



Centro de Investigación en Computación



Nombre: Daniela Téllez Jiménez

Curso: Introducción a CHATGPT con  
Python

**Práctica 102**

Profesor: Alan Badillo Salas

Julio 2023

## Introducción

CHATGPT es un sistema de chat con inteligencia artificial. Se trata de una de las IA más capaces que hemos probado en los últimos tiempos, siendo capaz de responder a cualquier cosa que le pidas, y de hacer muchas cosas que le solicites.

## Justificación

Gracias al uso del CHATGPT ha permitido facilitar ciertas tareas como redactar textos complejos, ya sea académicos, periodísticos, informativos, traducir textos de un idioma a otro, crear código de programación, etc. Además, entre más concretas sean las peticiones del usuario, más podrá afinar en sus respuestas.

## Practica

D

Solicita a CHATGPT que genere el código para abrir un archivo weight-height.csv desde la URL <https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv> usando Pandas

```
import pandas as pd

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Mostrar los primeros registros del DataFrame para verificar que se haya cargado correctamente
print(df)
```

El resultado esperado fue el siguiente:

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801
...	...	...	...
9995	Female	66.172652	136.777454
9996	Female	67.067155	170.867906
9997	Female	63.867992	128.475319
9998	Female	69.034243	163.852461
9999	Female	61.944246	113.649103

[10000 rows x 3 columns]

**D**

genere el código que imprima 10 valores aleatorios del dataset anterior usando la función `sample` de Pandas

```
import pandas as pd

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Mostrar los primeros registros del DataFrame para verificar que se haya cargado correctamente
a=df.sample(5)
print(a)
```

El resultado esperado fue el siguiente:

	Gender	Height	Weight
8092	Female	60.414751	124.706690
4688	Male	67.297973	175.338660
8857	Female	63.006480	131.387133
1159	Male	71.647580	200.286509
5465	Female	63.973925	133.459503

**D**

genere el código que describa la columna Gender que es categórica, pide que extraiga los valores únicos y sus conteos

```
import pandas as pd

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Describir la columna "Gender" (categórica)
descripcion_gender = df["Gender"].describe()

# Extraer los valores únicos de la columna "Gender"
valores_unicos_gender = df["Gender"].unique()

# Contar la frecuencia de cada valor en la columna "Gender"
conteo_gender = df["Gender"].value_counts()

# Mostrar los resultados
print("Descripción de la columna 'Gender':")
print(descripcion_gender)
print("\nValores únicos de la columna 'Gender':")
print(valores_unicos_gender)
print("\nConteo de valores en la columna 'Gender':")
print(conteo_gender)
```

El resultado esperado fue el siguiente:

```
Descripción de la columna 'Gender':
count      10000
unique         2
top         Male
freq        5000
Name: Gender, dtype: object

Valores únicos de la columna 'Gender':
['Male' 'Female']

Conteo de valores en la columna 'Gender':
Male      5000
Female    5000
Name: Gender, dtype: int64
```

**D**

genere el código que describa la columna Height que es numérica y obtenga los estadísticos: total, mínimo, máximo, promedio, desviación estándar, intervalo de confianza al 95%, y los cuartiles

```
import pandas as pd
import numpy as np

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Describir la columna "Height" (numérica)
descripcion_height = df["Height"].describe()

# Calcular el promedio de la columna "Height"
promedio_height = df["Height"].mean()

# Calcular la desviación estándar de la columna "Height"
desviacion_estandar_height = df["Height"].std()

# Calcular el intervalo de confianza al 95% de la columna "Height"
intervalo_confianza_95_height = 1.96 * (desviacion_estandar_height / np.sqrt(len(df)))

# Calcular los cuartiles de la columna "Height"
cuartiles_height = df["Height"].quantile([0.25, 0.5, 0.75])

# Mostrar los resultados
print("Descripción de la columna 'Height':")
print(descripcion_height)
print("\nPromedio de la columna 'Height':", promedio_height)
print("Desviación estándar de la columna 'Height':", desviacion_estandar_height)
print("Intervalo de confianza al 95% de la columna 'Height':", intervalo_confianza_95_height)
print("\nCuartiles de la columna 'Height':")
print(cuartiles_height)
```

El resultado esperado fue el siguiente:

```
Descripción de la columna 'Height':
count    10000.000000
mean       66.367560
std        3.847528
min       54.263133
25%       63.505620
50%       66.318070
75%       69.174262
max       78.998742
Name: Height, dtype: float64

Promedio de la columna 'Height': 66.36755975482124
Desviación estándar de la columna 'Height': 3.8475281207732293
Intervalo de confianza al 95% de la columna 'Height': 0.07541155116715528

Cuartiles de la columna 'Height':
0.25    63.505620
0.50    66.318070
0.75    69.174262
Name: Height, dtype: float64
```

**D**

genere el código que describa la columna Weight que es numérica y obtenga los estadísticos: total, mínimo, máximo, promedio, desviación estándar, intervalo de confianza al 95%, y los cuartiles

```
import pandas as pd
import numpy as np

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Describir la columna "Weight" (numérica)
descripcion_weight = df["Weight"].describe()

# Calcular el promedio de la columna "Weight"
promedio_weight = df["Weight"].mean()

# Calcular la desviación estándar de la columna "Weight"
desviacion_estandar_weight = df["Weight"].std()

# Calcular el intervalo de confianza al 95% de la columna "Weight"
intervalo_confianza_95_weight = 1.96 * (desviacion_estandar_weight / np.sqrt(len(df)))

# Calcular los cuartiles de la columna "Weight"
cuartiles_weight = df["Weight"].quantile([0.25, 0.5, 0.75])

# Mostrar los resultados
print("Descripción de la columna 'Weight':")
print(descripcion_weight)
print("\nPromedio de la columna 'Weight':", promedio_weight)
print("Desviación estándar de la columna 'Weight':", desviacion_estandar_weight)
print("Intervalo de confianza al 95% de la columna 'Weight':", intervalo_confianza_95_weight)
print("\nCuartiles de la columna 'Weight':")
print(cuartiles_weight)
```

El resultado esperado fue el siguiente:

```
Descripción de la columna 'Weight':
count    10000.000000
mean      161.440357
std       32.108439
min       64.700127
25%      135.818051
50%      161.212928
75%      187.169525
max       269.989699
Name: Weight, dtype: float64

Promedio de la columna 'Weight': 161.44035683283076
Desviación estándar de la columna 'Weight': 32.108439006519596
Intervalo de confianza al 95% de la columna 'Weight': 0.6293254045277841

Cuartiles de la columna 'Weight':
0.25    135.818051
0.50    161.212928
0.75    187.169525
Name: Weight, dtype: float64
```



genere el código que grafique el histograma de la columna Height usando Seaborn

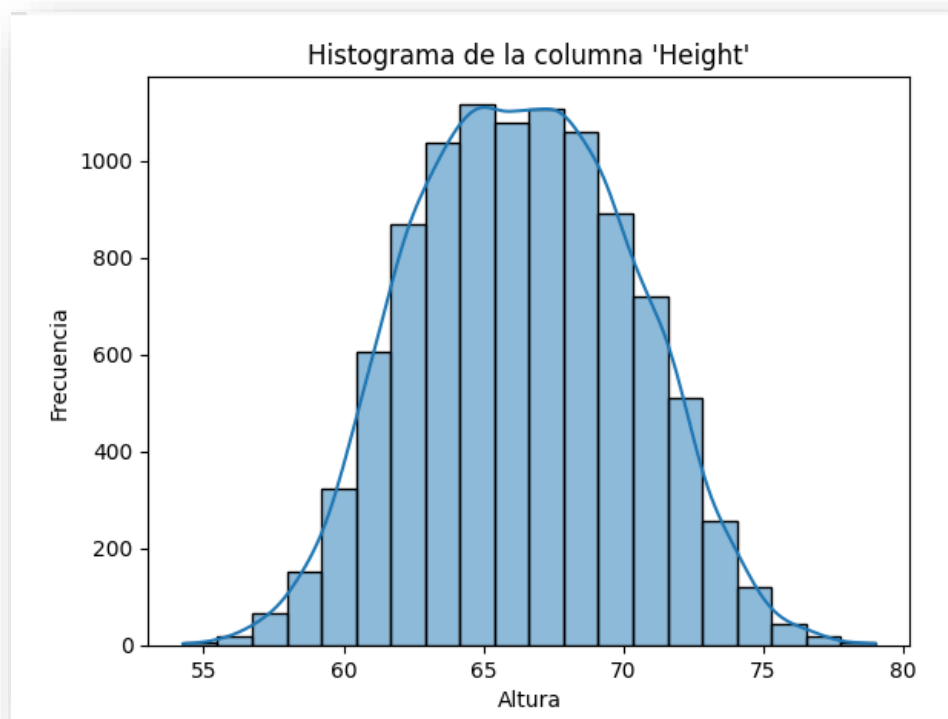
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Graficar el histograma de la columna "Height" utilizando Seaborn
sns.histplot(data=df, x="Height", bins=20, kde=True)
plt.title("Histograma de la columna 'Height'")
plt.xlabel("Altura")
plt.ylabel("Frecuencia")
plt.show()
```

El resultado esperado fue el siguiente:



**D**

genere el código que grafique el histograma de la columna Weight usando Seaborn

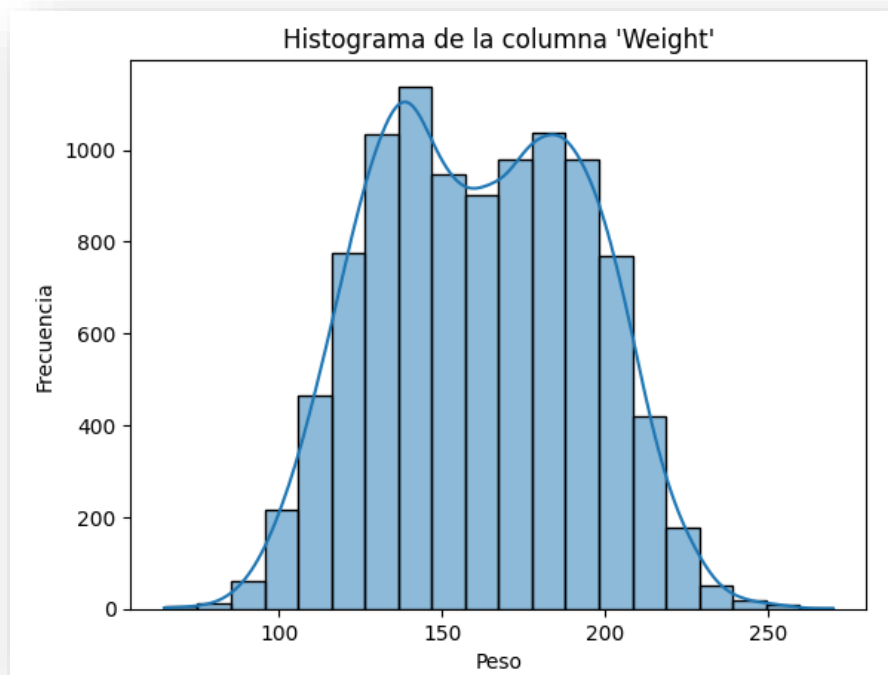
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Graficar el histograma de la columna "Weight" utilizando Seaborn
sns.histplot(data=df, x="Weight", bins=20, kde=True)
plt.title("Histograma de la columna 'Weight'")
plt.xlabel("Peso")
plt.ylabel("Frecuencia")
plt.show()
```

El resultado esperado fue el siguiente:





**D** genere el código que grafique la correlación entre las columnas Height y Weight usando Seaborn

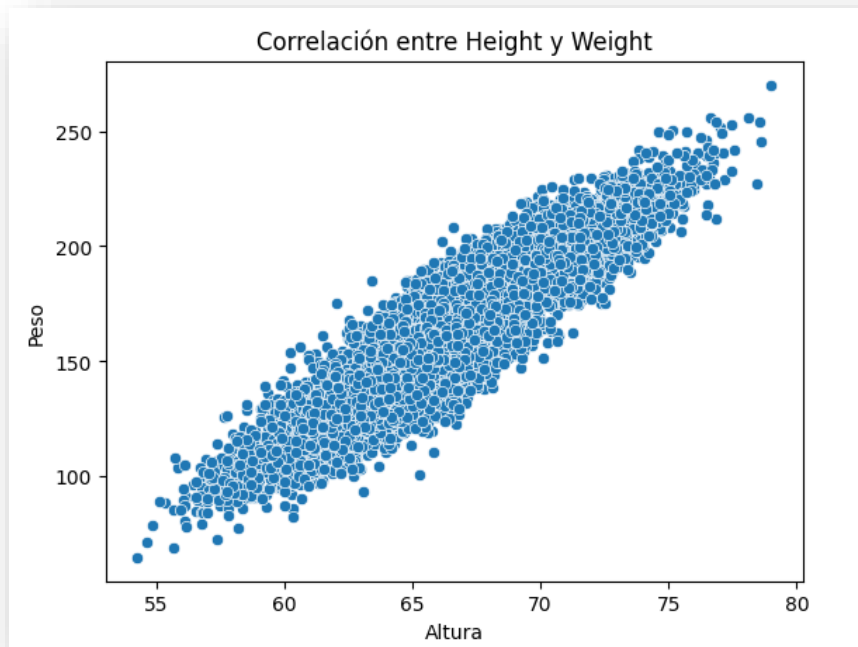
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Graficar la correlación entre las columnas "Height" y "Weight" utilizando Seaborn
sns.scatterplot(data=df, x="Height", y="Weight")
plt.title("Correlación entre Height y Weight")
plt.xlabel("Altura")
plt.ylabel("Peso")
plt.show()
```

El resultado esperado fue el siguiente:



**D**

genere el código que obtenga la correlación entre las columnas Height y Weight usando corr de Numpy

```
import pandas as pd
import numpy as np

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Obtener la correlación entre las columnas "Height" y "Weight" utilizando Numpy
correlacion_height_weight = np.corrcoef(df["Height"], df["Weight"])[0, 1]

print("Correlación entre Height y Weight:", correlacion_height_weight)
```

Correlación entre Height y Weight: 0.9247562987409146

El resultado esperado fue el siguiente:

Correlación entre Height y Weight: 0.9247562987409146

**D**

genere el código que reporte el valor promedio de la columna Height donde Gender es Male y donde Gender es Female y repita lo mismo para la columna Weight

```
import pandas as pd

# URL del archivo CSV
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/weight-height.csv"

# Leer el archivo CSV desde la URL
df = pd.read_csv(url)

# Calcular el valor promedio de la columna "Height" para Gender Male y Female
promedio_height_male = df[df["Gender"] == "Male"]["Height"].mean()
promedio_height_female = df[df["Gender"] == "Female"]["Height"].mean()

# Calcular el valor promedio de la columna "Weight" para Gender Male y Female
promedio_weight_male = df[df["Gender"] == "Male"]["Weight"].mean()
promedio_weight_female = df[df["Gender"] == "Female"]["Weight"].mean()

# Reportar los resultados
print("Valor promedio de la columna 'Height' para Gender Male:", promedio_height_male)
print("Valor promedio de la columna 'Height' para Gender Female:", promedio_height_female)
print("\nValor promedio de la columna 'Weight' para Gender Male:", promedio_weight_male)
print("Valor promedio de la columna 'Weight' para Gender Female:", promedio_weight_female)
```

El resultado esperado fue el siguiente:

```
Valor promedio de la columna 'Height' para Gender Male: 69.02634590621741
Valor promedio de la columna 'Height' para Gender Female: 63.70877360342507

Valor promedio de la columna 'Weight' para Gender Male: 187.0206206581932
Valor promedio de la columna 'Weight' para Gender Female: 135.86009300746835
```

## Ejercicio 2

D

genere el código que extraiga el texto de una imagen PNG desde la URL [https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/UITD\\_PAPER.png](https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/UITD_PAPER.png) usando OCR en Python

```
import requests
from PIL import Image
import pytesseract

# URL de la imagen PNG
url = "https://raw.githubusercontent.com/dragonnomada/chat-gpt-2023/main/datasets/UITD_PAPER.png"

# Obtener la imagen desde la URL
response = requests.get(url, stream=True)
response.raise_for_status()

# Crear una imagen PIL desde la respuesta
image = Image.open(response.raw)

# Realizar OCR en la imagen para extraer el texto
texto_extraido = pytesseract.image_to_string(image)

# Mostrar el texto extraído
print("Texto extraído de la imagen:")
print(texto_extraido)
```

El resultado esperado fue el siguiente:

```
Texto extraído de la imagen:
1. "Saving Time with the User Interface Transition Diagrams (UITD) Editor"
2. "The UITD Editor: A Tool to Simplify User-System Interaction Modeling"
3. "Streamlining User Interface Design with the UITD Editor: A Comparative
Study"

4. "UITD Editor: Simplifying User-System Interaction Modeling for
Developers"

Can the User Interface Transition Diagrams editor (UITD editor) save users' time?

Cervantes-Ojeda J., Badillo-Salas A., Gomez-Fuentes M.C.
Department of Applied Mathematics and Systems
Universidad Autonoma Metropolitana
México City, México
jcervantes@cua.uam.mx, dragonnomada123@gmail.com, mgomez@cua.uam.mx

Abstract- The User Interface Transition Diagram
(UITD) is a formal modeling notation that simplifies
the specification and design of user-system
```

D

traduzca el texto a español:

Texto extraído de la imagen:

1. "Saving Time with the User Interface Transition Diagrams (UITD) Editor"
2. "The UITD Editor: A Tool to Simplify User-System Interaction Modeling"
3. "Streamlining User Interface Design with the UITD Editor: A Comparative



Texto extraído de la imagen:



1. "Ahorrando tiempo con el Editor de Diagramas de Transición de Interfaz de Usuario (UITD)"
2. "El Editor UITD: Una herramienta para simplificar la modelización de interacciones usuario-sistema"
3. "Optimizando el diseño de interfaz de usuario con el Editor UITD: Un estudio comparativo"
4. "Editor UITD: Simplificando la modelización de interacciones usuario-sistema para desarrolladores"

¿Puede el editor de Diagramas de Transición de Interfaz de Usuario (UITD editor) ahorrar tiempo a los usuarios?

Cervantes-Ojeda J., Badillo-Salas A., Gomez-Fuentes M.C.

Departamento de Matemáticas Aplicadas y Sistemas

Universidad Autónoma Metropolitana

D

quiénes son los autores del artículo



Los autores del artículo son:

- Cervantes-Ojeda J.
- Badillo-Salas A.
- Gomez-Fuentes M.C.

D cuáles son los correos relacionados al artículo?



En el texto proporcionado no se incluyen direcciones de correo electrónico completas para los autores. Los correos electrónicos de los autores se mencionan de la siguiente manera:



- [jcervantes@cua.uam.mx](mailto:jcervantes@cua.uam.mx)
- [dragonnomadal23@gmail.com](mailto:dragonnomadal23@gmail.com)
- [mgomez@cua.uam.mx](mailto:mgomez@cua.uam.mx)

D quién es la universidad, el departamento, la ciudad y el país del artículo?



Según el texto proporcionado, la afiliación de los autores se menciona de la siguiente manera:



- Universidad: Universidad Autónoma Metropolitana
- Departamento: Departamento de Matemáticas Aplicadas y Sistemas
- Ciudad: Ciudad de México
- País: México

Estos detalles indican que la universidad es la Universidad Autónoma Metropolitana, el departamento es el Departamento de Matemáticas Aplicadas y Sistemas, la ciudad es la Ciudad de México, y el país es México.

Regenerate




## explica el ABSTRACT

El "Abstract" es una sección crucial en un artículo académico o científico que resume brevemente el contenido, los objetivos, los resultados clave y las implicaciones del estudio. Su función principal es proporcionar a los lectores una visión general rápida y concisa del artículo sin necesidad de leer todo el contenido. El "Abstract" es especialmente útil para aquellos que buscan determinar si el artículo es relevante para sus intereses de investigación o si desean obtener una idea de los resultados sin profundizar en los detalles técnicos.

En el "Abstract" que proporcionaste, los autores describen la esencia de su investigación. Aquí hay una breve explicación de los puntos clave:

1. **El problema:** Se introduce el "User Interface Transition Diagram" (UITD), una notación de modelado formal que simplifica la especificación y el diseño de las interacciones entre usuarios y sistemas.

2. **Importancia:** Se destaca  comunicar tanto con interesados técnicos como no técnicos durante la fase de obtención de requisitos. Además, se menciona que es fácil de entender y completo.
3. **Método:** Los autores llevaron a cabo un estudio comparativo para evaluar la eficiencia de la creación de UITD utilizando el software "draw.io" en comparación con un editor UITD dedicado.
4. **Resultados:** Se indica que los resultados mostraron que el editor UITD fue más eficiente y preferido por los participantes.
5. **Implicaciones:** Los hallazgos tienen implicaciones importantes para desarrolladores de software, diseñadores y otros interesados involucrados en la especificación y el diseño de las interacciones usuario-sistema. Además, sugieren que el uso de herramientas



## explique la introducción

La introducción de un artículo académico o científico es una sección fundamental que cumple varios propósitos esenciales. Su función principal es contextualizar el estudio, presentar el problema de investigación, destacar la importancia de dicho problema y, a menudo, proporcionar un vistazo de los objetivos y la estructura del artículo. La introducción tiene como objetivo captar la atención del lector, establecer la base para el trabajo realizado y justificar la necesidad de investigar el tema.

En la introducción proporcionada:

1. **Contexto:** Se introduce el concepto de "User Interface Transition Diagram" (UITD) como una notación formal para modelar las interacciones entre usuarios y sistemas. Esto establece el escenario al identificar el tema central del artículo: la simplificación de la especificación y el diseño de estas interacciones.

2. **Relevancia:** Se destaca la herramienta de comunicación valiosa durante la fase de obtención de requisitos, y se menciona que es fácil de entender y completo. Esto establece el contexto para comprender por qué se está estudiando esta notación y su importancia potencial.

3. **Problema:** Se plantea una pregunta fundamental: ¿cuál es la eficiencia del proceso de creación de UITDs utilizando un software general (draw.io) en comparación con un editor especializado de UITD? Esto plantea el problema de investigación y establece la base para el estudio comparativo.

4. **Objetivos:** Aunque no se mencionan explícitamente en el fragmento proporcionado, la introducción a menudo presenta los objetivos específicos del estudio o lo que los autores buscan lograr a través de su investigación. Estos objetivos pueden incluir evaluar, comparar,



D

extraiga las palabras principales

A continuación, he extraído las palabras clave y términos importantes del texto proporcionado:

1. User Interface Transition Diagram (UITD)
2. Editor de UITD
3. Simplificación
4. Modelado
5. Especificación
6. Diseño
7. Interacciones usuario-sistema
8. Eficiencia
9. Draw.io
10. Comparativo
11. Herramientas especializadas
12. Obtención de requisitos
13. Desarrolladores de software
14. Diseñadores
15. Notación de modelado formal
16. Comunicación
17. Requisitos del sistema
18. Estudio comparativo
19. Valoración de herramientas
20. Implicaciones

## Título del artículo

### Autores

- Cervantes-Ojeda J.
- Badillo-Salas A.
- Gomez-Fuentes M.C.

### Resumen

El *User Interface Transition Diagram* (UITD) es una notación de modelado formal que simplifica la especificación y el diseño de las interacciones entre usuarios y sistemas. Es una valiosa herramienta de comunicación para los interesados técnicos y no técnicos durante la fase de obtención de requisitos, ya que proporciona una notación simple y completa que es fácil de entender. En este artículo, investigamos la eficiencia de crear UITDs utilizando draw.io, un software de diagramación ampliamente utilizado, en comparación con un editor UITD dedicado. Los resultados muestran que el editor UITD es más eficiente y preferido por los participantes, lo que resalta la importancia de utilizar herramientas especializadas para crear modelos formales como UITDs. Los hallazgos de este estudio tienen implicaciones para desarrolladores de software, diseñadores y otros interesados involucrados en la especificación y el diseño de interacciones usuario-sistema.

### Introducción

El *User Interface Transition Diagram* (UITD) es una notación de modelado formal que simplifica la especificación y el diseño de interacciones usuario-sistema. Puede utilizarse para modelar el flujo de las interfaces de usuario que tendrá el sistema, convirtiéndose en una valiosa herramienta de comunicación para los interesados técnicos y no técnicos durante la fase de obtención de requisitos. Además, es técnicamente preciso, lo que permite utilizarlo de manera confiable para iniciar el desarrollo del sistema [1].

Simplicidad es clave cuando se trabaja con notaciones de modelado, especialmente para no expertos [3]. El UITD proporciona una notación que es completa y lo suficientemente simple para que los interesados no técnicos lo entiendan [2]. Es una notación de modelado formal que es fácil de aprender, facilitando la comunicación entre interesados con y sin habilidades técnicas en el desarrollo de software. En comparación con otras herramientas de modelado formal, el UITD tiene varias ventajas, incluyendo su simplicidad, completitud, y la capacidad de modelar interacciones usuario-sistema de manera clara y concisa.

D

genere el código para que reescriba el artículo en forma Latex

```

\documentclass{article}

\title{Título del artículo}
\author{Cervantes-Ojeda J., Badillo-Salas A.,
Gomez-Fuentes M.C.}
\date{} % Dejar la fecha vacía

\begin{document}
\maketitle

\begin{abstract}
El \textit{User Interface Transition Diagram} (UITD)
es una notación de modelado formal que simplifica la
especificación y el diseño de las interacciones entre
usuarios y sistemas. Es una valiosa herramienta de
comunicación para los interesados técnicos y no
técnicos durante la fase de obtención de requisitos,
ya que proporciona una notación simple y completa que
es fácil de entender. En este artículo, investigamos
la eficiencia de crear UITDs utilizando \textit{draw.
io}, un software de diagramación ampliamente
utilizado, en comparación con un editor UITD dedicado.
Los resultados muestran que el editor UITD es más

```

Los hallazgos de este estudio tienen implicaciones para desarrolladores de software, diseñadores y otros interesados involucrados en la especificación y el diseño de interacciones usuario-sistema.

```
\end{abstract}
```

```
\section{Introducción}
```

El \textit{User Interface Transition Diagram} (UITD) es una notación de modelado formal que simplifica la especificación y el diseño de interacciones usuario-sistema. Puede utilizarse para modelar el flujo de las interfaces de usuario que tendrá el sistema, convirtiéndose en una valiosa herramienta de comunicación para los interesados técnicos y no técnicos durante la fase de obtención de requisitos. Además, es técnicamente preciso, lo que permite utilizarlo de manera confiable para iniciar el desarrollo del sistema [1].

Simplicidad es clave cuando se trabaja con notaciones de modelado, especialmente para no expertos [3]. El UITD proporciona una notación que es completa y lo suficientemente simple para que los interesados no técnicos lo entiendan [2]. Es una notación de modelado



genere el código para escribir el artículo generado de Markdown en un archivo PDF

```
pandoc articulo.md -o articulo.pdf
```

## Conclusión

Durante la práctica se desarrolló programación Python, conceptos de programación, manipulación de datos, estadísticas, visualización y procesamiento de texto. Así como el manejo de ciertas bibliotecas como: Pandas, Seaborn, y cómo convertir contenido entre diferentes formatos como Markdown y PDF.

En resumen, hicimos una introducción a la programación que nos permitirá entender poco a poco el concepto de análisis de datos debido a que por medio de la biblioteca Pandas nos permite el manejo de grandes cantidades de información para que posteriormente sean visualizadas en diferentes graficas.