanto nos cuesta mantener esos productos en orden.

Crearemos un sistema que nos facilite el ordenamiento de los productos lo cual es de mucha utilidad para las empresas.

OBJETIVOS.

OBJETIVO GENERAL.

Presentar los avances en el proceso de creación del proyecto, para verificar el trabajo continuo del equipo de trabajo en las diferentes actividades que este requiere.

OBJETIVOS ESPECÍFICOS.

Concretar las diferentes herramientas que han de utilizarse en la creación del proyecto, con el fin de establecer los métodos para la realización del proyecto.

Conocer el planteamiento del problema a resolver y así mismo plantear una solución, tomando en cuenta el uso de las diferentes herramientas informáticas que nos permiten hacer con mayor facilidad las cosas, consecuentemente ahorrar más tiempo.

PLANTEAMIENTO DEL PROBLEMA.

A diario las personas generan información, la cual puede ser clasificada y almacenada de muchas maneras, desde un sistema informático hasta en papel, pero surgen

Las preguntas, ¿Que información debe ser almacenada? ¿Como almacenarla? ¿Como acceder a ella con mayor facilidad? ¿Donde utilizar esa información?

De estas y muchas otras preguntas surge la necesidad de crear un programa que almacene información, y que esta información pueda ser consultada, clasificada y ordenada en este programa, para facilitar el acceso a ella.

Por eso tomando en cuenta este problema decidimos plantear: ¿Como almacenar los diferentes productos de una farmacia en un sistema de inventario, y que en este sistema pueda consultar los datos que en el ingrese?

Marco referencial

Python es un lenguaje de programación que gracias a sus características ha llegado a ser un lenguaje muy conocido en la actualidad. A continuación se listan las principales características que este lenguaje posee:

Simple:

Python es un lenguaje muy simple, por lo que es muy fácil iniciarse en este lenguaje. El pseudo-código natural de Python es una de sus grandes fortalezas.

Propósito General:

Usando el lenguaje Python se puede crear todo tipo de programas; programas de propósito general y también se pueden desarrollar páginas Web.

Open Source:

Debito a la naturaleza de Python de ser Open Suorce; ha sido modificado para que pueda funcionar en diversas plataformas (Linux, Windows, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE y PocketPC).

Al ser Open Source es gratuito.

Lenguaje Orientado a Objetos:

Al ser un Lenguaje Orientado a Objetos es construido sobre objetos que combinan datos
y funcionalidades.

Lenguaje de Alto

Nivel:

Al programar en Python no nos debemos preocupar por detalles de bajo nivel, (como manejar la memoria empleada por el programa).

Incrustable:

Se puede insertar lenguaje Python dentro un programa C/C++ y de esta manera ofrecer las facilidades del scripting.

Extensas Librerías:

Python contiene una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero.

Las librerías pueden ayudar a hacer varias cosas como expresiones regulares, generación de documentos, evaluación de unidades, pruebas, procesos, bases de datos, navegadores web, CGI, ftp, correo electrónico, XML, XML-RPC, HTML, archivos WAV, criptografía, GUI, y también otras funciones dependientes del Sistema.

Sintaxis clara:

Python tiene una sintaxis muy visual, gracias a que maneja una sintaxis identada (con márgenes), que es de carácter obligatorio. Para separar los bloques de código en Python se debe tabular hacia dentro. Esto ayuda a que todos los programadores adopten las mismas notaciones y que los programas hechos en Python tengan un aspecto muy similar.

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP).

Marco conceptual

El sistema de inventario es una solución basada en la web, diseñado para pequeñas y medianas empresas que buscan un sistema completo de gestión de inventario sin reemplazar el software real para la contabilidad. El Sistema de Inventario es lo
Suficientemente flexible para satisfacer las necesidades de las empresas que requieren un control de inventario y facturación más ágil. El sistema también automatiza los procesos de compras y ventas, generando así un reporte con gráficos comparativos de cada mes.

El Sistema de Inventario ofrece además, un conjunto de características que lo hacen una de las mejores opciones para administrar de forma eficiente el inventario de su empresa o negocio.

También puede ser definido como conjunto de normas, métodos y procedimientos aplicado de manera sistemática para planificar y controlar los materiales y productos que se emplean en una organización. Este sistema puede ser manual o automatizado. Para el control de los costos, elemento clave de la administración de cualquier organización, existen sistemas que permiten estimar los costos de las mercancías que son adquiridas y luego procesadas o vendidas.

Tipos de modelo de sistema de inventario.

Sistema de Inventario Perpetuo: el negocio mantiene un registro continuo para cada artículo del inventario. Los registros muestran por lo tanto el inventario disponible todo el tiempo. Los registros perpetuos son útiles para preparar los estados financieros mensuales, trimestral o provisionalmente.

Sistema de Inventario Periódico: En el sistema de inventario periódico el negocio no mantiene un registro continuo del inventario disponible, más bien, al fin del periodo, el negocio hace un conteo físico del inventario disponible y aplica los costos unitarios para determinar el costo del inventario final. Ésta es la cifra de inventario que aparece en el
Balance General. Se utiliza también para calcular el costo de las mercancías vendidas. El
Sistema periódico es conocido también como sistema físico, porque se **apoya** en el conteo físico real del inventario. El sistema periódico es

generalmente utilizado para contabilizar los artículos del inventario que tienen un costo unitario bajo.

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar
Los datos a las necesidades de cada uno.

Historia – Antecedentes

El inventario comenzó en la antigüedad, cuando los pueblos, debido a la escasez, deciden almacenar grandes cantidades de alimentos, para hacer frente a ellas; así se idea un mecanismo de control para su reparto.

En los años 70s se requería mantener un inventario suficiente con el fin de no parar ningún proceso y no agotar un producto terminado. Se tenían altos márgenes de rentabilidad y se justifica tener altos niveles de inventario, con dos elementos a favor: altos índices a la inflación y restricción a las importaciones.

En los años 80s se pensaba en tener un inventario suficiente e imprimirle cierta dinámica. Se empezó a hablar de flujo de inventarios. Ya se calculaba el índice de rotación de inventarios (Ventas (consumo) / valor inventario promedio) y posteriormente la velocidad del inventario. Incluso, se llego a estudiar la forma de tener cero inventarios.

En los años 90 se compra con más facilidad y los índices de inflación son bajos, algunas empresas se aceleraron e incrementaron sus niveles de inventario. Se acrecentó el problema que tienen muchas organizaciones: exceso de inventario.

En la actualidad se emplean diferentes procesos y maquinas que facilitan la realización
De los inventarios. Se cree que los inventarios son un método o procedimiento que ayuda
A tener un control de las mercancías y a llevar un orden en la empresa.

Los inventarios aparecieron con la necesidad que tuvo el hombre de almacenar y llevar un orden o control en sus bienes para poder sobrevivir en la época de escasez.

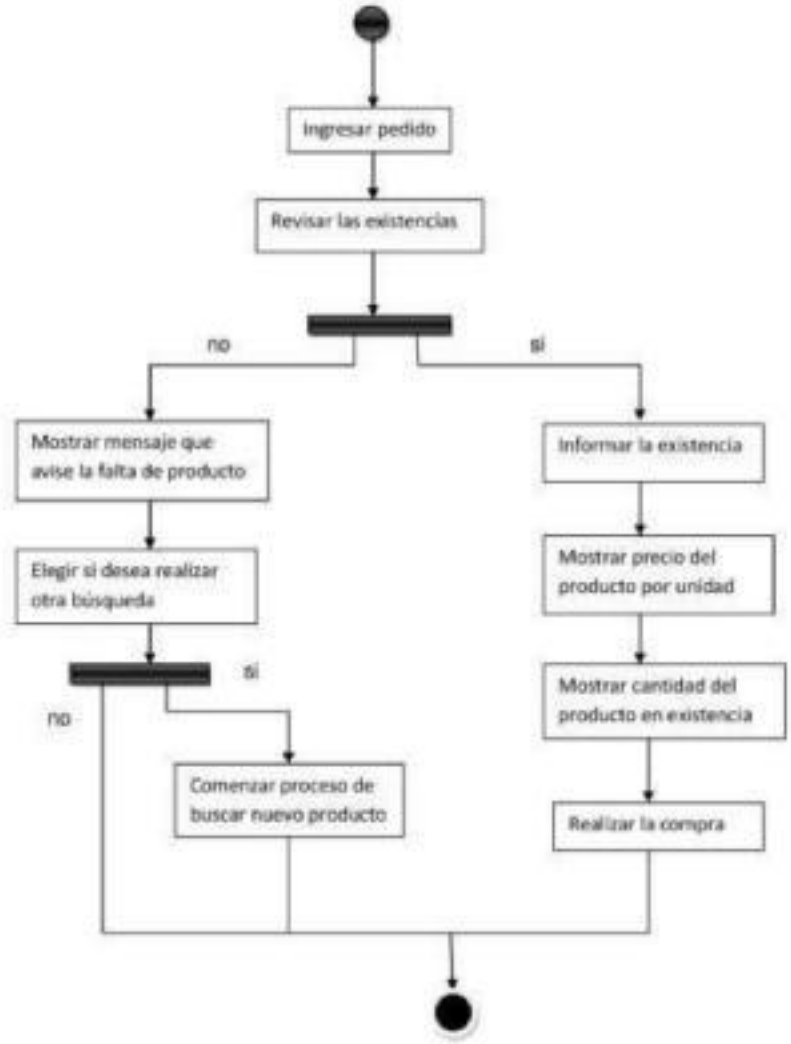
Justificación

Se ha decidido llevar a cabo este proyecto, para poner en práctica muchas de las técnicas que se están utilizando en el transcurso de nuestro aprendizaje; gracias a este proyecto se verá reflejado mucho de lo aprendido, ya que abarca gran parte de los temas vistos hasta el momento.

La importancia que tiene tipo de sistema también ha sido considerada por el gran énfasis

Que se tiene a futuro, se aprenderán nuevas técnicas para optimizar el rendimiento de nuestro trabajo hasta el día de hoy.

Diagrama De Actividades



Código en python de cómo puede ser manipulada la información de registros:

```
#Uso de un diccionario para manejo de registros en memoria
dic = {}
ind_cant = 0
ind_nom = 1
salida = False
while not salida:
        print "BIENVENIDO, QUE DESEAS HACER."
        print ("\n1) Ingresar articulo." )
        print ("2) Consultar articulo.")
        print ("3) Eliminar articulo.")
        print ("4) Salir")
        opc = int(input("\nElija una opcion por favor: "))
        if opc == 1:
                cla = int(input("Ingrese clave: "))
                cant = int(input("Ingrese cantidad: "))
                nom = input("Ingrese nombre: ")
                dic[cla] = [cant, nom]
        elif opc == 2:
                cla = int(input("Ingrese clave: "))
                if cla in dic:
                        print ("Cantidad: ", dic[cla][ind_cant])
                        print ("Nombre: ", dic[cla][ind_nom]
                else:
                        print "No existe."

        elif opc == 3:
                cla = int(input("Ingrese clave: "))
                if cla in dic:
                        del dic[cla]
                else:
                        print("No existe.")

        elif opc == 4:
                salida = True
```

CODIGO DE PYTHON PARA REGISTRAR INVENTARIO

~/Escritorio/Proyecto_Final/producto_view.py - Sublime Text (UNREGISTERED)

producto_view.py

```
# -*- coding: utf-8 *-*


class ProductoView:

    def __init__(self):
        self.tab1 = "    "
        self.tab2 = "    " * 2
        self.tab3 = "    " * 5
        self.txt_opt = "%sElija una opción: " % self.tab2
        self.txt_Nombre_producto = "%sProducto: " % self.tab3
        self.txt_Fecha_caducidad = "%sFecha de caducidad: " % self.tab3
    self.txt_Precio_unitario = "%sPrecio unitario: " % self.tab3
    self.txt_Cantidad = "%sCantidad: " % self.tab3
    self.txt_id = "%sID del producto: " % self.tab3
        pass

    def mostrar_menu(self):
        """Vista del menú de opciones"""

        menu = """
        Menú del Gestor de Productos
            (1) Crear un producto
            (2) Ver listado de productos
            (3) Editar un producto
            (4) Eliminar un producto

            (0) Salir

        """
        print menu

        opcion = raw_input(self.txt_opt)
        return opcion

    def crear_producto(self):
        """Vista del formulario para crear nuevo producto"""

        print """
```

Line 1, Column 1

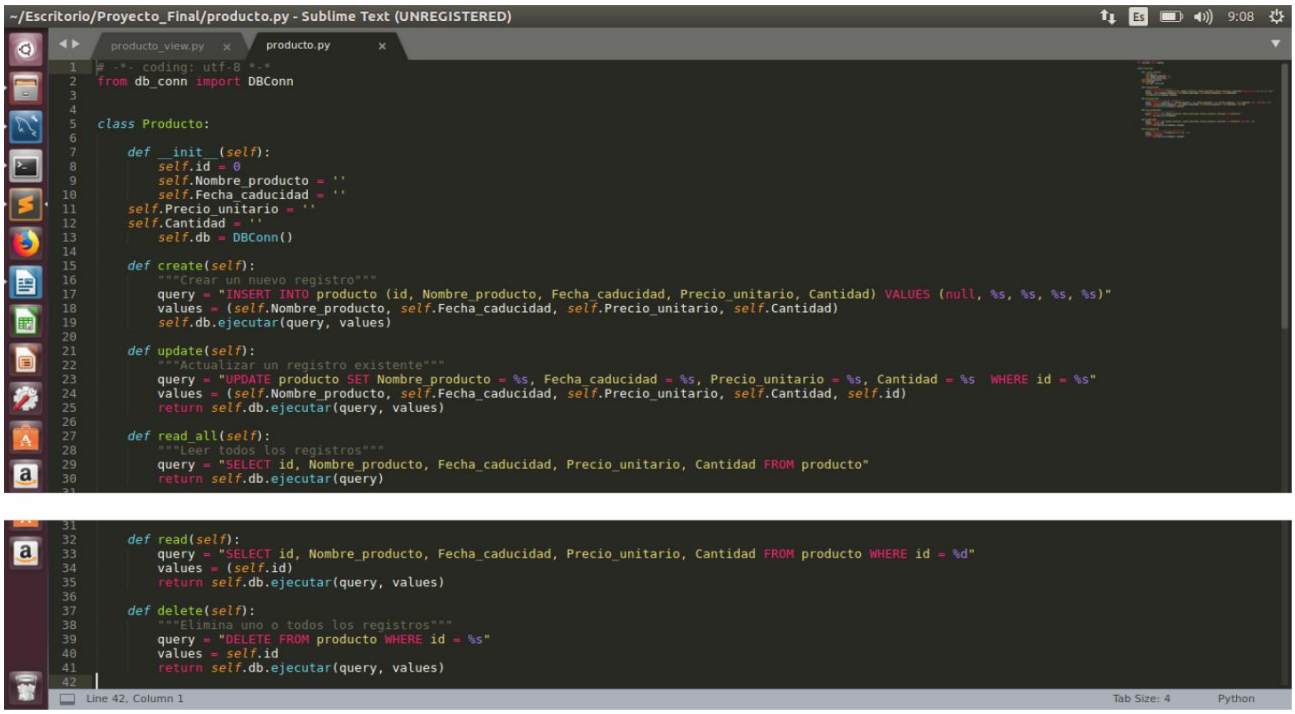
```
        print """
        Producto creado con éxito!
        """

    def listar_producto(self, listado):
        """Vista para el listado de productos"""

        print """
            LISTADO DE PRODUCTOS:
        """
        for row in listado:
            id = row[0]
            Nombre_producto = row[1]
            Fecha_caducidad = row[2]
        Precio_unitario = row[3]
        Cantidad = row[4]
            print "%s[%d] %s (%s) %s (%s)" % (self.tab3, id, Nombre_producto, Fecha_caducidad, Precio_unitario, Cantidad)

    def editar_producto(self, listado):
        """Vista del formulario para editar un producto"""

        self.listar_producto(listado)
        print "\n\n"
        id = raw_input(self.txt_id)
        print "\n"
        Nombre_producto = raw_input(self.txt_Nombre_producto)
        Fecha_caducidad = raw_input(self.txt_Fecha_caducidad)
```

Line 77, Column 1

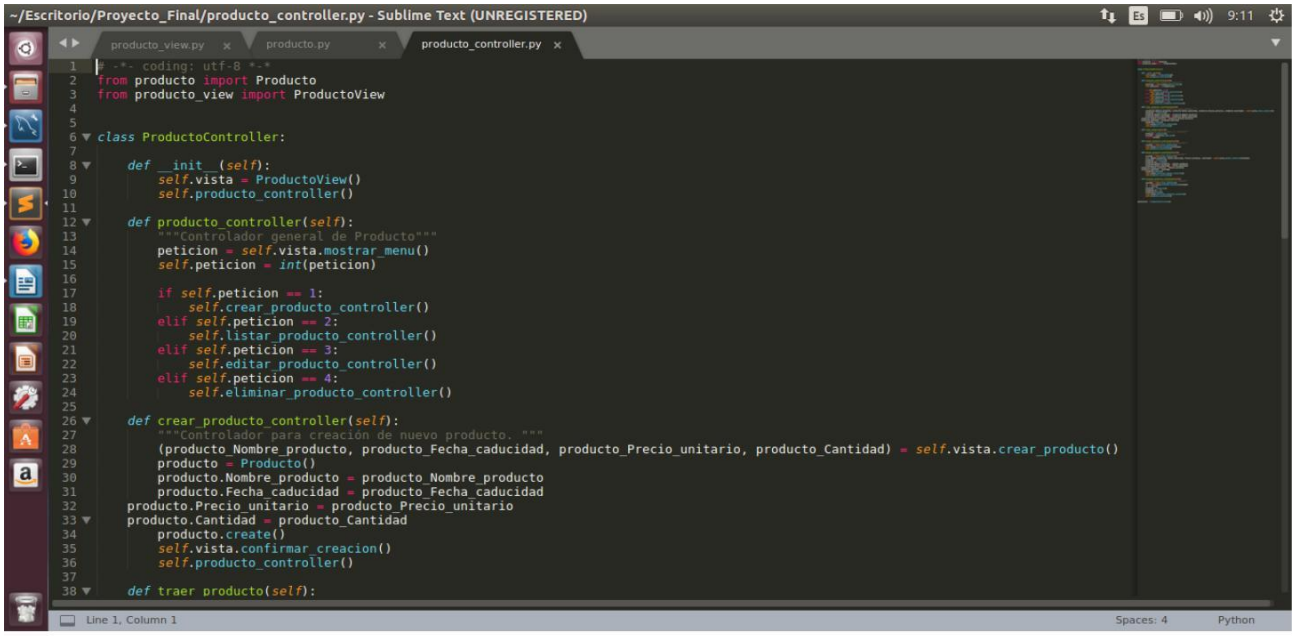
```
    def editar_producto(self, listado):
        """Vista del formulario para editar un producto"""

        self.listar_producto(listado)
        print "\n\n"
        id = raw_input(self.txt_id)
        print "\n"
        Nombre_producto = raw_input(self.txt_Nombre_producto)
        Fecha_caducidad = raw_input(self.txt_Fecha_caducidad)
    Precio_unitario = input(self.txt_Precio_unitario)
    Cantidad = input(self.txt_Cantidad)
        return (id, Nombre_producto, Fecha_caducidad, Precio_unitario, Cantidad)


    def confirmar_editar_producto(self):
        """Vista de confirmación de edición"""

        print """
        Producto editado correctamente.
        """

    def eliminar_producto(self, listado):
        """Vista de formulario para eliminar producto"""

        self.listar_producto(listado)
        print "\n\n"
        id = raw_input(self.txt_id)
        print "\n"
        return id

    def confirmar_eliminar_producto(self):
        """Vista de cofirmación eliminar producto"""
        print """
        Producto eliminado correctamente.
        """
```

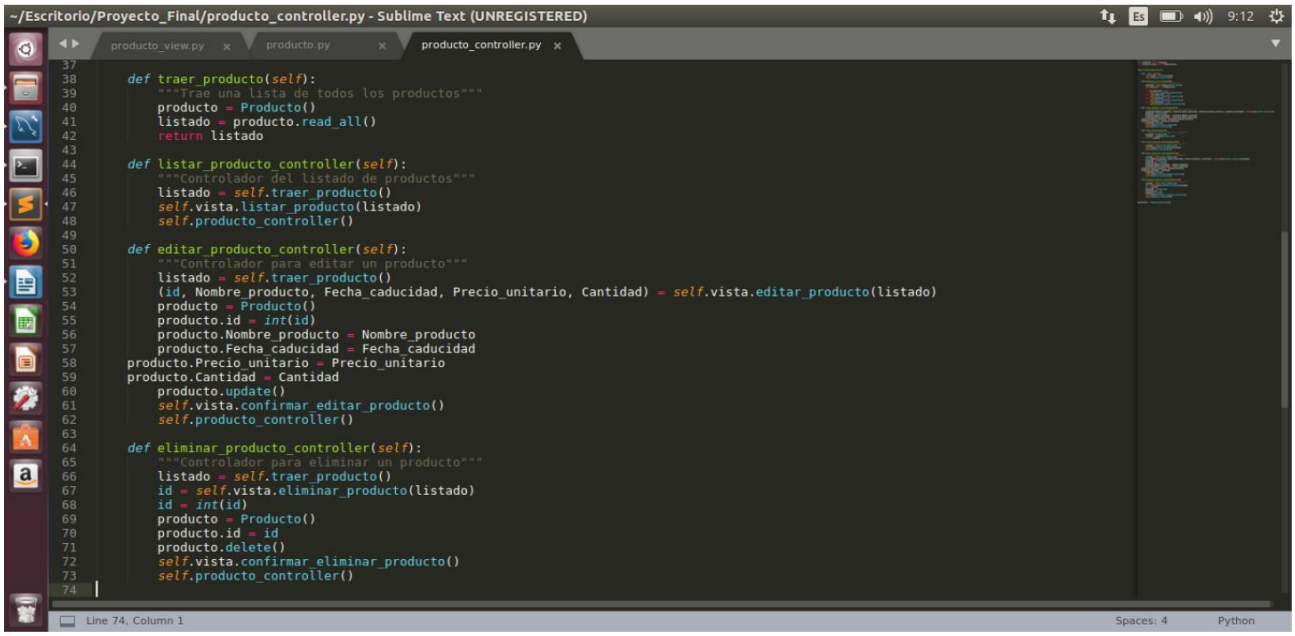
Line 104, Column 1

IMPORTANDO A LA BASE DE DATOS

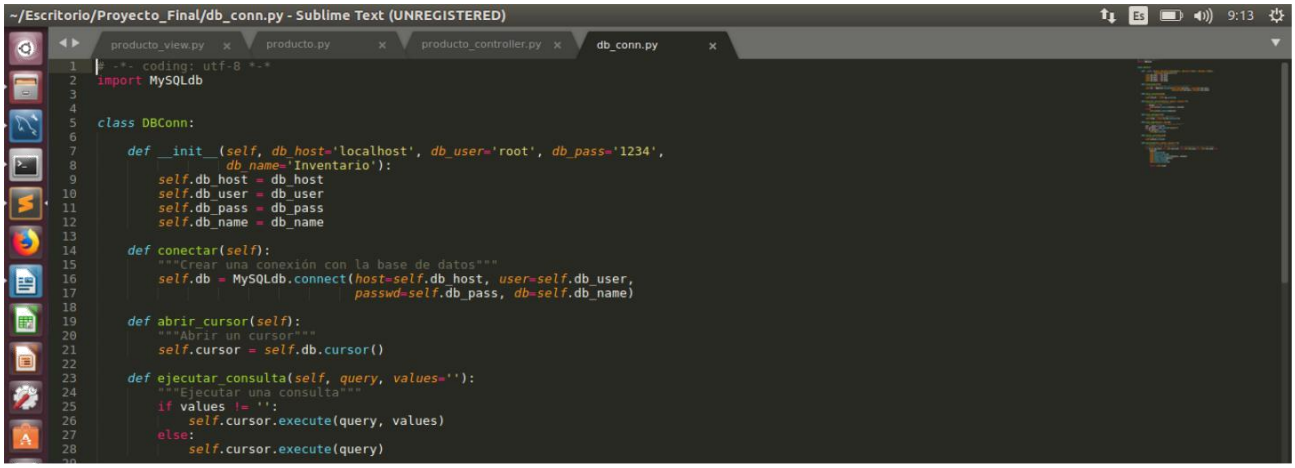


CONTROLADOR

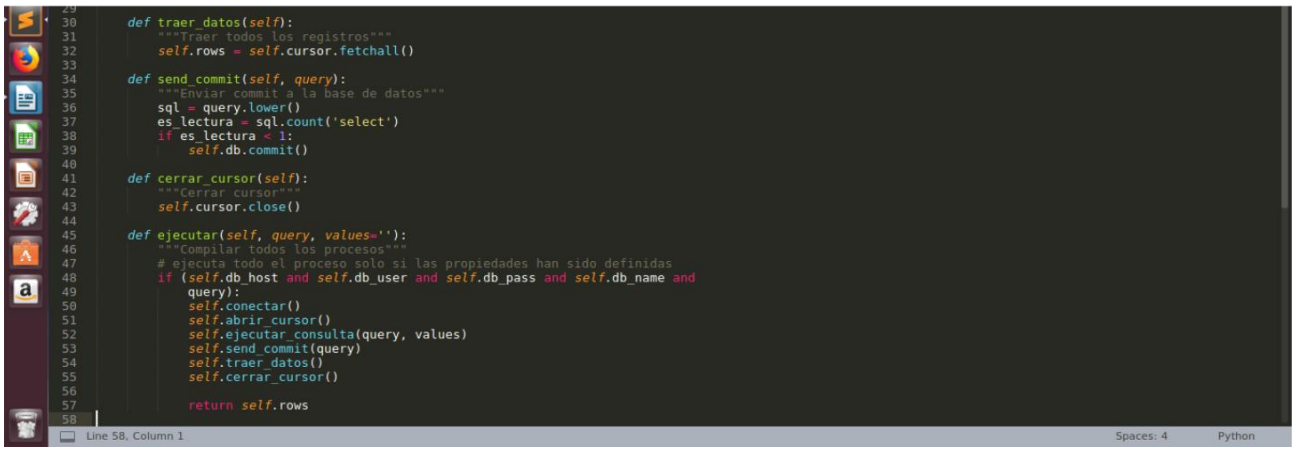




CONEXIÓN A LA BASE DE DATOS

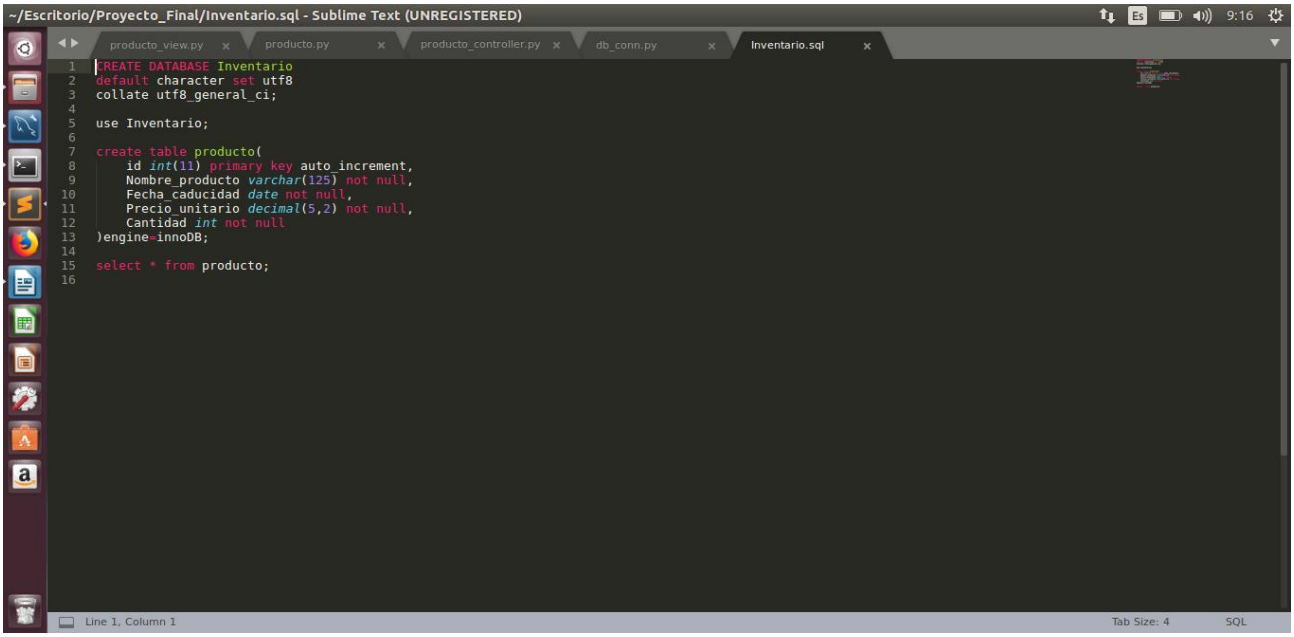



```python
# -*- coding: utf-8 *-*
import MySQLdb


class DBConn:

    def __init__(self, db_host='localhost', db_user='root', db_pass='1234',
                 db_name='Inventario'):
        self.db_host = db_host
        self.db_user = db_user
        self.db_pass = db_pass
        self.db_name = db_name

    def conectar(self):
        """Crear una conexión con la base de datos"""
        self.db = MySQLdb.connect(host=self.db_host, user=self.db_user,
                                  passwd=self.db_pass, db=self.db_name)

    def abrir_cursor(self):
        """Abrir un cursor"""
        self.cursor = self.db.cursor()

    def ejecutar_consulta(self, query, values=''):
        """Ejecutar una consulta"""
        if values != '':
            self.cursor.execute(query, values)
        else:
            self.cursor.execute(query)
```



```python
    def traer_datos(self):
        """Traer todos los registros"""
        self.rows = self.cursor.fetchall()

    def send_commit(self, query):
        """Enviar commit a la base de datos"""
        sql = query.lower()
        es_lectura = sql.count('select')
        if es_lectura < 1:
            self.db.commit()

    def cerrar_cursor(self):
        """Cerrar cursor"""
        self.cursor.close()

    def ejecutar(self, query, values=''):
        """Compilar todos los procesos"""
        # ejecuta todo el proceso solo si las propiedades han sido definidas
        if (self.db_host and self.db_user and self.db_pass and self.db_name and
            query):
            self.conectar()
            self.abrir_cursor()
            self.ejecutar_consulta(query, values)
            self.send_commit(query)
            self.traer_datos()
            self.cerrar_cursor()

            return self.rows

```

CODIGO EN MYSQL



CONCLUSIONES

Finalmente, como se puede ver e muchos negocios es muy necesario un sistema de inventario para poder agilizar las transacciones y búsqueda en los productos. Se ha tomado a bien que se realice un sistema de inventario para poder identificar la existencia y cantidad que se ha registrado y así evitarse las confusiones; así también se tendrá la posibilidad de poder introducir de manera fácil un producto. Con la ayuda de este proyecto podremos darnos una idea de la magnitud que puede tener un algoritmo que tiene la posibilidad de almacenar información.

DIAGRAMA DE ACTIVIDADES

Etapa	Descripción	Fecha
1° Avance	Se definió que se realizaría, que lenguaje de programación Sería utilizado y la magnitud del proyecto.	5 de Marzo de 2018
2° Avance	Se creó la primera etapa del código que sería utilizado y la Base de datos.	18 de Abril de 2018
3° Avance	Se finalizaron los códigos en python y base datos; se estableció la conexión, se empezaron las pruebas para el buen Funcionamiento del programa.	20 de Mayo de 2018

Bibliografía

http://www.cuatrorios.org/index.php/option=com_content&view=article&id=161:prin cipales-caracteristicas-del-lenguaje-python&catid=39:blogsfeeds

https://obedalvarado.pw/sistema-de-control-de-inventario/

https://realizacioninvetariosa4-1.wikispaces.com/Tipos+de+sistemas+y+modelos+de+inventario