

# Sistema de registro y gestión para el trabajo informal - (UN)Registro

**Santiago Andrés Gómez Peña**

**Equipo de trabajo 9**

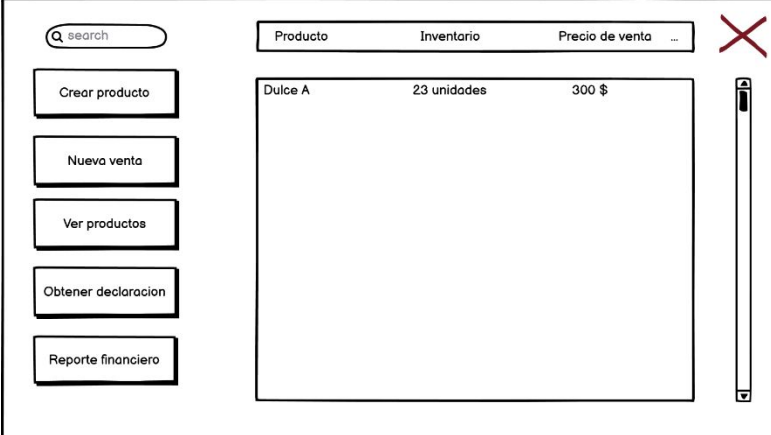


UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA  
SEDE BOGOTÁ



## Problema que se resolvió

Debido a las altas tasas de informalidad presentes en Colombia, se decide diseñar un programa para ayudar a los trabajadores informales con su organización de inventario y seguimiento de operaciones de compra y venta de productos.



The screenshot shows a web application interface for inventory management. On the left side, there is a vertical menu with five buttons: "Crear producto", "Nueva venta", "Ver productos", "Obtener declaracion", and "Reporte financiero". At the top of the main area, there is a search bar labeled "search" and a table header with three columns: "Producto", "Inventario", and "Precio de venta". The table contains one row of data: "Dulce A", "23 unidades", and "300 \$". A red 'X' icon is visible in the top right corner of the application window.

Producto	Inventario	Precio de venta
Dulce A	23 unidades	300 \$

## Requerimientos funcionales

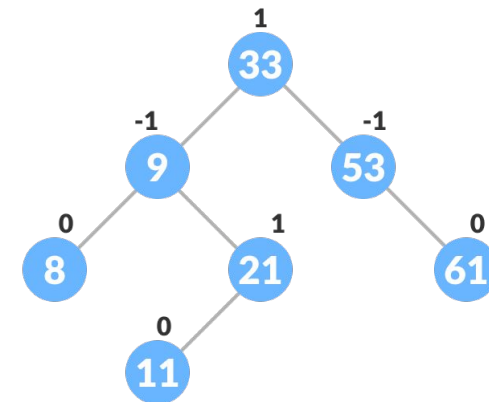
- Crear usuario
- Acceder
- Agregar productos
- Eliminar productos
- Comprar inventario
- Vender productos
- Detalles de un producto
- Lista de productos
- Lista de transacciones



The screenshot shows a Tkinter window titled "tk" with standard window controls (minimize, maximize, close). The window contains a menu bar with the following items: "Agregar", "Eliminar", "Comprar", "Vender", "Buscar", "Productos en registro", "Transacciones realizadas", and "Cerrar sesion". Below the menu bar, there are four input fields labeled "Nombre", "Numero", "Costo", and "Ingreso".

## Implementación de nuevas estructuras de datos en la solución del problema planteado

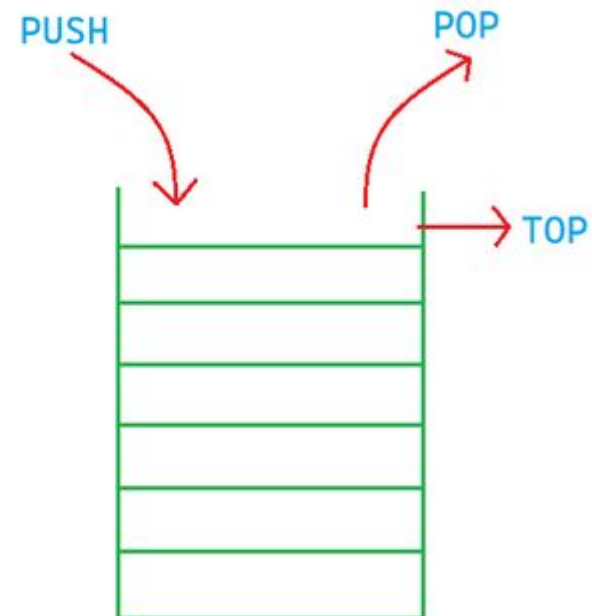
Se cambió la estructura de almacenamiento de datos de una lista enlazada por un árbol de búsqueda binaria, con el fin de mejorar la respuesta de compra y venta de productos.



[1]

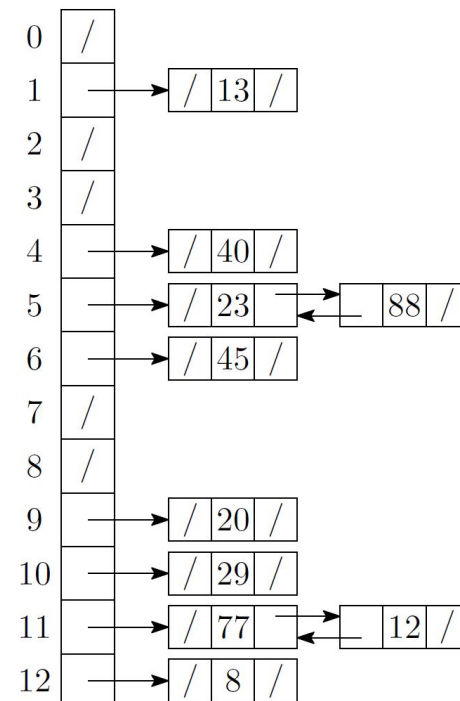
## Implementación de nuevas estructuras de datos en la solución del problema planteado

Para guardar las transacciones históricas, se implementó una pila que guarda los valores de compra y venta junto con la fecha



## Implementación de nuevas estructuras de datos en la solución del problema planteado

Para guardar los usuarios, se utiliza una tabla hash con encadenamiento que utiliza el nombre del usuario para calcular su posición.



## Pruebas y análisis comparativo del uso de las nuevas estructuras de datos implementadas

Agregar inventario				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,24	2,90	34,37	415,3

Comprar inventario				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,30	3,20	45,34	594,19

Eliminar inventario				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,15	2,80	30,08	360,97

Imprimir Productos (con print())				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,17	1,41	14,23	143,83



## Pruebas y análisis comparativo del uso de las nuevas estructuras de datos implementadas

Agregar usuario				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,02	0,20	1,61	15,19

Acceder				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,01	0,13	1,23	12,37

Transacciones históricas (sin print())				
N de datos	10 mil	100 mil	1 millón	10 millones
Tiempo (s)	0,001	0,01	0,09	0,99





## Conclusiones

Al trabajar en las diferentes estructuras, se pudo observar las diferentes opciones que nos brindan las estructuras de datos y como podemos implementarlas en diferentes proyectos, de manera que podamos almacenar la información y optimizar el programa según las necesidades del mismo.

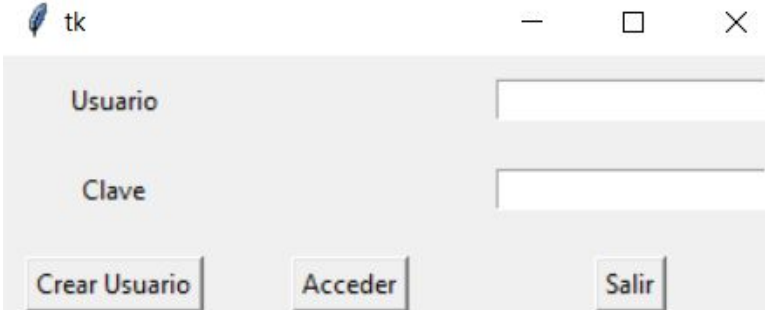
Es necesario considerar las diferentes dificultades, el gasto computacional y el gasto en memoria que cada estructura tiene con el fin de evitar fallos y que funcione correctamente en dispositivos de diversa índole.



## Dificultades y lecciones aprendidas

Las mayores dificultades fueron el manejo de los nodos en un árbol AVL, para poder modificar los nodos, obtener la altura y poder realizar las respectivas rotaciones, además del diseño de la función polyhash y del guardado de los datos después de cerrar el programa

Entre las dificultades sin resolver, está el imprimir el listado de productos y de transacciones en una interfaz gráficas



tk

Usuario

Clave

Crear Usuario Acceder Salir



tk

Agregar Eliminar Comprar Vender

Buscar Productos en registro Transacciones realizadas Cerrar sesion

Nombre

Numero

Costo

Ingreso



## Referencias

1. Python Data Structure and Algorithm Tutorial – AVL Tree, recuperado de <https://setscholars.net/python-data-structure-and-algorithm-tutorial-avl-tree/>
2. Estructura de datos de pila, recuperado de:  
<http://dhtrust.org/instrucciones/estructura-de-datos-de-pila-introduccion-y-programa/>
3. Hash tables, recuperado de:  
<https://www.win.tue.nl/~kbuchin/teaching/JBP030/notebooks/P-hashtables-sol.html>