

KIBU Programa de Ventas

El proyecto **KIBU Programa de Ventas** se desarrollo utilizando **Metodologías Agiles**, está en una etapa de desarrollo inicial, no es un proyecto terminado.

Contenido

Herramientas Utilizadas	2
Trello	2
GitHub	2
Lucidchart	2
Diagrama de Casos de Uso	2
Diagrama de Clases	2
Python y Tkinter	2
Proceso de Desarrollo	3
1. Análisis de los Requerimientos.....	3
Puntos importantes.....	3
Rasgos a Evaluar	3
Como lograre cumplir con los rasgos a evaluar	3
2. Creación del Repositorio	3
3. Crear Diagramas UML	4
4. Prueba de concepto	5
Clases.....	5
Interfaz de Usuario	5
5. Integración	5
6. Pasos a seguir	5

Herramientas Utilizadas

Para la realización del proyecto se utilizaron las siguientes herramientas:

Trello

Se utilizó esta herramienta para planificar las tareas y documentar el proceso ágil de desarrollo.

<https://trello.com/b/1ZoZmJTb/kibu-programa-de-ventas>

GitHub

Esta herramienta se utilizó para guardar las diferentes versiones de los archivos del proyecto y tener una copia de respaldo.

<https://github.com/Galia666/KIBU-Programa-de-Ventas>

Lucidchart

Esta herramienta online se utilizó para realizar los diagramas UML iniciales, se realizaron dos diagramas.

Diagrama de Casos de Uso

<https://app.lucidchart.com/invitations/accept/a136bdd4-433c-4a27-a179-4ed54ef0c8ee>

Diagrama de Clases

<https://app.lucidchart.com/invitations/accept/28d1c87e-c9e9-4242-b721-fe2eed679c07>

Python y Tkinter

Para el desarrollo del proyecto se utilizó Python 3.8.3 y el paquete Tkinter para la interfaz de usuario.

Proceso de Desarrollo

1. Análisis de los Requerimientos

Se analizo el documento provisto para el concurso, llegando a las siguientes conclusiones:

Puntos importantes

- Creación de un programa computacional tan completo como sea posible.
- El lenguaje de programación debe ser Python, se pueden utilizar módulos si se requieren
- **Fecha de entrega** del concurso Miércoles 10 de Junio a las 8:00 pm.

Rasgos a Evaluar

1. Complejidad del programa
2. Estructura del código
3. Creatividad en el código
4. Intuitividad del Programa
5. Documentación del código

Como lograre cumplir con los rasgos a evaluar

1. El programa debe ser lo menos complejo posible para que sea facil de utilizar y sean beneficiados tanto el comprador como el vendedor.
2. El código debe estar estructurado de una manera:
 - **organizada**, para un mejor entendimiento del código.
 - **sencillo**, para que otro programador pueda entender lo que se hizo.
 - **documentado**, para que el lector sepa cómo se hizo el código.
 - **de acceso público**, para que otras personas puedan utilizarlo o mejorarlo.
3. Encontrar soluciones creativas a los problemas que se presenten.
4. El programa debe ser fácil de usar, sencillo de comprender, debe poder ser utilizado por personas que no tengan conocimientos previos en computación.
5. La documentación del código se realizara en GitHub mientras se desarrolla el programa

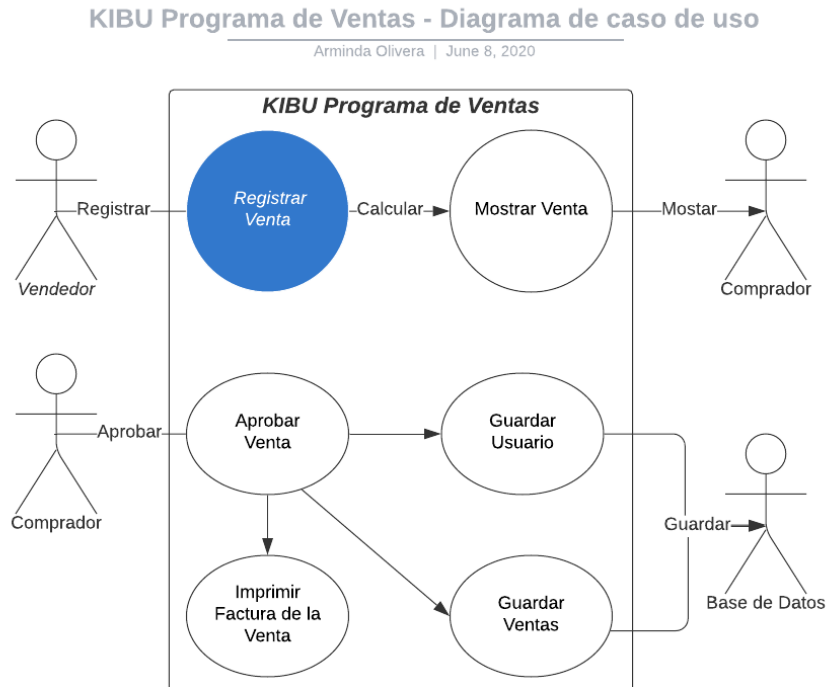
2. Creación del Repositorio

Se creó un nuevo repositorio para el proyecto y se eligió la licencia **Apache License 2.0**

Initial commit

3. Crear Diagramas UML

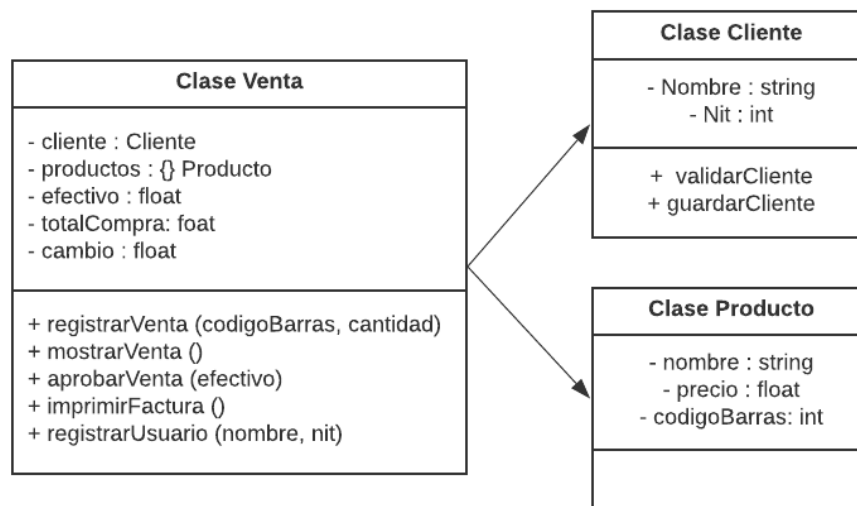
Se procedió a crear un diagrama de casos de uso para comprender el funcionamiento del programa. - [Diagramas UML](#)



Una vez definido el comportamiento de los Usuarios se procedió a crear un diagrama de clases.

KIBU Programa de Ventas - Diagrama de clases

Arminda Olivera | June 8, 2020



4. Prueba de concepto

Clases

Se procedió a escribir las clases y se las ejecuto con valores predeterminados para comprobar el funcionamiento. - [Codigo para realizar ventas](#)

Interfaz de Usuario

Se creó una Interfaz de Usuario (UI) con valores predeterminados - [Prueba de concepto UI](#)

5. Integración

Se procedió a escribir las clases en limpio, en archivos propios y sin valores predefinidos.

Se crearon las clases para la Interfaz de Usuario y se creó una clase principal Main.py como punto de entrada para ejecutar el programa.

[Integrar UI con las Clases](#)

6. Pasos a seguir

Todavía se tiene algunos valores por defecto en el código que deben ser reemplazados para su funcionamiento.

Se necesita crear Test Unitarios que validen el funcionamiento correcto.

Los botones de productos deben mostrar imágenes en vez de texto.

Toda la información debe guardarse en una base de datos.