

PROYECTO CRUD TIENDA

Sebastian Moreno Castañeda

Cristian Jesus Gonzalez Coley

Grupo: 30126

Docente: RAFAEL ENRIQUE BENEDETTI NAVARRO

CORPORACIÓN UNIFICADA NACIONAL DE EDUCACIÓN SUPERIOR

INGENIERÍA DE SISTEMAS

INGENIERÍA WEB

Bogotá D.C.

2020

INTRODUCCIÓN

Teniendo en cuenta las situaciones presentes de salud, se hace necesario crear una solución virtual, en la que se ofrecerá un catálogo virtual sobre prendas de vestir, así dar una reinversión a este negocio. Este desarrollo ofrece nuevas posibilidades al comerciante, para ofrecer sus productos, control de los clientes, ventas y ganancias. Será un software desarrollado a la medida del usuario.

OBJETIVO GENERAL

Se desarrollará una aplicación a la medida del clientes, que promocióne las prendas de vestir que maneja el cliente, también podrá revisar el catálogo en línea y realizar compras al por mayor o al detal, lo que ofrecerá al usuario información en tiempo real.

OBJETIVOS ESPECÍFICOS

- ❖ Desarrollar la aplicación, como modelo de negocio web, teniendo conexión a un motor de base de datos.
- ❖ Evaluar la utilidad del uso del software teniendo en cuenta variables como inversión económica, eficiencia del sistema.

REQUERIMIENTO FUNCIONALES

Requerimientos de Proceso

REQUERIMIENTO 1	
CATEGORÍA	Requerimiento de proceso
DESCRIPCIÓN	Consulta de productos existentes
ENTRADA	Se podrá ver los productos disponibles
SALIDA	La aplicación mostrar los productos actuales
RESUMEN	Productos del catálogo

REQUERIMIENTO 2	
CATEGORÍA	Requerimiento de proceso
DESCRIPCIÓN	Se podrán agregar productos nuevos
ENTRADA	Se podrán agregar otros productos
SALIDA	La aplicación mostrar las categorías de los productos
RESUMEN	Productos del catálogo

REQUERIMIENTO 3	
CATEGORÍA	Requerimiento de proceso
DESCRIPCIÓN	Se podrá eliminar los productos disponibles
ENTRADA	Se podrá eliminar productos disponibles
SALIDA	La aplicación mostrar las categorías de los productos
RESUMEN	Productos del catálogo

Requerimientos de Interfaz Gráfica

REQUERIMIENTO 4	
CATEGORÍA	Requerimiento de interfaz gráfica
DESCRIPCIÓN	La aplicación tendrá responsive desing
ENTRADA	La aplicación cumplirá con el responsive design
SALIDA	La aplicación cargará en todos los dispositivos

RESUMEN	Comportamiento interfaz
----------------	-------------------------

REQUERIMIENTO 5	
CATEGORÍA	Requerimiento de interfaz gráfica
DESCRIPCIÓN	La aplicación validará todos los campos
ENTRADA	La aplicación validará y distinguirá entre campos vacíos, números y letras
SALIDA	Todos los campos para llenar del usuario, estarán diligenciados
RESUMEN	Validación de campos

Requerimientos de seguridad

REQUERIMIENTO 6	
CATEGORÍA	Requerimiento de Seguridad
DESCRIPCIÓN	Base de datos
ENTRADA	Se realizará copia de los datos de la bd
SALIDA	Se mantendrá copia en otro servidor de la bd, semanalmente
RESUMEN	Copia de seguridad

Requerimientos de Interfaz externa

REQUERIMIENTO 7	
CATEGORÍA	Requerimiento de Interfaz externa
DESCRIPCIÓN	El catálogo no tendrá conexión con otras interfaces externas
ENTRADA	El catálogo no contara con conexión a otra interfaces
SALIDA	
RESUMEN	Conexión a otras interfaces

REQUERIMIENTOS NO FUNCIONALES

Requerimientos de Eficiencia

REQUERIMIENTO 8	
CATEGORÍA	Requerimiento de eficiencia
DESCRIPCIÓN	Respuesta de la base de datos
ENTRADA	Los datos modificados en la base de datos deberán ser actualizados en un tiempo de 10 segundos máximo
SALIDA	Rendimiento de la base de datos
RESUMEN	Rendimiento de la base de datos

REQUERIMIENTO 9	
CATEGORÍA	Requerimiento de eficiencia
DESCRIPCIÓN	Ingreso al sistema
ENTRADA	El catálogo deberá responder al usuario en el momento de la solicitud y/o asignación en 5 segundos
SALIDA	
RESUMEN	Tiempo de espera

Requerimientos de Seguridad lógica y de la información

REQUERIMIENTO 10	
CATEGORÍA	Requerimiento de seguridad lógica y de la información
DESCRIPCIÓN	Tratamiento de datos
ENTRADA	La aplicación, tendrá el tratamiento de los datos, según la Ley de habeas Data de nuestro país
SALIDA	
RESUMEN	Uso de la información

REQUERIMIENTOS

Sistema operativo LINUX MINT version 20 LTS
FRAMEWORK FLASK ULTIMA VERSION
Editor de texto VISUAL STUDIO CODE, ATOM
Versión de PYTHON 3.6,3
Entorno virtual VIRTUALENV
Base de datos en HEROKU con POSTGRE
Se usará BOOTSTRAP 5 como framework para estilos
CSS3 puro para estilizar algunas hojas

ESTRUCTURA DEL PROYECTO

Para el proyecto se usará un entorno virtual que es donde se ejecutará el proyecto, en este documento se intentará detallar cómo se instalará el entorno virtual para correr el entorno virtual de python y la instalación del framework elegido para el proyecto llamado flask

COMO SE INSTALA FLASK

Para el proyecto seleccione el framework flask, porque se acomoda a las necesidades del proyecto, en especial por ser la primera vez que realizo un proyecto con el framework flask

A continuación mostrare los pasos requeridos para instalar flask en mi equipo

Revisamos la versión de Python instalada previamente en el equipo

```
○ → python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Luego de haber verificar la versión instalada de Python, procedemos a instalar la librería pip

```
○ → sudo apt-get install python3-pip
[sudo] password for desarrollo:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.1).
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
```

Se verifica la version de pip instalada

```
○ → pip3 --version
pip 20.1 from /home/desarrollo/.local/lib/python3.6/site-packages/pip (python 3.6)
```

```
○ → pip3 freeze
appdirs==1.4.3
apt-xapi-an-index==0.47
```


Después de instalar pip vamos a instalar la librería virtualenv, que será nuestro entorno virtual donde residirá nuestra aplicación.

```
○ → pip3 install virtualenv

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: virtualenv in ./local/lib/python3.6/site-packages (20.0.20)
Requirement already satisfied: distlib<1,>=0.3.0 in ./local/lib/python3.6/site-packages (from virtualenv) (0.3.0)
Requirement already satisfied: appdirs<2,>=1.4.3 in ./local/lib/python3.6/site-packages (from virtualenv) (1.4.3)
Requirement already satisfied: importlib-resources<2,>=1.0; python_version < "3.7" in ./local/lib/python3.6/site-packages (from virtualenv) (1.5.0)
Requirement already satisfied: importlib-metadata<2,>=0.12; python_version < "3.8" in ./local/lib/python3.6/site-packages (from virtualenv) (1.6.0)
Requirement already satisfied: six<2,>=1.9.0 in ./local/lib/python3.6/site-packages (from virtualenv) (1.14.0)
Requirement already satisfied: filelock<4,>=3.0.0 in ./local/lib/python3.6/site-packages (from virtualenv) (3.0.12)
Requirement already satisfied: zipp>=0.4; python_version < "3.8" in ./local/lib/python3.6/site-packages (from importlib-resources<2,>=1.0; python_version < "3.7"->virtualenv) (3.1.0)
```

A continuación verificamos el paquete virtualenv instalado .

```
virtualenv==20.0.20
```

El siguiente paso, es crear la carpeta del proyecto, la cual abriremos seguidamente

```
○ → cd prueba/
```

Dentro de la carpeta creada, vamos a abrir el entorno virtual

```
○ → python3 -m virtualenv vmyenv
created virtual environment CPython3.6.9.final.0-64 in 1042ms
  creator CPython3Posix(dest=/home/desarrollo/prueba/vmyenv, clear=False, global=False)
  seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, via=copy, app_data_dir=/home/desarrollo/.local/share/virtualenv/seed-app-data/v1.0.1)
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

Luego de instalado el entorno virtual, verificamos que se haya creado correctamente.

```
2020-05-15 10:42:14 development on /prueba
○ → ls
venv vmyenv
```

Finalmente ejecutaremos el entorno virtual

```
○ → source vmyenv/bin/activate
```

Con el entorno virtual en ejecución, instalamos el framework flask

```
2020-05-13 18:43:02 ● development in ~/prueba
○ → pip3 install Flask
Collecting Flask
  Using cached Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting Werkzeug>=0.15
  Using cached Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Collecting itsdangerous>=0.24
  Using cached itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting click>=5.1
  Using cached click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting Jinja2>=2.10.1
  Using cached Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting MarkupSafe>=0.23
  Using cached MarkupSafe-1.1.1-cp36-cp36m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: Werkzeug, itsdangerous, click, MarkupSafe, Jinja2, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.1.0
```

Verificamos la correcta instalación del entorno

```
○ → pip freeze
click==7.1.2
Flask==1.1.2
itsdangerous==1.1.0
Jinja2==2.11.2
MarkupSafe==1.1.1
Werkzeug==1.0.1
```

Vamos a crear nuestra primera aplicación en el framework, importando el framework instalado en el paso anterior en el archivo index.py. Creamos una estructura básica de muestra.

```
index.py X
index.py
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def hello_world():
6      return 'Hello, World!'
7
8  if __name__ == '__main__':
9      app.run()
10
11
```

Ejecutaremos el proyecto

```
2020-05-13 21:45:13 ⌚ development in ~
○ → cd prueba/

2020-05-13 21:47:26 ⌚ development in ~/prueba
○ → ls
vdanjo venv vmyenv

2020-05-13 21:47:32 ⌚ development in ~/prueba
○ → source vmyenv/bin/activate

2020-05-13 21:47:58 ⌚ development in ~/prueba
○ → cd vmyenv/

2020-05-13 21:48:03 ⌚ development in ~/prueba/vmyenv
○ → ls
bin index.py lib pyvenv.cfg

2020-05-13 21:48:04 ⌚ development in ~/prueba/vmyenv
○ → python index.py
* Serving Flask app "index" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  nt.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [13/May/2020 21:48:28] "GET / HTTP/1.1" 200 -
```

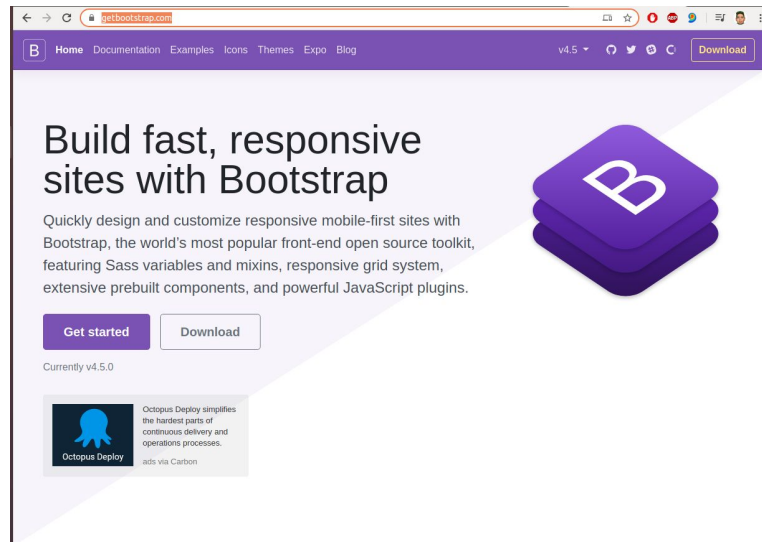
Ahora relacionamos los avances actuales del proyecto, la hoja principal del py, hoja de estilo y diseño web

```
2020-05-16 20:06:53 ⌚ development in ~/proyecto final
○ → python index.py
* Serving Flask app "index" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  nt.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 178-161-373
127.0.0.1 - - [16/May/2020 20:58:57] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [16/May/2020 20:58:57] "GET /favicon.ico HTTP/1.1" 200 -
```

Hojas de Estilo

Para el proyecto se usó la librería de BOOTSTRAP 5, que nos ayudará a realizar una implementación de CSS, más rápida además de implementar layouts más fáciles de controlar que si se hiciera con CSS puro; sin embargo también se usaron estilos propios para personalizar el color.

Para la instalación de BOOTSTRAP, nos dirigimos ala web oficial <https://getbootstrap.com/> y copiamos las rutas de CDN Y JS, en nuestro layout principal



Complete CSS

<https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css>

Click to copy

HTML

```
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-6xWa9qMvh3PBpNqDHY5tsfEaEXpbSU5aJv64tsIFPoukMBEDNGZOwhBwrePKvqjY4w" sha384-6xWa9qMvh3PBpNqDHY5tsfEaEXpbSU5aJv64tsIFPoukMBEDNGZOwhBwrePKvqjY4w">
```

Complete JavaScript

<https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js>

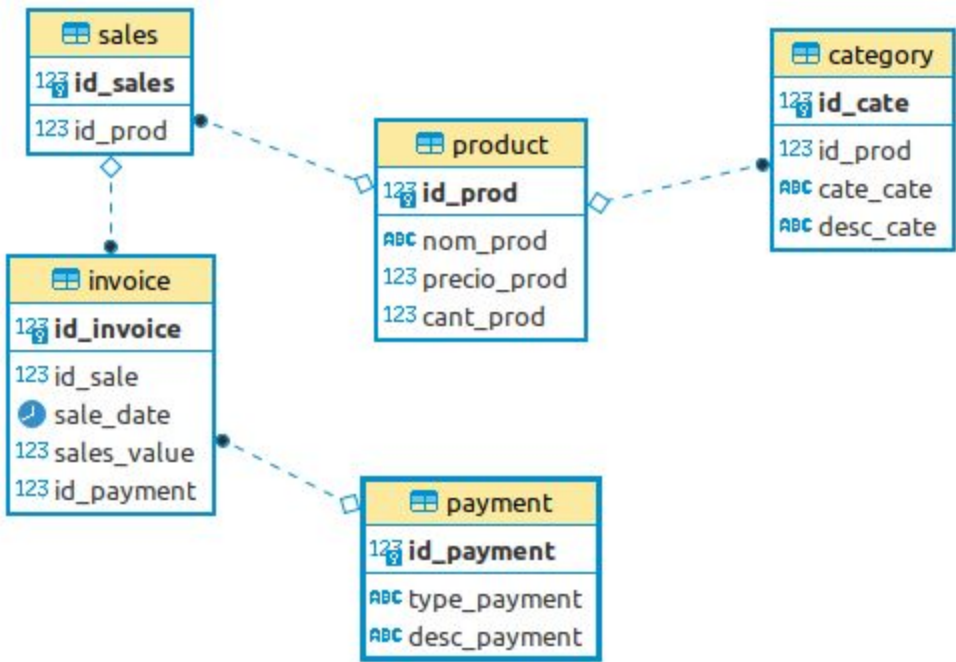
Click to copy

HTML

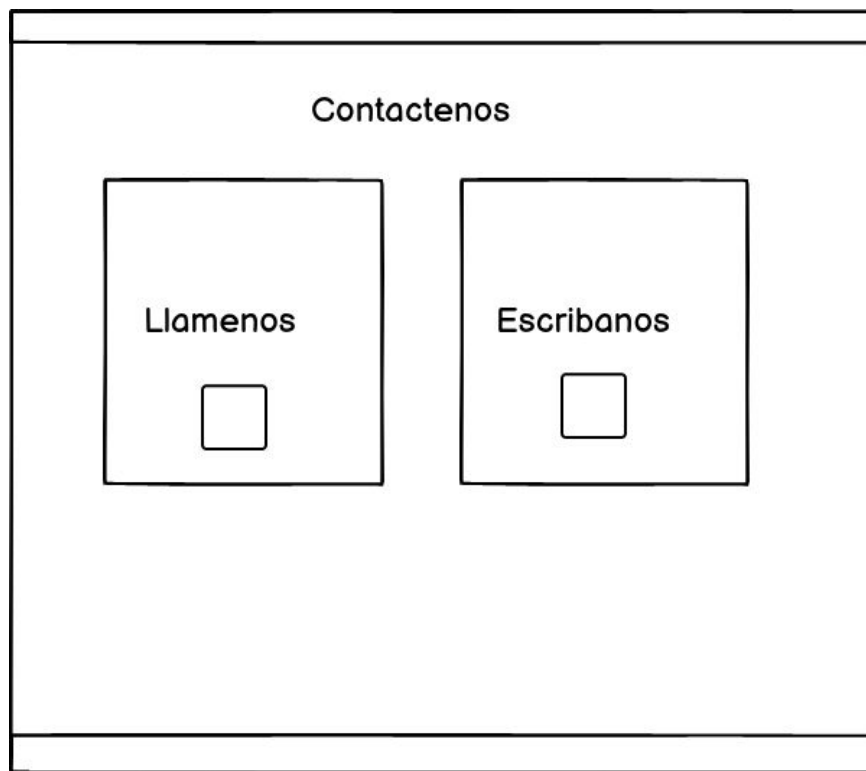
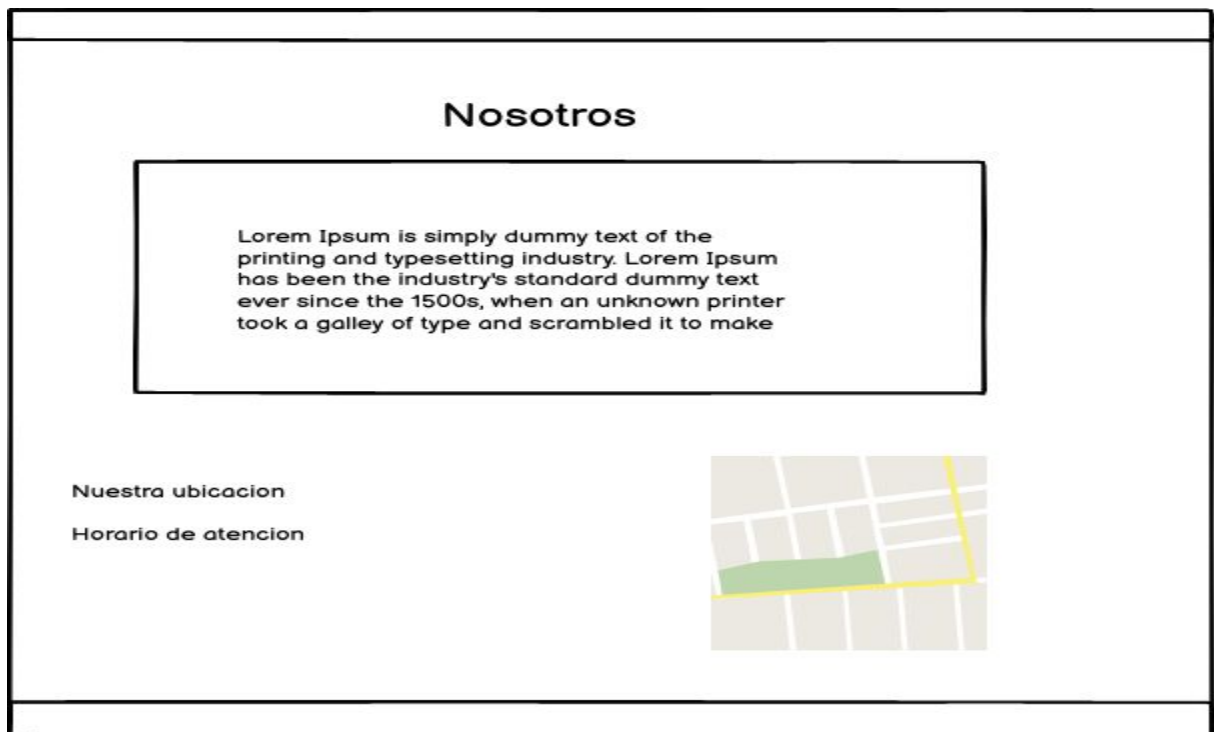
```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z3vEACXEaoiAA0fUEXPOe7E56o9058A4ss3VhUkwu71Xol2IRKwYQ" sha384-OgVRvuATP1z3vEACXEaoiAA0fUEXPOe7E56o9058A4ss3VhUkwu71Xol2IRKwYQ"></script>
```

Click to copy

MODELO RELACIONAL

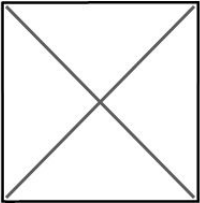


WIREFRAME



Producto

Producto 1



Descripcion del producto

Categorias

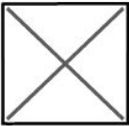


GUARDAR




EDITAR

CANCELAR

Categorias

CATEGORIAS





RESUMEN VENTA



Usuario



Producto

Efectivo

Tarjeta

Transferencia

Medio de pago

Resumen de compra

Comprar

Cancelar

CONCLUSIONES

- Este proyecto nos sirvió para aplicar los conocimientos adquiridos a lo largo de la carrera en programación y bases de datos.
- El enfoque del proyecto será acerca de la seguridad web