

## Resumen

En este proyecto se estudió el ensamblaje de una PC con todos sus componentes fundamentales. Se tomaron en cuenta las posibles rutas de ensamblaje y se diseñaron en un autómata AFD para una mayor libertad de escoger una ruta de ensamblaje por el usuario.

El objetivo principal es la búsqueda del mejor ensamblaje en función de variables esenciales como lo son la necesidad y el presupuesto del usuario, dando como resultado el ensamblaje del mejor PC posible con sus principales componentes, esto para ayudar a muchos usuarios nuevos en el mundo de los computadores.

## Introducción

Actualmente se sigue presentando un problema en las personas interesadas en armar un PC y es el desconocimiento de los diferentes o mejores componentes para el armado de un PC, gracias a esto se busca una solución para los menos experimentados en el tema ya que se requiere de un conocimiento previo para el armado de un PC. A causa de las diferentes tiendas y precios, se puede presentar un mayor problema ya que esto perjudica de gran manera al usuario, no solo desinformándolo sino perjudicando su economía gracias a los precios sobrevaluados en el mercado computacional.

Presentamos una solución a esta problemática con el desarrollo de un programa que te da las diferentes opciones para el ensamblaje de un PC en base al presupuesto y a las diferentes necesidades de la persona que necesite de esta ayuda. Así mismo, presentamos una guía con los diferentes precios basados en consultas hechas en las principales tiendas de estos productos (Amazon, Aliexpress, etc.).

## Proceso y método

La realización de este proyecto se efectuó en base a cuatro principales fases metodológicas:

- **Investigación:** de los diferentes temas para la realización del proyecto, decidimos usar autómatas AFD, ya que toma una ruta simple desde el autómata inicial hasta el autómata final.
- **Elaboración:** diseñamos un autómata que contiene todos los estados de acuerdo a los componentes y sus diferentes rutas hacía los componentes principales, realizamos diferentes tablas con las referencias de cada componente y un precio de mercado actual aproximado.
- **Codificación:** de dos diferentes programas, uno con la implementación del autómata AFD con la librería automatalib y otro con el cálculo exacto del valor total de la suma de todos los componentes en el desarrollador de software de código abierto Jupyter Notebook, y el lenguaje Python.
- **Análisis:** de los resultados obtenidos para diferentes rutas en el autómata diseñado observando los diferentes ensamblajes posibles en base a un presupuesto ya definido.



Figura 1. Logo Jupyter Notebook.

Monitor (tasa refresco)	Precio (USD)
60 hz	110
75 hz	142
144 hz	195

Figura 2. Una de las tablas de precios de un componente de acuerdo al mercado actual.

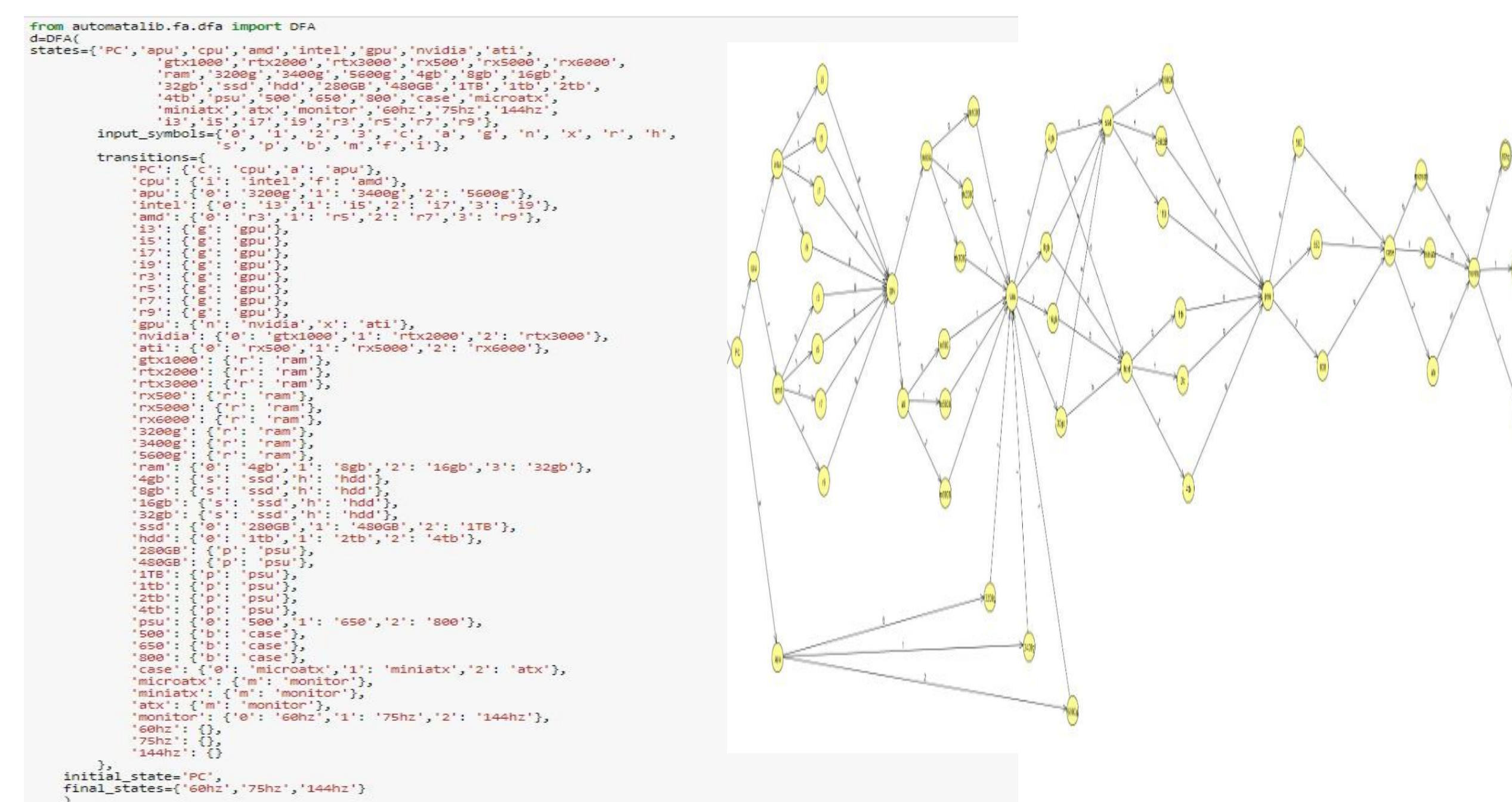
## Resultados

En la realización de este proyecto se observó que hay diferentes formas de armar un PC, ya sea con cpu y un ati o solamente con una apu, todo esto dependiendo del usuario.

En la implementación del autómata gracias a la utilización de la librería automatalib se examina una ruta simple desde el autómata inicial de PC hasta el autómata final de monitor, cada componente con su respectiva referencia, este nos da una palabra que se introduce en la ejecución del segundo programa para dar a conocer el precio total de la ruta de componentes elegida por el usuario obteniendo el mejor resultado esperado por el usuario, el segundo programa arroja una frase de error si la palabra digitada es incorrecta ya que está pensado en base al autómata del primer programa.

A continuación se muestra la parte de gráfica del autómata junto a su implementación y el apartado de suma de la palabra obtenida en la implementación:

Ejecución del programa 1. Implementación del automata AFD



Ejecución del programa parte 2. Suma de precios de los componentes

```
def checker(cadena,DFA):
    try:
        DFA.validate_input(cadena)
        dic_precios = {'c': {'0': 0, '1': {'0': 185, '1': 245, '2': 376, '3': 467},
        'f': {'0': 187, '1': 287, '2': 411, '3': 518}, 'a': {'0': 217, '1': 265, '2': 315},
        'g': {'0': 0, '1': {'0': 218, '1': 512, '2': 725}, 'x': {'0': 287, '1': 528, '2': 698},
        'h': {'0': 26, '1': 35, '2': 68, '3': 87}, 's': {'0': 27, '1': 49, '2': 105},
        'n': {'0': 47, '1': 51, '2': 83}, 'p': {'0': 37, '1': 48, '2': 89}, 'b': {'0': 68, '1': 78, '2': 88},
        'm': {'0': 118, '1': 142, '2': 195 }}
        suma = 0
        for i in range(0,len(cadena)):
            for k in dic_precios:
                if k == cadena[i] and k != 'c' and k != 'g':
                    suma+=dic_precios[k][cadena[i+1]]
        return print("Su compra es valida: los articulos comprados fueron:",
        [step for step in DFA.validate_input(cadena, step=True)],"y su precio calculado fue de: ",suma, "USD")
    except Exception as e:
        print("El automata no puede definir esta build \n", e)
checker('cl3gn2r33p2b2n2',d)
```

Su compra es valida: los articulos comprados fueron: ['PC', 'cpu', 'intel', 'i9', 'gpu', 'nvidia', 'rtx3080', 'ram', '32gb', 's sd', '1tb', 'psu', '800w', 'case', 'atx', 'monitor', '144hz'] y su precio calculado fue de: 1748 USD

## Trabajo Futuro

Finalizado el proyecto hay diferentes opciones para seguir con este añadiéndole diferentes funcionalidades, un posible añadido a futuro en este programa sería el de una mayor cantidad de opciones de personalización para una mayor sensación de satisfacción por parte del usuario.

## Información de contacto

Brayan Ferney Uribe Rodríguez, Email: [brayan2191946@correo.uis.edu.co](mailto:brayan2191946@correo.uis.edu.co)  
Fabian Ernesto Hernández Ramírez, Email: [fabian2191925@correo.uis.edu.co](mailto:fabian2191925@correo.uis.edu.co)  
Santiago Quintero Jaimes, Email: [santiago2180445@correo.uis.edu.co](mailto:santiago2180445@correo.uis.edu.co)

Gustavo Adolfo Garzón Villamizar, Email: [gustavo.garzon@saber.uis.edu.co](mailto:gustavo.garzon@saber.uis.edu.co)

## Referencias Bibliográficas (en formato APA)

1. Pypi.org (2021). Librería Automatalib v5.0.0. Recuperado en 13 de Octubre de 2021, de <https://pypi.org/project/automata-llb/>.