



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN



INGENIERÍA EN CIENCIA DE DATOS

NOMBRE:

MARTHA PATRICIA ALVAREZ CARRILLO

MATERIA:

INTRODUCCIÓN A LA CIENCIA DE DATOS

PROFESOR:

JAIMÉ ALEJANDRO ROMERO SIERRA

PROYECTO FINAL

ANÁLISIS DE LA ROTACIÓN DE DESARROLLADORES

EN LA INDUSTRIA TECNOLÓGICA

FECHA DE ENTREGA:

27/NOVIEMBRE/2024



PROYECTO:

ANÁLISIS

DE LA

ROTACIÓN DE

DESARROLLADORES

EN LA

INDUSTRIA TECNOLÓGICA

1) Introducción

1) Descripción del Proyecto:

Este proyecto tiene como objetivo analizar la rotación de empleados dentro de una empresa para identificar patrones que permitan predecir qué empleados tienen una mayor probabilidad de abandonar la organización. A través del análisis de variables como el salario, la antigüedad y el nivel educativo, buscamos establecer qué factores inciden más en la retención de personal.

A. Objetivo del Proyecto

- **"Reducir la tasa de abandono de empleados en un 10% durante el próximo año".**

El objetivo es tanto ambicioso como necesario, ofreciendo una meta clara que permitirá evaluar el éxito del proyecto. La reducción específica del 10% en la rotación de empleados puede traducirse en un ahorro significativo en costos asociados con la contratación y la capacitación, además de mejorar la continuidad del conocimiento y la experiencia dentro de la empresa. Este objetivo también implica un compromiso hacia el bienestar de los empleados, buscando crear un ambiente donde se sientan valorados y motivados para permanecer. Al fijar una meta cuantificable, el proyecto se orienta hacia resultados concretos, que son cruciales para la sostenibilidad a largo plazo de cualquier organización en el sector tecnológico.

B. Descripción del Problema

- La industria tecnológica se enfrenta a un fenómeno alarmante: una alta tasa de rotación de empleados.
- La rotación de empleados ha emergido como un desafío crítico para la organización, impactando tanto su estabilidad como su rendimiento general. Recientemente, se ha observado un incremento notable en las tasas de rotación, particularmente entre ciertos grupos demográficos y posiciones clave. Este fenómeno no solo implica la pérdida de talento, sino también un costo financiero significativo asociado a la contratación y capacitación de nuevos empleados. La falta de continuidad en los equipos de trabajo puede afectar la moral del personal restante, generar ineficiencias operativas y disminuir la calidad del servicio ofrecido a los clientes.
- Uno de los factores que contribuyen a la rotación es la insatisfacción laboral, que puede estar relacionada con múltiples elementos, como la falta de oportunidades de desarrollo profesional, una cultura organizacional deficiente o una compensación que no se alinea con las expectativas del mercado. Además, los datos indican que los empleados con menos años de experiencia tienden a abandonar la empresa con mayor frecuencia, lo que sugiere una falta de integración o un desajuste en las expectativas respecto al ambiente laboral.
- El problema se agrava al observar que ciertas posiciones críticas, como los analistas y gerentes, están siendo ocupadas por personas con alta rotación, lo que podría comprometer la calidad del liderazgo y la dirección estratégica de la organización. Esta situación plantea interrogantes sobre la eficacia de las políticas de retención actuales y la percepción que los empleados tienen sobre su lugar en la empresa.

- Por todo lo anterior, es esencial abordar este problema de manera integral. Este proyecto se propone no solo identificar las causas de esta alta rotación, sino también ofrecer recomendaciones basadas en datos concretos. La identificación de las causas de rotación, no solo permitirá a la organización implementar estrategias de retención más efectivas, sino que también facilitará la creación de un entorno laboral que fomente la satisfacción, el compromiso y la motivación para mejorar con la continuidad del conocimiento y la experiencia dentro de la empresa. Al hacerlo, la empresa no solo podrá reducir la rotación, sino que también fortalecerá su reputación como un lugar de trabajo atractivo comprometido con el bienestar de sus empleados, lo que, a su vez, atraerá talento de calidad y aumentará la competitividad en el mercado.

C. Definición de Stakeholders Clave

- Identificar a las partes interesadas en el proyecto y su rol es crucial para comprender quiénes se verán afectados por los resultados del proyecto y cómo se utilizarán los hallazgos. Esto también ayuda a establecer un canal de comunicación efectivo para asegurar que las soluciones propuestas sean viables y alineadas con las necesidades de la organización. La colaboración de los stakeholders no solo enriquecerá el análisis, sino que también aumentará la probabilidad de éxito en la implementación de soluciones.
 - **Gerentes de Recursos Humanos:** Implementarán estrategias basadas en los hallazgos.
 - Este departamento será el motor detrás de la implementación de estrategias de retención, encargándose de diseñar políticas que fomenten un ambiente laboral positivo. Su experiencia será vital para interpretar los hallazgos y aplicarlos en acciones concretas que beneficien tanto a la organización como a los empleados.
 - **Equipo de Talento:** Identificará áreas problemáticas en la gestión del talento.
 - Los encargados de atraer y desarrollar talento desempeñarán un papel crucial en la identificación de áreas de mejora. Su capacidad para alinear las expectativas de los empleados con las necesidades organizacionales será clave para crear un entorno que promueva la retención.
 - **Desarrolladores actuales:** Su retención es fundamental para el éxito del análisis.
 - Involucrar a los empleados en el proceso de investigación mediante encuestas y entrevistas permitirá obtener información valiosa sobre sus percepciones y experiencias. Sus voces son fundamentales para entender las dinámicas internas y los factores que influyen en su decisión de permanecer o abandonar la organización.

D. Recursos Disponibles

Esta sección proporciona un panorama claro de las herramientas y tecnologías que se utilizarán en el análisis. Es fundamental especificar las herramientas de análisis y visualización, ya que esto impacta en la calidad y profundidad del análisis. La descripción de los datos disponibles permite a los stakeholders comprender qué información se utilizará para abordar las preguntas del proyecto.

- **Tecnología y Herramientas:**

- Para llevar a cabo este análisis, se utilizarán herramientas avanzadas de análisis de datos como *Python*, que permitirá la manipulación eficiente de grandes volúmenes de datos, y bibliotecas como Pandas para la exploración de datos. Para la visualización, se emplearán *Matplotlib* y *Seaborn*, que ayudarán a ilustrar patrones y tendencias de manera clara y comprensible. Estas herramientas son fundamentales para desentrañar la complejidad de los datos, permitiendo a los investigadores generar insights que podrían ser cruciales en la formulación de estrategias efectivas.

- **Datos:**

- La base de datos incluirá información demográfica y laboral de los empleados, tales como Edad, Género, Nivel Educativo, Título del Trabajo o Cargo, Años de Experiencia y Salario. Cada variable aporta un contexto valioso que facilitará el análisis de los factores que inciden en la rotación. Por ejemplo, la relación entre la experiencia laboral y la satisfacción podría ser un indicador clave, mientras que el análisis de la remuneración puede revelar discrepancias que afectan la retención. La combinación de estas herramientas y datos robustos es esencial para generar un análisis profundo que identifique los puntos críticos y las áreas de mejora.

E. Información de la Base de Datos

Esta sección proporciona una descripción detallada del conjunto de datos, lo que es crucial para planificar el análisis y seleccionar los métodos adecuados para obtener conclusiones significativas.

La Base de Datos consta de 7,421 registros de empleados, que incluyen las siguientes variables:

Columnas:

Número	Nombre de la Columna
1	Edad
2	Género
3	Nivel Educativo
4	Título del Trabajo o Cargo
5	Años de Experiencia
6	Salario.

Tipos de datos:

Número	Nombre de la Columna	Tipo de Dato	Tipo
1	Edad	Numérico	int
2	Género	Categórico	string
3	Nivel Educativo	Categórico	string
4	Título del Trabajo o Cargo	Categórico	string
5	Años de Experiencia	Numérico	int
6	Salario	Numérico	float

Esta variedad nos permitirá realizar análisis comparativos y estadísticos, identificando tendencias y correlaciones significativas que puedan ayudar a entender mejor los factores que afectan la rotación de empleados.

2) *Fuentes de Datos Identificadas*

- La identificación de fuentes de datos relevantes es esencial para garantizar que el análisis sea robusto y confiable. Cada fuente de datos contribuirá a un entendimiento más completo de las dinámicas que afectan la rotación de empleados, lo que permitirá a los investigadores abordar el problema desde múltiples ángulos.
 - **Registros de Empleados:** Estos registros incluirán datos demográficos y de desempeño, proporcionando un contexto fundamental para el análisis. Serán esenciales para identificar patrones de rotación y correlacionarlos con variables específicas.
 - **Encuestas de Satisfacción Laboral:** Recogerán la perspectiva de los empleados sobre su experiencia laboral, lo que permitirá identificar áreas críticas que necesitan atención. Estas encuestas son una herramienta valiosa para capturar la voz del empleado y entender sus motivaciones y preocupaciones.
 - **Evaluaciones de Desempeño:** Las métricas de rendimiento proporcionarán datos adicionales que ayudarán a establecer una relación entre el rendimiento y la probabilidad de abandono. Este análisis podrá revelar si los empleados más talentosos están abandonando la organización, lo que podría ser un indicativo de problemas en la cultura o el reconocimiento.

3) *Justificación del Proyecto*

- La rotación de empleados es un fenómeno crítico que afecta a las organizaciones de diversas maneras. En un entorno empresarial competitivo, la capacidad de retener talento se ha convertido en un indicador clave de la salud organizacional. Las empresas enfrentan costos significativos relacionados con la contratación y capacitación de nuevos empleados, así como la pérdida de conocimiento y experiencia acumulada cuando un empleado decide dejar la organización. Por lo tanto, es fundamental entender las causas subyacentes de la rotación y actuar en consecuencia.
- Este proyecto tiene como objetivo analizar la satisfacción laboral y los factores que influyen en la retención de empleados. A través de un enfoque basado en datos, se buscará identificar patrones y tendencias que revelen por qué ciertos grupos de empleados abandonan la organización más rápidamente que otros. Por ejemplo, se investigará si los empleados con menos años de experiencia tienen mayores probabilidades de dejar la empresa, o si existen diferencias significativas en la rotación entre distintos niveles educativos o entre géneros.
- La relevancia de este análisis radica en que no solo se trata de retener talento, sino de crear un ambiente laboral positivo y productivo. La satisfacción de los empleados no solo impacta en su deseo de permanecer en la organización, sino que también afecta directamente a la productividad, la innovación y el clima organizacional. Un empleado satisfecho es más propenso a comprometerse con los objetivos de la empresa, a colaborar con sus compañeros y a contribuir a un entorno de trabajo positivo.
- Además, este proyecto permitirá a la organización adoptar medidas proactivas y estratégicas para mejorar la retención de empleados. Las conclusiones derivadas del análisis servirán para implementar programas de formación y desarrollo profesional, mejorar las condiciones laborales y establecer políticas que fomenten un equilibrio entre la vida laboral y personal. Estas acciones no solo beneficiarán a los empleados, sino que también se traducirán en un aumento en la competitividad de la organización en el mercado.

- Finalmente, invertir en la retención de talento y en la satisfacción laboral no solo es un imperativo ético, sino que también representa una decisión financiera inteligente. Al reducir la rotación, la empresa puede optimizar sus recursos y maximizar su retorno de inversión en capital humano. En resumen, este proyecto no solo busca mitigar un problema existente, sino que pretende sentar las bases para una cultura organizacional sólida, sostenible y orientada hacia el éxito a largo plazo.

A. Preguntas Clave

- Las preguntas clave nos guiarán en el proceso de investigación y ayudarán a estructurar el análisis. Estas preguntas están diseñadas para profundizar en los aspectos de rotación que podrían no ser evidentes a primera vista y pueden conducir a descubrimientos significativos que informen las decisiones estratégicas.
 - ¿Qué factores específicos influyen en la decisión de un empleado de abandonar su puesto?
 - ¿Qué perfiles de empleados muestran una mayor propensión a la rotación?
 - ¿Cuáles son las prácticas de retención más efectivas en otras organizaciones del sector tecnológico?
 - ¿Cómo afecta el salario en la decisión de permanecer en la empresa?
 - ¿Qué relación existe entre el nivel educativo y la lealtad de los empleados (retención)?
 - ¿Qué roles tienen mayor rotación?
 - ¿Existen diferencias significativas por género que influyan en la rotación de personal?
 - ¿Existen roles específicos con tasas de abandono significativamente más altas que otros?
 - ¿Qué años de experiencia son más críticos para la retención?
 - ¿Cómo influyen los beneficios ofrecidos en la decisión de quedarse?
 - ¿Cómo impacta la cultura organizacional en la rotación de personal?
 - ¿Qué beneficios adicionales podrían influir en la decisión de quedarse en la empresa?
 - ¿Qué importancia tienen las oportunidades de desarrollo profesional en la retención?

B. Hipótesis Iniciales

- Estas hipótesis servirán como guías en el análisis, permitiendo a los investigadores probar su validez y, en el proceso, obtener insights que puedan transformar la gestión del talento en la organización. Cada hipótesis está diseñada para ser probada con los datos disponibles, lo que permitirá establecer conexiones significativas entre variables.
 - **Hipótesis 1:** "Los empleados con menos de 5 años de experiencia son más propensos a dejar la organización."
 - Esta hipótesis se fundamenta en la creencia de que los empleados novatos pueden sentir menos apego a la cultura organizacional y estar más abiertos a oportunidades externas.
 - **Hipótesis 2:** "La falta de un título de posgrado se correlaciona con mayores tasas de rotación."
 - Esto sugiere que los empleados con menor formación académica pueden percibir limitaciones en su crecimiento profesional dentro de la empresa.
 - **Hipótesis 3:** "Los empleados que perciben salarios inferiores a la media del sector tienen más probabilidades de abandonar la empresa."
 - Esta hipótesis aborda una realidad crítica en el mundo laboral actual, donde la compensación es un factor decisivo en la retención.

2) Metodología

Como parte de la Metodología abarcaremos 2 temas importantes:

- A. Proceso de Limpieza de Datos
- B. Análisis Exploratorio de Datos (EDA)

A. Proceso de Limpieza de Datos

Para garantizar la calidad de los datos y poder realizar un análisis adecuado, se llevó a cabo un proceso de limpieza de los datos, incluyendo la identificación y tratamiento de valores faltantes, datos duplicados y formatos inconsistentes en una base de datos. Este proceso es fundamental para asegurar la calidad y la integridad de los análisis posteriores. A continuación, se detallan los pasos:

1. Recepción de la Base de Datos

Descargar el archivo de la base de datos "ensuciada" proporcionado por el profesor. Este archivo contiene errores comunes, como valores duplicados, valores faltantes (NaNs) y errores en el formato de algunas columnas, que afectan el análisis y la calidad de los datos.

```
1 #Cargar DataFrame
2 from google.colab import drive
3 drive.mount('/content/drive')
```

```
1 #Carga el archivo CSV en un DataFrame llamado df.
2 import pandas as pd
3
4 df=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/practicas/CD/Base_fs_sucio.csv')
5 df
6 #print(df)
7
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	NaN	NaN	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0
...
7416	28.0	Male	Bachelor's	Software Engineer	3.0	NaN
7417	30.0	Female	Bachelor's Degree	Marketing Coordinator	5.0	95000.0
7418	35.0	Female	PhD	Director of Marketing	12.0	170000.0
7419	26.0	Male	Master's Degree	Digital Marketing Manager	3.0	50000.0
7420	30.0	Male	Bachelor's Degree	Software Engineer	4.0	65000.0

7421 rows × 6 columns

2. Análisis Inicial de la Base de Datos

Realizar un análisis preliminar para comprender la naturaleza y distribución de los errores.

Realizar un análisis preliminar para comprender la naturaleza y distribución de los errores.

```
1 df.shape
2 # 7,421
```

```
(7421, 6)
```

```
1 df.columns
```

```
Index(['Age', 'Gender', 'Education Level', 'Job Title', 'Years of Experience',
       'Salary'],
      dtype='object')
```

a) Visualizar valores únicos de las columnas

```
1 # Visualizar valores únicos de la columna [1]
2 df['Age'].unique()

→ array([32., 28., 45., 36., 52., 29., 42., 31., 26., 38., 48., 35., 40.,
       27., nan, 39., 25., 51., 34., 47., 30., 41., 37., 24., 43., 33.,
       50., 46., 49., 23., 53., 44., 61., 57., 62., 55., 56., 54., 60.,
       58., 22., 21.])
```

```
1 # Visualizar valores únicos de la columna [2]
2 df['Gender'].unique()
3

→ array(['Male', 'Female', nan, 'Other'], dtype=object)
```

```
1 # Visualizar valores únicos de la columna [3]
2 df['Education Level'].unique()
3

→ array(["Bachelor's", "Master's", 'PhD', nan, "Bachelor's Degree",
       "Master's Degree", 'High School', 'phD'], dtype=object)
```

```
1 # Visualizar valores únicos de la columna [4]
2 df['Job Title'].unique()
3
```

```
array(['Software Engineer', 'Data Analyst', 'Senior Manager', nan,
       'Director', 'Marketing Analyst', 'Product Manager',
       'Sales Manager', 'Marketing Coordinator', 'Senior Scientist',
       'Software Developer', 'HR Manager', 'Financial Analyst',
       'Project Manager', 'Customer Service Rep', 'Operations Manager',
       'Marketing Manager', 'Senior Engineer', 'Data Entry Clerk',
       'Sales Director', 'Business Analyst', 'VP of Operations',
       'IT Support', 'Recruiter', 'Financial Manager',
       'Social Media Specialist', 'Junior Developer', 'Product Designer',
       'CEO', 'Accountant', 'Data Scientist', 'Marketing Specialist',
       'Technical Writer', 'HR Generalist', 'Project Engineer',
       'Customer Success Rep', 'Sales Executive', 'UX Designer',
       'Operations Director', 'Network Engineer',
       'Administrative Assistant', 'Strategy Consultant', 'Copywriter',
       'Account Manager', 'Director of Marketing', 'Help Desk Analyst',
       'Customer Service Manager', 'Business Intelligence Analyst',
       'VP of Finance', 'Graphic Designer', 'UX Researcher',
       'Social Media Manager', 'Director of Operations',
       'Senior Data Scientist', 'Junior Accountant',
       'Digital Marketing Manager', 'IT Manager',
       'Customer Service Representative', 'Business Development Manager',
       'Senior Financial Analyst', 'Web Developer', 'Research Director',
       'Technical Support Specialist', 'Creative Director',
```

'Senior Software Engineer', 'Human Resources Director',
'Content Marketing Manager', 'Technical Recruiter', 'bbb',
'Chief Technology Officer', 'Junior Designer', 'Financial Advisor',
'Junior Account Manager', 'Senior Project Manager',
'Principal Scientist', 'Sales Associate', 'Supply Chain Manager',
'Senior Marketing Manager', 'Training Specialist',
'Research Scientist', 'Junior Software Developer',
'Public Relations Manager', 'Operations Analyst',
'Event Coordinator', 'Product Marketing Manager',
'Senior HR Manager', 'Junior Web Developer',
'Senior Project Coordinator', 'Digital Content Producer',
'IT Support Specialist', 'Senior Marketing Analyst',
'Customer Success Manager', 'Senior Graphic Designer',
'Supply Chain Analyst', 'Senior Business Analyst',
'Office Manager', 'Junior HR Generalist', 'Senior Product Manager',
'Junior Operations Analyst', 'Senior HR Generalist',
'Sales Operations Manager', 'Senior Software Developer',
'Junior Web Designer', 'Senior Training Specialist',
'Senior Research Scientist', 'Junior Sales Representative',
'Junior Marketing Manager', 'Junior Data Analyst',
'Senior Product Marketing Manager', 'Junior Business Analyst',
'Junior Marketing Specialist', 'Junior Project Manager',
'Senior Accountant', 'Director of Sales', 'Junior Recruiter',
'Senior Business Development Manager', 'Senior Product Designer',
'Junior Customer Support Specialist',
'Senior IT Support Specialist', 'Junior Financial Analyst',
'Senior Operations Manager', 'Director of Human Resources',
'Junior Software Engineer', 'Senior Sales Representative',
'Director of Product Management', 'Junior Copywriter',
'Senior Marketing Coordinator', 'Senior Human Resources Manager',
'Junior Business Development Associate', 'Senior Account Manager',
'Senior Researcher', 'Junior HR Coordinator',
'Director of Finance', 'Junior Marketing Coordinator',
'Junior Data Scientist', 'Senior Operations Analyst',
'Senior Human Resources Coordinator', 'Senior UX Designer',
'Junior Product Manager', 'Senior Marketing Specialist',
'Senior IT Project Manager', 'Senior Quality Assurance Analyst',
'Director of Sales and Marketing', 'Senior Account Executive',
'Director of Business Development', 'Junior Social Media Manager',
'Senior Human Resources Specialist', 'Senior Data Analyst',
'Director of Human Capital', 'Junior Advertising Coordinator',
'Senior Sales Manager', 'Junior UX Designer',
'Senior Marketing Director', 'Senior IT Consultant',
'Senior Financial Advisor', 'Junior Business Operations Analyst',
'Junior Social Media Specialist',
'Senior Product Development Manager', 'Junior Operations Manager',
'Senior Software Architect', 'Junior Research Scientist',
'Junior Marketing Analyst', 'Senior Financial Manager',
'Senior HR Specialist', 'Senior Data Engineer',
'Junior Operations Coordinator', 'Director of HR',
'Senior Operations Coordinator', 'Junior Financial Advisor',
'Director of Engineering', 'Software Engineer Manager',
'Back end Developer', 'Senior Project Engineer',
'Full Stack Engineer', 'Front end Developer', 'Developer',
'Front End Developer', 'Director of Data Science',

```
'Human Resources Coordinator', 'Junior Sales Associate',
'Human Resources Manager', 'Juniour HR Generalist',
'Juniour HR Coordinator', 'Sales Representative',
'Digital Marketing Specialist', 'Receptionist',
'Marketing Director', 'Social M', 'Social Media Man',
'Delivery Driver'], dtype=object)
```

```
1 # Visualizar valores únicos de la columna [5]
2 df['Years of Experience'].unique()
3
4 # --->>> Se observa que se debe de cambiar 'bbb' pues no sería un dato para Años de Experiencia
5
6 array(['5.0', '3.0', '15.0', '7.0', '20.0', '2.0', '12.0', '4.0', '1.0',
       '10.0', 'nan', '18.0', '6.0', '14.0', '16.0', '0.0', '19.0', '9.0',
       '13.0', '11.0', '25.0', '21.0', '8.0', '22.0', 'bbb', '23.0',
       '24.0', '17.0', '1.5', '31.0', '30.0', '28.0', '33.0', '27.0',
       '34.0', '29.0', '26.0', '32.0'], dtype=object)
```

```
1 # Visualizar valores únicos de la columna [6]
2 df['Salary'].unique()
3
```

```
array([ 90000., 65000., 150000., 60000., 200000., nan, 120000.,
       80000., 45000., 110000., 75000., 130000., 40000., 125000.,
       115000., 35000., 180000., 190000., 50000., 140000., 250000.,
       55000., 95000., 105000., 170000., 70000., 160000., 100000.,
       30000., 220000., 135000., 175000., 185000., 85000., 145000.,
       155000., 350., 195000., 198000., 196000., 193000., 92000.,
       165000., 162000., 197000., 142000., 182000., 210000., 550.,
       122485., 169159., 187081., 166109., 78354., 90249., 132720.,
       161568., 127346., 120177., 69032., 101332., 121450., 166375.,
       185119., 166512., 186963., 75072., 163398., 103947., 179180.,
       175966., 190004., 152039., 76742., 191790., 139398., 95845.,
       160976., 126753., 161393., 139817., 181714., 114776., 105725.,
       52731., 106492., 73895., 119836., 99747., 168287., 115920.,
       128078., 51265., 165919., 188651., 55538., 193964., 104702.,
       172955., 138032., 82683., 155414., 154207., 148446., 102859.,
       138662., 181699., 188232., 51832., 188484., 138286., 181132.,
       73938., 119224., 142360., 151315., 181021., 134641., 173851.,
       104127., 178859., 98568., 134858., 94502., 149217., 107895.,
       101186., 62852., 139095., 106278., 90452., 168304., 126593.,
       152203., 183138., 130275., 191915., 62807., 174305., 133326.,
       75656., 155944., 137775., 51831., 182237., 151901., 158254.,
       167207., 112439., 194214., 84407., 139413., 143084., 192344.,
       106132., 184816., 150248., 170995., 88035., 119419., 173582.,
       174436., 71699., 163558., 166828., 144496., 193746., 122581.,
       79767., 177177., 89843., 113563., 128712., 161621., 121454.,
       179987., 72649., 52612., 184006., 131960., 102465., 149748.,
       171036., 146351., 185462., 107718., 90944., 63901., 181902.,
       136533., 136285., 191818., 176643., 70022., 99363., 152944.,
```

```

123386., 168906., 183020., 47898., 135853., 149198., 106662.,
89995., 143814., 174726., 68732., 187951., 137336., 191159.,
102868., 154281., 111535., 107906., 180958., 108607., 178284.,
75969., 197354., 174324., 123781., 141735., 187120., 61095.,
179045., 130355., 103282., 157872., 117314., 186321., 129686.,
68611., 177913., 68472., 113065., 125091., 172925., 126916.,
76898., 579., 103579., 163780., 137878., 92438., 84181.,
174821., 126520., 152168., 190543., 192292., 52807., 174938.,
124071., 73640., 156486., 138859., 52831., 182392., 151078.,
158966., 167924., 113334., 194778., 77606., 140010., 142421.,
192756., 106686., 150729., 171652., 88552., 119918., 174985.,
174336., 72389., 163978., 166958., 145052., 195270., 122970.,
80247., 177862., 114290., 128999., 162454., 122354., 179756.,
73218., 184480., 102828., 150301., 171468., 147326., 185982.,
108267., 91397., 100867., 64182., 182506., 136986., 136662.,
191510., 177347., 70397., 155795., 132638., 178684., 106218.,
191239., 65840., 52779., 185038., 136449., 110707., 151670.,
167015., 146508., 190596., 104378., 70216., 101733., 55935.,
180367., 135596., 136062., 191267., 146075., 131547., 100679.,
186794., 91062., 132442., 82944., 188288., 141090., 152726.,
124141., 67556., 182768., 148727., 91903., 147708., 163209.,
120288., 170226., 134979., 137489., 83577., 117904., 134482.,
184660., 100151., 88678., 181285., 154990., 108204., 175684.,
77766., 192211., 144647., 162231., 121120., 79652., 177002.,
182013., 108799., 135378., 183530., 150901., 82697., 194638.,
130356., 152560., 121432., 63789., 183690., 151310., 100358.,
148437., 168691., 32000., 38000., 89000., 33000., 25000.,
62000., 138000., 47000., 26000., 174000., 41000., 99000.,
117000., 225000., 36000., 146000., 113000., 168000., 122000.,
96000., 49000., 68000., 127000., 71000., 240000., 152000.,
119000., 131000., 101000., 137000., 112000., 91000., 179000.,
74000., 228000., 37000., 204000., 61000., 157000., 52000.,
58000., 219000., 77000., 104000., 183000., 43000., 48000.,
42000., 500., 57000., 72000., 31000., 28000., 215000.,
100052.])

```

b) Contar Datos de cada valor único de las columnas

```

1 # Contar cuantos valores hay de cada valor único de la columna [1]
2 df['Age'].value_counts()
3

```

Age	count
27	541
29	488
30	475
28	462
33	434
26	412
31	388
32	372

```
34          323
36          308
25          290
24          256
35          216
42          184
43          168
37          166
39          162
38          157
45          153
41          141
44          126
46          111
50          104
23          103
48          99
49          98
40          94
54          75
47          49
51          31
52          31
55          18
21          18
22          17
56          12
57          10
53          9
58          8
60          6
62          5
61          2
```

dtype: int64

```
1 # Contar cuantos valores hay de cada valor único de la columna [2]
2 df['Gender'].value_counts()
3
```

```
count
Gender
Male    3897
Female   3211
Other     14
```

dtype: int64

```
1 # Contar cuantos valores hay de cada valor único de la columna [3]
2 df['Education Level'].value_counts()
3
```

	count
Bachelor's Degree	2427
Master's Degree	1683
PhD	1446
Bachelor's	789
High School	470
Master's	304
phD	1

dtype: int64

```
1 # Contar cuantos valores hay de cada valor único de la columna [4]
2 df['Job Title'].value_counts()
3
```

	count
Software Engineer	537
Data Scientist	472
Software Engineer Manager	403
Data Analyst	378
Full Stack Engineer	327
...	...
Senior IT Support Specialist	1
Junior Customer Support Specialist	1
Business Intelligence Analyst	1
Junior Recruiter	1
Office Manager	1

189 rows × 1 columns

dtype: int64

```
1 # Contar cuantos valores hay de cada valor único de la columna [5]
2 df['Years of Experience'].value_counts() # Validar el Tipo de Datos
3
```

Years of Experience	count
3	618
2	612
1	558
4	545
6	471
8	441

5	425
9	399
7	383
11	315
12	303
14	274
16	248
13	220
10	199
bbb	140
15	133
18	133
19	126
0	123
17	111
20	65
22	47
21	45
23	43
25	30
24	21
28	17
32	13
27	12
1.5	12
29	11
30	8
33	6
26	6
31	5
34	2

dtype: int64

```
1 # Contar cuantos valores hay de cada valor único de la columna [6]
2 df['Salary'].value_counts()
3
```

	count
Salary	
140000.0	309
120000.0	295
160000.0	283
55000.0	251
60000.0	240
...	...
106662.0	1
89995.0	1
143814.0	1
174726.0	1
100052.0	1
434 rows × 1 columns	
	dtype: int64

c) Mostrar un resumen estadístico de los datos

Usar df.describe() para obtener estadísticas descriptivas de las columnas numéricas, lo que ayuda a identificar valores atípicos y la distribución general.

```
1 # 2. c) Mostrar un resumen estadístico de los datos:
2
3 #print(df_Trad.describe())
4 df.describe() #sólo se muestran los datos numéricos
5
```

	Age	Salary
count	7122.000000	6971.000000
mean	33.666807	115628.243724
std	7.633494	52565.497372
min	21.000000	350.000000
25%	28.000000	70000.000000
50%	32.000000	115000.000000
75%	38.000000	160000.000000
max	62.000000	250000.000000

Se observa que no se muestra información de todas las columnas, por lo que se tendrá que cambiar los datos a números, para mostrar resumen estadístico completo, por lo que primero se procede con la Traducción de los datos (no incluir acentos en los datos de las columnas)

Traducción y Cambio a Valor Numérico

```
1 # Se genera df para la traducción de los datos
2 df_Trad= df
3 df_Trad.head(5)
4
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Nan	Nan	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

Renombrar columnas

```
1 # Renombrar la columna [1]
2 df_Trad = df_Trad.rename(columns={'Age': 'Edad'})
3 df_Trad.head(5)
4
```

	Edad	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Nan	Nan	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

```
1 # Renombrar la columna [2]
2 df_Trad = df_Trad.rename(columns={'Gender': 'Genero'})
3 df_Trad.head(5)
4
```

	Edad	Genero	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Nan	Nan	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

```
1 # Renombrar la columna [3]
2 df_Trad = df_Trad.rename(columns={'Education Level': 'Nivel_Educativo'})
3 df_Trad.head(5)
4
```

	Edad	Genero	Nivel_Educativo	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Nan	Nan	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

```
1 # Renombrar la columna [4]
2 df_Trad = df_Trad.rename(columns={'Job Title': 'Titulo_Del_Trabajo'})
3 df_Trad.head(5)
4
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Nan	Nan	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

```
1 # Renombrar la columna [5]
2 df_Trad = df_Trad.rename(columns={'Years of Experience': 'Años_De_Experiencia'})
3 df_Trad.head(5)
4
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	NaN	NaN	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

```
1 # Renombrar la columna [6]
2 df_Trad = df_Trad.rename(columns={'Salary': 'Salario'})
3 df_Trad.head(5)
4
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salario
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	NaN	NaN	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

Traducción de Datos

```
1 # Diccionario para traducir el contenido de la columna [2] Genero
2 traduccionGenero = {
3     'Female': 'Mujer',
4     'Male': 'Hombre',
5     'Other': 'Otro'
6 }
7
```

```
1 # Reemplazar (traducir) el contenido de la columna [2] 'Genero'
2 df_Trad['Genero'] = df_Trad['Genero'].replace(traducciónGenero)
3 df_Trad.head(5)
4
```

Edad Genero Nivel_Educativo Titulo_Del_Trabajo Años_De_Experiencia Salario

0	32.0	Hombre	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Mujer	Master's	Data Analyst	3.0	65000.0
2	45.0	Hombre	PhD	Senior Manager	15.0	150000.0
3	36.0	Mujer	NaN	NaN	7.0	60000.0
4	52.0	Hombre	Master's	Director	20.0	200000.0

```
1 # Diccionario para traducir el contenido de la columna [3] Nivel_Educativo
2 traducciónNivelEducativo = {
3     "Bachelor's": 'Licenciatura',
4     "Master's": 'Maestria',
5     "PhD": 'Doctorado',
6     "Master's Degree": 'Titulo de Maestria',
7     "Bachelor's Degree": 'Titulo de Licenciatura',
8     "High School": 'Escuela Secundaria',
9     "phD": 'Doctorado'
10 }
11
```

```
1 # Reemplazar (traducir) el contenido de la columna [3] 'Nivel_Educativo'
2 df_Trad['Nivel_Educativo'] = df_Trad['Nivel_Educativo'].replace(traducciónNivelEducativo)
3 df_Trad.head(5)
4
```

Edad Genero Nivel_Educativo Titulo_Del_Trabajo Años_De_Experiencia Salario

0	32.0	Hombre	Licenciatura	Software Engineer	5.0	90000.0
1	28.0	Mujer	Maestria	Data Analyst	3.0	65000.0
2	45.0	Hombre	Doctorado	Senior Manager	15.0	150000.0
3	36.0	Mujer	NaN	NaN	7.0	60000.0
4	52.0	Hombre	Maestria	Director	20.0	200000.0

```
1 # Diccionario para traducir el contenido de la columna [4] Titulo_Del_Trabajo
2 traducciónTituloTrabajo = {
```

'Software Engineer': 'Ingeniero de Software',
'Data Analyst': 'Analista de Datos',
'Senior Manager': 'Gerente Senior',
'Director': 'Director',
'Product Manager': 'Gerente de Producto',
'Sales Manager': 'Gerente de Ventas',
'Marketing Coordinator': 'Coordinador de Marketing',

'Financial Analyst': 'Analista Financiero',
'Project Manager': 'Gerente de Proyectos',
'Customer Service Rep': 'Representante de Servicio al Cliente',
'Data Entry Clerk': 'Empleado de Ingreso de Datos',
'Business Analyst': 'Analista de Negocios',
'VP of Operations': 'VP de Operaciones',
'IT Support': 'Soporte Tecnico',
'Financial Manager': 'Gerente Financiero',
'Social Media Specialist': 'Especialista en Redes Sociales',
'Junior Developer': 'Desarrollador Junior',
'Product Designer': 'Diseñador de Producto',
'CEO': 'CEO',
'Data Scientist': 'Cientifico de Datos',
'Marketing Specialist': 'Especialista en Marketing',
'Technical Writer': 'Redactor Tecnico',
'HR Generalist': 'Generalista de Recursos Humanos',
'Project Engineer': 'Ingeniero de Proyectos',
'Customer Success Rep': 'Representante de exito del Cliente',
'UX Designer': 'Diseñador UX',
'Operations Director': 'Director de Operaciones',
'Administrative Assistant': 'Asistente Administrativo',
'Strategy Consultant': 'Consultor de Estrategia',
'Copywriter': 'Redactor Publicitario',
'Director of Marketing': 'Director de Marketing',
'Help Desk Analyst': 'Analista de Mesa de Ayuda',
'VP of Finance': 'VP de Finanzas',
'Graphic Designer': 'Diseñador Grafico',
'Senior Engineer': 'Ingeniero Senior',
'Social Media Manager': 'Gerente de Redes Sociales',
'Director of Operations': 'Director de Operaciones',
'Marketing Analyst': 'Analista de Marketing',
'HR Manager': 'Gerente de Recursos Humanos',
'Senior Data Scientist': 'Cientifico de Datos Senior',
'Junior Accountant': 'Contador Junior',
'Digital Marketing Manager': 'Gerente de Marketing Digital',
'Business Development Manager': 'Gerente de Desarrollo de Negocios',
'Web Developer': 'Desarrollador Web',
'Recruiter': 'Reclutador',
'Research Director': 'Director de Investigacion',
'Technical Support Specialist': 'Especialista en Soporte Tecnico',
'Creative Director': 'Director Creativo',
'Operations Manager': 'Gerente de Operaciones',
'Senior Software Engineer': 'Ingeniero de Software Senior',
'Technical Recruiter': 'Reclutador Tecnico',
'bbb': 'bbb',
'Chief Technology Officer': 'Director de Tecnologia',
'Financial Advisor': 'Asesor Financiero',
'Junior Account Manager': 'Gerente de Cuentas Junior',

'Principal Scientist': 'Científico Principal',
'Supply Chain Manager': 'Gerente de Cadena de Suministro',
'Senior Marketing Manager': 'Gerente de Marketing Senior',
'Training Specialist': 'Especialista en Capacitación',
'Junior Software Developer': 'Desarrollador de Software Junior',
'Operations Analyst': 'Analista de Operaciones',
'Event Coordinator': 'Coordinador de Eventos',
'Product Marketing Manager': 'Gerente de Marketing de Producto',
'Senior HR Manager': 'Gerente de Recursos Humanos Senior',
'Junior Web Developer': 'Desarrollador Web Junior',
'Senior Project Coordinator': 'Coordinador de Proyectos Senior',
'Digital Content Producer': 'Productor de Contenido Digital',
'Customer Success Manager': 'Gerente de éxito del Cliente',
'Supply Chain Analyst': 'Analista de Cadena de Suministro',
'Senior Business Analyst': 'Analista de Negocios Senior',
'Senior Financial Analyst': 'Analista Financiero Senior',
'Office Manager': 'Gerente de Oficina',
'Senior Product Manager': 'Gerente de Producto Senior',
'Junior Operations Analyst': 'Analista de Operaciones Junior',
'Customer Service Manager': 'Gerente de Servicio al Cliente',
'Senior Scientist': 'Científico Senior',
'Senior HR Generalist': 'Generalista de Recursos Humanos Senior',
'Junior Web Designer': 'Diseñador Web Junior',
'Senior Training Specialist': 'Especialista en Capacitación Senior',
'Senior Research Scientist': 'Científico Investigador Senior',
'Junior Sales Representative': 'Representante de Ventas Junior',
'Senior Project Manager': 'Gerente de Proyectos Senior',
'Junior Data Analyst': 'Analista de Datos Junior',
'Junior Business Analyst': 'Analista de Negocios Junior',
'Junior Project Manager': 'Gerente de Proyectos Junior',
'Senior Accountant': 'Contador Senior',
'Director of Sales': 'Director de Ventas',
'Senior Business Development Manager': 'Gerente de Desarrollo de Negocios Senior',
'Senior Product Designer': 'Diseñador de Producto Senior',
'Junior Customer Support Specialist': 'Especialista en Soporte al Cliente Junior',
'Senior Marketing Analyst': 'Analista de Marketing Senior',
'Senior IT Support Specialist': 'Especialista en Soporte Técnico Senior',
'Junior Financial Analyst': 'Analista Financiero Junior',
'Senior Operations Manager': 'Gerente de Operaciones Senior',
'Director of Human Resources': 'Director de Recursos Humanos',
'Junior Software Engineer': 'Ingeniero de Software Junior',
'Senior Sales Representative': 'Representante de Ventas Senior',
'Director of Product Management': 'Director de Gestión de Producto',
'Junior Copywriter': 'Redactor Junior',
'Senior Marketing Coordinator': 'Coordinador de Marketing Senior',
'Senior Human Resources Manager': 'Gerente Senior de Recursos Humanos',
'Junior Business Development Associate': 'Asociado de Desarrollo de Negocios Junior',
'Senior Account Manager': 'Gerente de Cuentas Senior',

'Senior Researcher': 'Investigador Senior',
'Junior HR Coordinator': 'Coordinador de Recursos Humanos Junior',
'Director of Finance': 'Director de Finanzas',
'Junior Data Scientist': 'Científico de Datos Junior',
'Senior Operations Analyst': 'Analista de Operaciones Senior',
'Senior Human Resources Coordinator': 'Coordinador de Recursos Humanos Senior',
'Senior UX Designer': 'Diseñador UX Senior',
'Junior Product Manager': 'Gerente de Producto Junior',
'Senior Marketing Specialist': 'Especialista en Marketing Senior',
'Senior IT Project Manager': 'Gerente de Proyectos de TI Senior',
'Senior Quality Assurance Analyst': 'Analista de Aseguramiento de Calidad Senior',
'Senior Account Executive': 'Ejecutivo de Cuentas Senior',
'Director of Business Development': 'Director de Desarrollo de Negocios',
'Junior Social Media Manager': 'Gerente de Redes Sociales Junior',
'Senior Human Resources Specialist': 'Especialista en Recursos Humanos Senior',
'Senior Data Analyst': 'Analista de Datos Senior',
'Director of Human Capital': 'Director de Capital Humano',
'Junior Advertising Coordinator': 'Coordinador de Publicidad Junior',
'Junior UX Designer': 'Diseñador UX Junior',
'Senior Marketing Director': 'Director de Marketing Senior',
'Junior HR Generalist': 'Generalista de Recursos Humanos Junior',
'Junior Marketing Coordinator': 'Coordinador de Marketing Junior',
'Senior Financial Advisor': 'Asesor Financiero Senior',
'Junior Business Operations Analyst': 'Analista de Operaciones de Negocios Junior',
'Junior Social Media Specialist': 'Especialista en Redes Sociales Junior',
'Junior Operations Manager': 'Gerente de Operaciones Junior',
'Senior Software Architect': 'Arquitecto de Software Senior',
'Junior Marketing Specialist': 'Especialista en Marketing Junior',
'Senior Software Developer': 'Desarrollador de Software Senior',
'Junior Marketing Analyst': 'Analista de Marketing Junior',
'Senior IT Consultant': 'Consultor de TI Senior',
'Senior Financial Manager': 'Gerente Financiero Senior',
'Junior Marketing Manager': 'Gerente de Marketing Junior',
'Junior Operations Coordinator': 'Coordinador de Operaciones Junior',
'Director of HR': 'Director de Recursos Humanos',
'Senior Operations Coordinator': 'Coordinador de Operaciones Senior',
'Senior Data Engineer': 'Ingeniero de Datos Senior',
'Junior Financial Advisor': 'Asesor Financiero Junior',
'Director of Engineering': 'Director de Ingeniería',
'Senior Project Engineer': 'Ingeniero de Proyectos Senior',
'Full Stack Engineer': 'Ingeniero Full Stack',
'Front end Developer': 'Desarrollador Front End',
'Back end Developer': 'Desarrollador Back End',
'Software Engineer Manager': 'Gerente de Ingenieros de Software',
'Front End Developer': 'Desarrollador Front End',
'Software Developer': 'Desarrollador de Software',
'Director of Data Science': 'Director de Ciencia de Datos',
'Marketing Manager': 'Gerente de Marketing',

```
'Human Resources Coordinator': 'Coordinador de Recursos Humanos',
'Junior Sales Associate': 'Asociado de Ventas Junior',
'Human Resources Manager': 'Gerente de Recursos Humanos',
'Junior HR Generalist': 'Generalista de Recursos Humanos Junior',
'Junior HR Coordinator': 'Coordinador de Recursos Humanos Junior',
'Senior Product Marketing Manager': 'Gerente Senior de Marketing de Producto',
'Sales Associate': 'Asociado de Ventas',
'Content Marketing Manager': 'Gerente de Marketing de Contenidos',
'Sales Director': 'Director de Ventas',
'Sales Representative': 'Representante de Ventas',
'Research Scientist': 'Científico Investigador',
'Digital Marketing Specialist': 'Especialista en Marketing Digital',
'Receptionist': 'Recepcionista',
'Marketing Director': 'Director de Marketing',
'Social Media Man': 'Gerente de Redes Sociales',
'Customer Service Representative': 'Representante de Servicio al Cliente',
'Delivery Driver': 'Conductor de Entrega',
'Sales Executive': 'Ejecutivo de Ventas',
'Junior Research Scientist': 'Científico Investigador Junior',
'Sales Operations Manager': 'Gerente de Operaciones de Ventas'
```

{

```
1 # Reemplazar (traducir) el contenido de la columna [4] 'Titulo_Del_Trabajo'
2 df_Trad['Titulo_Del_Trabajo'] = df_Trad['Titulo_Del_Trabajo'].replace(traduccionTituloTrabajo)
3 df_Trad.head(5)
4
```

The screenshot shows a Jupyter Notebook interface. On the left, there is a code cell containing Python code for replacing values in a column of a DataFrame. On the right, there is a data preview cell showing a portion of a DataFrame with columns: Edad, Genero, Nivel_Educativo, Titulo_Del_Trabajo, Años_De_Experiencia, and Salario. The data includes rows for various employees with their respective details.

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salario
0	32.0	Hombre	Licenciatura	Ingeniero de Software	5.0	90000.0
1	28.0	Mujer	Maestria	Analista de Datos	3.0	65000.0
2	45.0	Hombre	Doctorado	Gerente Senior	15.0	150000.0
3	36.0	Mujer	Nan	Nan	7.0	60000.0
4	52.0	Hombre	Maestria	Director	20.0	200000.0

Convertir las columnas a valores numéricos

```
1 # Se genera df para mostrar resumen estadístico
2 dfNum = df_Trad
3 dfNum.head(5)
4
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salario
0	32.0	Hombre	Licenciatura	Ingeniero de Software	5.0	90000.0
1	28.0	Mujer	Maestria	Analista de Datos	3.0	65000.0
2	45.0	Hombre	Doctorado	Gerente Senior	15.0	150000.0
3	36.0	Mujer	NaN	NaN	7.0	60000.0
4	52.0	Hombre	Maestria	Director	20.0	200000.0

```
1 # Convertir la columna [2] 'Genero' a valores numéricos
2 dfNum['Genero'] = dfNum['Genero'].map({
3     'Hombre': 1,
4     'Mujer': 2,
5     'Otro': 3
6 })
7 dfNum.head(5)
8
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salario
0	32.0	1.0	Licenciatura	Ingeniero de Software	5.0	90000.0
1	28.0	2.0	Maestria	Analista de Datos	3.0	65000.0
2	45.0	1.0	Doctorado	Gerente Senior	15.0	150000.0
3	36.0	2.0	NaN	NaN	7.0	60000.0
4	52.0	1.0	Maestria	Director	20.0	200000.0

```
1 # Convertir la columna [3] 'Nivel_Educativo' a valores numéricos
2 dfNum['Nivel_Educativo'] = dfNum['Nivel_Educativo'].map({
3     'Licenciatura': 1,
4     'Maestria': 2,
5     'Doctorado': 3,
6     'Titulo de Maestria': 4,
7     'Titulo de Licenciatura': 5,
8     'Escuela Secundaria': 6,
9     'Doctorado': 7
10 })
11 dfNum.head(5)
12
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salario
0	32.0	1.0	1.0	Ingeniero de Software	5.0	90000.0
1	28.0	2.0	2.0	Analista de Datos	3.0	65000.0
2	45.0	1.0	7.0	Gerente Senior	15.0	150000.0
3	36.0	2.0	NaN	NaN	7.0	60000.0
4	52.0	1.0	2.0	Director	20.0	200000.0

```
# Convertir la columna [4] 'Titulo_Del_Trabajo' a valores numéricos
dfNum['Titulo Del Trabajo'] = dfNum['Titulo Del Trabajo'].map({
    'Ingeniero de Software': 1,
    'Analista de Datos': 2,
    'Gerente Senior': 3,
    'Director': 4,
    'Gerente de Producto': 5,
    'Gerente de Ventas': 6,
    'Coordinador de Marketing': 7,
    'Analista Financiero': 8,
    'Gerente de Proyectos': 9,
    'Representante de Servicio al Cliente': 10,
    'Empleado de Ingreso de Datos': 11,
    'Analista de Negocios': 12,
    'VP de Operaciones': 13,
    'Soporte Tecnico': 14,
    'Gerente Financiero': 15,
    'Especialista en Redes Sociales': 16,
    'Desarrollador Junior': 17,
    'Diseñador de Producto': 18,
    'CEO': 19,
    'Cientifico de Datos': 20,
    'Especialista en Marketing': 21,
    'Redactor Tecnico': 22,
    'Generalista de Recursos Humanos': 23,
    'Ingeniero de Proyectos': 24,
    'Representante de exito del Cliente': 25,
    'Diseñador UX': 26,
    'Director de Operaciones': 27,
    'Asistente Administrativo': 28,
    'Consultor de Estrategia': 29,
    'Redactor Publicitario': 30,
    'Director de Marketing': 31,
    'Analista de Mesa de Ayuda': 32,
    'VP de Finanzas': 33,
    'Diseñador Grafico': 34,
    'Ingeniero Senior': 35,
    'Gerente de Redes Sociales': 36,
    'Director de Operaciones': 37,
    'Analista de Marketing': 38,
    'Gerente de Recursos Humanos': 39,
    'Cientifico de Datos Senior': 40,
    'Contador Junior': 41,
    'Gerente de Marketing Digital': 42,
    'Gerente de Desarrollo de Negocios': 43,
    'Desarrollador Web': 44,
    'Reclutador': 45,
    'Director de Investigacion': 46,
```

'Especialista en Soporte Tecnico': 47,
'Director Creativo': 48,
'Gerente de Operaciones': 49,
'Ingeniero de Software Senior': 50,
'Reclutador Tecnico': 51,
'bbb': 52,
'Director de Tecnologia': 53,
'Asesor Financiero': 54,
'Gerente de Cuentas Junior': 55,
'Cientifico Principal': 56,
'Gerente de Cadena de Suministro': 57,
'Gerente de Marketing Senior': 58,
'Especialista en Capacitacion': 59,
'Desarrollador de Software Junior': 60,
'Analista de Operaciones': 61,
'Coordinador de Eventos': 62,
'Gerente de Marketing de Producto': 63,
'Gerente de Recursos Humanos Senior': 64,
'Desarrollador Web Junior': 65,
'Coordinador de Proyectos Senior': 66,
'Productor de Contenido Digital': 67,
'Gerente de exito del Cliente': 68,
'Analista de Cadena de Suministro': 69,
'Analista de Negocios Senior': 70,
'Analista Financiero Senior': 71,
'Gerente de Oficina': 72,
'Gerente de Producto Senior': 73,
'Analista de Operaciones Junior': 74,
'Gerente de Servicio al Cliente': 75,
'Cientifico Senior': 76,
'Generalista de Recursos Humanos Senior': 77,
'Diseñador Web Junior': 78,
'Especialista en Capacitacion Senior': 79,
'Cientifico Investigador Senior': 80,
'Representante de Ventas Junior': 81,
'Gerente de Proyectos Senior': 82,
'Analista de Datos Junior': 83,
'Analista de Negocios Junior': 84,
'Gerente de Proyectos Junior': 85,
'Contador Senior': 86,
'Director de Ventas': 87,
'Gerente de Desarrollo de Negocios Senior': 88,
'Diseñador de Producto Senior': 89,
'Especialista en Soporte al Cliente Junior': 90,
'Analista de Marketing Senior': 91,
'Especialista en Soporte Tecnico Senior': 92,
'Analista Financiero Junior': 93,
'Gerente de Operaciones Senior': 94,

'Director de Recursos Humanos': 95,
'Ingeniero de Software Junior': 96,
'Representante de Ventas Senior': 97,
'Director de Gestión de Producto': 98,
'Redactor Junior': 99,
'Coordinador de Marketing Senior': 100,
'Gerente Senior de Recursos Humanos': 101,
'Asociado de Desarrollo de Negocios Junior': 102,
'Gerente de Cuentas Senior': 103,
'Investigador Senior': 104,
'Coordinador de Recursos Humanos Junior': 105,
'Director de Finanzas': 106,
'Científico de Datos Junior': 107,
'Analista de Operaciones Senior': 108,
'Coordinador de Recursos Humanos Senior': 109,
'Diseñador UX Senior': 110,
'Gerente de Producto Junior': 111,
'Especialista en Marketing Senior': 112,
'Gerente de Proyectos de TI Senior': 113,
'Analista de Aseguramiento de Calidad Senior': 114,
'Ejecutivo de Cuentas Senior': 115,
'Director de Desarrollo de Negocios': 116,
'Gerente de Redes Sociales Junior': 117,
'Especialista en Recursos Humanos Senior': 118,
'Analista de Datos Senior': 119,
'Director de Capital Humano': 120,
'Coordinador de Publicidad Junior': 121,
'Diseñador UX Junior': 122,
'Director de Marketing Senior': 123,
'Generalista de Recursos Humanos Junior': 124,
'Coordinador de Marketing Junior': 125,
'Asesor Financiero Senior': 126,
'Analista de Operaciones de Negocios Junior': 127,
'Especialista en Redes Sociales Junior': 128,
'Gerente de Operaciones Junior': 129,
'Arquitecto de Software Senior': 130,
'Especialista en Marketing Junior': 131,
'Desarrollador de Software Senior': 132,
'Analista de Marketing Junior': 133,
'Consultor de TI Senior': 134,
'Gerente Financiero Senior': 135,
'Gerente de Marketing Junior': 136,
'Coordinador de Operaciones Junior': 137,
'Director de Recursos Humanos': 138,
'Coordinador de Operaciones Senior': 139,
'Ingeniero de Datos Senior': 140,
'Asesor Financiero Junior': 141,
'Director de Ingeniería': 142,

```

'Iingeniero de Proyectos Senior': 143,
'Iingeniero Full Stack': 144,
'Desarrollador Front End': 145,
'Desarrollador Back End': 146,
'Gerente de Ingenieros de Software': 147,
'Desarrollador Front End': 148,
'Desarrollador de Software': 149,
'Director de Ciencia de Datos': 150,
'Gerente de Marketing': 151,
'Coordinador de Recursos Humanos': 152,
'Asociado de Ventas Junior': 153,
'Gerente de Recursos Humanos': 154,
'Generalista de Recursos Humanos Junior': 155,
'Coordinador de Recursos Humanos Junior': 156,
'Gerente Senior de Marketing de Producto': 157,
'Asociado de Ventas': 158,
'Gerente de Marketing de Contenidos': 159,
'Director de Ventas': 160,
'Representante de Ventas': 161,
'Científico Investigador': 162,
'Especialista en Marketing Digital': 163,
'Recepcionista': 164,
'Director de Marketing': 165,
'Gerente de Redes Sociales': 166,
'Representante de Servicio al Cliente': 167,
'Conductor de Entrega': 168,
'Ejecutivo de Ventas': 169,
'Científico Investigador Junior': 170,
'Gerente de Operaciones de Ventas': 171
})

```

```
dfNum.head(5)
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Años_De_Experiencia	Salario
0	32.0	1.0	1.0	1.0	5.0	90000.0
1	28.0	2.0	2.0	2.0	3.0	65000.0
2	45.0	1.0	7.0	3.0	15.0	150000.0
3	36.0	2.0	NaN	NaN	7.0	60000.0
4	52.0	1.0	2.0	4.0	20.0	200000.0

Mostrar un resumen estadístico de los datos (Final)

```
1 # Mostrar un resumen estadístico de los datos:  
2 #print(df_Trad.describe())  
3 df_Trad.describe()  
4
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Salario
count	7122.000000	7122.000000	7120.000000	7103.000000	6971.000000
mean	33.6666807	1.454788	4.664747	85.74011	115628.243724
std	7.633494	0.501919	1.777825	64.21260	52565.497372
min	21.000000	1.000000	1.000000	1.00000	350.000000
25%	28.000000	1.000000	4.000000	20.00000	70000.000000
50%	32.000000	1.000000	5.000000	83.00000	115000.000000
75%	38.000000	2.000000	6.000000	148.00000	160000.000000
max	62.000000	3.000000	7.000000	171.00000	250000.000000

```
1 # Disminuyendo el número de decimales  
2 df_Trad.describe().style.format(precision=2)  
3  
4 # De aquí podemos ver por ejemplo que:  
5 # El 75% de los datos son Mujeres de 38 años  
6 # El 50% de los datos son Hombres de 32 años  
7 # El 25% de los datos son Hombres de 28 años  
8
```

	Edad	Genero	Nivel_Educativo	Titulo_Del_Trabajo	Salario
count	7122.00	7122.00	7120.00	7103.00	6971.00
mean	33.67	1.45	4.66	85.74	115628.24
std	7.63	0.50	1.78	64.21	52565.50
min	21.00	1.00	1.00	1.00	350.00
25%	28.00	1.00	4.00	20.00	70000.00
50%	32.00	1.00	5.00	83.00	115000.00
75%	38.00	2.00	6.00	148.00	160000.00
max	62.00	3.00	7.00	171.00	250000.00

d) Calcular el porcentaje de valores faltantes por columna:

Utilizar df.isnull().mean() * 100 para calcular el porcentaje de valores faltantes en cada columna, lo que ayuda a determinar la severidad de los problemas de datos.

```
1 # d) Calcular el porcentaje de valores faltantes por columna:  
2  
3 # Porcentaje de valores faltantes  
4 ...  
5 Edad           4.029107  
6 Genero         4.029107  
7 Nivel_Educativo 4.056057  
8 Titulo_Del_Trabajo 4.285137  
9 Años_De_Experiencia 4.056057  
10 Salario        6.063873  
11 dtype: float64  
12 ...  
13 missing_percentage = df_Trad.isnull().mean() * 100  
14 #print(missing_percentage)  
15 missing_percentage  
16
```

0

Edad	4.029107
Genero	4.029107
Nivel_Educativo	4.056057
Titulo_Del_Trabajo	4.042582
Años_De_Experiencia	4.056057
Salario	6.063873

dtype: float64

e) Identificar si hay filas duplicadas:

Usar df.duplicated().sum() para contar el número de filas duplicadas, lo que es crucial para asegurar que cada observación sea única.

```
1 # e) Identificar si hay filas duplicadas:  
2  
3 # Total de filas duplicadas  
4 before_total_duplicates = df_Trad.duplicated().sum()  
5 print(f'Total de filas duplicadas inicial: {before_total_duplicates}')  
6 # 4,223
```

0

Total de filas duplicadas inicial: 4223

f) Analizar los tipos de datos de las columnas:

Utilizar df.dtypes para verificar que cada columna tenga el tipo de dato esperado, lo cual es importante para evitar errores en los análisis posteriores.

```
1 # f) Analizar los tipos de datos de las columnas:  
2  
3 # Tipos de datos  
4 '''  
5 Age           float64  
6 Gender        object  
7 Education Level object  
8 Job Title    object  
9 Years of Experience object  
10 Salary        float64  
11 dtype: object  
12 '''  
13 print(df.dtypes)  
14 #df_Trad.dtypes  
15  
16 # --->>> Edad debe de ser int  
17 # --->>> Años_De_Experiencia de ser int  
18
```



```
Age           float64  
Gender        object  
Education Level object  
Job Title    object  
Years of Experience object  
Salary        float64  
dtype: object
```

3. Limpieza de Datos

```
1 # Se genera df para limpieza de datos  
2 df_Clean = df  
3 df_Clean.shape # 7,421  
4
```



```
(7421, 6)
```

Realizar las siguientes tareas de limpieza:

a) Eliminación o imputación de valores faltantes:

```
1 # a) Eliminación o imputación de valores faltantes:  
2  
3 # Identificar valores faltantes  
4 missing_values_before = df_Clean.isnull().sum()  
5 #print(missing_values_before)  
6 missing_values_before  
7
```

0	
Edad	299
Genero	299
Nivel_Educativo	301
Titulo_Del_Trabajo	318
Años_De_Experiencia	301
Salario	450

```
1 # Se genera df para cambiar los datos nulos  
2 df_DropNa = df_Trad  
3
```

```
1 df_DropNa.shape # No se elimina ningún dato  
2 # 7,421  
3
```

(7421, 6)

```
1 # Se crea una lista de todos los nombres de las columnas  
2 lista_col = df_DropNa.columns  
3 lista_col  
4
```

```
Index(['Edad', 'Genero', 'Nivel_Educativo', 'Titulo_Del_Trabajo',  
       'Años_De_Experiencia', 'Salario'],  
      dtype='object')
```

```
1 # Hay que validar que NO debe de tener NaN  
2  
3 for nom_colum in lista_col:  
4     df_DropNa= df_DropNa.dropna(subset=[nom_colum])  
5
```

```
1 df_DropNa.shape # No se elimina ningún dato  
2 # 5,688  
3
```

```
4 (5688, 6)
```

```
1 # Inicialmente se tenían 7,421 registros y al utilizar el dropna quedarían 5,688 por lo que perderíamos 1,733  
2  
3 # --->>> Se recomienda No utilizar dropna  
4  
5 # De acuerdo al Análisis, se va a reemplazar o llenar los datos NaN para después decidir si se utilizan  
6 # se requiere hacer el cambio de los NaN, para la modificación de los Tipos de Datos de las columnas.  
7  
8 df_Clean.shape # Por lo que seguiremos trabajando el df de la Limpieza en donde No se elimina ningún dato  
9
```

```
10 (7421, 6)
```

b) Calcular el porcentaje de valores faltantes por columna

Porcentaje de valores faltantes (Inicial)

```
1 # Porcentaje de valores faltantes por columna  
2  
3 ...  
4 Edad 4.029107  
5 Genero 4.029107  
6 Nivel_Educativo 4.056057  
7 Titulo_Del_Trabajo 4.042582  
8 Años_De_Experiencia 4.056057  
9 Salario 6.063873  
10 dtype: float64  
11 ...  
12 #Porcentaje de valores faltantes por columna:  
13 missing_percentage_before = df_Clean.isnull().mean() * 100  
14 #print(missing_percentage_before)  
15 missing_percentage_before  
16
```

```
0  
Edad 4.029107  
Genero 4.029107  
Nivel_Educativo 4.056057  
Titulo_Del_Trabajo 4.042582  
Años_De_Experiencia 4.056057  
Salario 6.063873
```

```
dtype: float64
```

```
▶ 1 df_Clean.info()
 2 # El número de registros de cada columna NO es igual al número de renglones 7,421 dado que tiene NaN
 3
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 7421 entries, 0 to 7420
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   Edad              7122 non-null    float64
 1   Genero             7122 non-null    object  
 2   Nivel_Educativo    7120 non-null    object  
 3   Titulo_Del_Trabajo 7121 non-null    object  
 4   Años_De_Experiencia 7120 non-null    object  
 5   Salario            6971 non-null    float64
dtypes: float64(2), object(4)
memory usage: 348.0+ KB
```

```
▶ 1 # se crea una lista de todos los nombres de las columnas
 2 lista_col_dNa = df_Clean.columns
 3 lista_col_dNa
→ Index(['Age', 'Gender', 'Education Level', 'Job Title', 'Years of Experience',
       'Salary'],
       dtype='object')
```

```
▶ 1 # Verificando si hay invalid_value
 2 for i in lista_col_dNa:
 3     print(f"En la columna {i} los invalid_value son: {df_Clean[df_Clean[i] == 'invalid_value'].shape[0]}")
 4
→ En la columna Age los invalid_value son: 0
En la columna Gender los invalid_value son: 0
En la columna Education Level los invalid_value son: 0
En la columna Job Title los invalid_value son: 0
En la columna Years of Experience los invalid_value son: 0
En la columna Salary los invalid_value son: 0
```

Reemplazar *NaN* por valores validos

```
▶ 1 df_Clean.shape
 2 # 7,421
 3
→ (7421, 6)
```

```
▶ 1 df_Clean.columns
 2
→ Index(['Edad', 'Genero', 'Nivel_Educativo', 'Titulo_Del_Trabajo',
       'Años_De_Experiencia', 'Salario'],
       dtype='object')
```

```
1 # Reemplazar NaN por valores validos --> 0 (cero)
2 df_Clean['Edad'].fillna(0, inplace=True) # --->>> No tenemos registros con Edad 0 años (NO es valor de la columna)
3
```

```
1 # Reemplazar NaN por valores validos --> 'Other'
2 df_Clean['Genero'].fillna("Otro", inplace=True) # --->>> Si tenemos más registros con Genero= Otro (si es valor de la columna)
3
```

<ipython-input-63-f4f7079d867c>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform

```
df_Clean['Genero'].fillna("Otro", inplace=True) # --->>> Si tenemos más registros con Genero= Otro (si es valor de la columna)
<ipython-input-63-f4f7079d867c>:2: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value 'Otr
df_Clean['Genero'].fillna("Otro", inplace=True) # --->>> Si tenemos más registros con Genero= otro (si es valor de la columna)
```

```
1 # Reemplazar NaN por valores validos --> 'Other'
2 df_Clean['Nivel_Educativo'].fillna("Otro", inplace=True) # --->>> Podría no ser indispensable tener el Nivel Educativo
3
```

<ipython-input-64-1c5d8e8f642e>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform

```
df_Clean['Nivel_Educativo'].fillna("Otro", inplace=True) # --->>> Podría no ser indispensable tener el Nivel Educativo
<ipython-input-64-1c5d8e8f642e>:2: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value 'Otr
df_Clean['Nivel_Educativo'].fillna("Otro", inplace=True) # --->>> Podría no ser indispensable tener el Nivel Educativo
```

```
1 # Reemplazar NaN por valores validos --> 'Other'
2 df_Clean['Titulo_Del_Trabajo'].fillna("Otro", inplace=True) # --->>> Podría no ser indispensable tener el Título del Trabajo
3
```

<ipython-input-65-57dac0427d3b>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform

```
df_Clean['Titulo_Del_Trabajo'].fillna("Otro", inplace=True) # --->>> Podría no ser indispensable tener el Título del Trabajo
<ipython-input-65-57dac0427d3b>:2: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value 'Otr
df_Clean['Titulo_Del_Trabajo'].fillna("Otro", inplace=True) # --->>> Podría no ser indispensable tener el Título del Trabajo
```

```
1 # Reemplazar NaN por valores validos --> 0 (cero)
2 df_Clean['Años_De_Experiencia'].fillna(0, inplace=True) # --->>> Si tenemos más registros con Años de Experiencia = 0 (si es valor de la columna)
3
```

<ipython-input-66-e1033a4b0257>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform

```
df_Clean['Años_De_Experiencia'].fillna(0, inplace=True) # --->>> Si tenemos más registros con Años de Experiencia = 0 (si es valor de la columna)
```

```
1 # Reemplazar NaN por valores validos --> 0 (cero)
2 df_Clean['Salario'].fillna(0, inplace=True) # --->>> No tenemos registros con Edad 0 años (NO es valor de la columna)
3
```

Validar Porcentaje de valores faltantes

```
1 # Porcentaje de valores faltantes por columna:  
2  
3 ...  
4 Edad 4.029107 -----> NO Cambia a 0 # --->> No tenemos registros con Edad 0 años (NO es valor de la columna)  
5 Genero 4.029107 -----> Cambia a 0  
6 Nivel_Educativo 4.056057 -----> Cambia a 0  
7 Titulo_Del_Trabajo 4.042582 -----> Cambia a 0  
8 Años_De_Experiencia 4.056057 -----> Cambia a 0  
9 Salario 6.063873 -----> NO Cambia a 0 # --->> No tenemos registros con Edad 0 años (NO es valor de la columna)|  
10 dtype: float64  
11 ...  
12 # Porcentaje de valores faltantes  
13 missing_percentage_after = df_clean.isnull().mean() * 100  
14 #print(missing_percentage_after)  
15 missing_percentage_after  
16
```

```
Edad 4.029107  
Genero 0.000000  
Nivel_Educativo 0.000000  
Titulo_Del_Trabajo 0.000000  
Años_De_Experiencia 0.000000  
Salario 6.063873  
  
dtype: float64
```

Eliminar NaN

```
1 df_Clean.shape  
2 # 7,421
```

```
1 # Se genera df para Cambiar los datos nulos  
2 df_DropNa = df_Clean  
3
```

```
1 # Se crea una lista de todos los nombres de las columnas  
2 #lista_col = df_DropNa.columns  
3 lista_col_dNa = ['Edad', 'Salario']  
4 lista_col_dNa  
5
```

```
1 df_DropNa.shape  
2 # 6,693  
3
```

```
(6693, 6)
```

Porcentaje de valores faltantes (Final)

```
1 # Porcentaje de valores faltantes por columna:  
2  
3 ...          Original  
4 Edad           4.029107  --->>> Cambia a 0  # --->>> Se borran con dropna  
5 Genero         0.000000  
6 Nivel_Educativo 0.000000  
7 Titulo_Del_Trabajo 0.000000  
8 Años_De_Experiencia 0.000000  
9 Salario        6.063873  --->>> Cambia a 0  # --->>> Se borran con dropna  
10 dtype: float64  
11 ...  
12 # Porcentaje de valores faltantes  
13 missing_percentage_after = df_DropNa.isnull().mean() * 100  
14 #print(missing_percentage_after)  
15 missing_percentage_after  
16
```

```
0  
Edad      0.0  
Genero    0.0  
Nivel_Educativo 0.0  
Titulo_Del_Trabajo 0.0  
Años_De_Experiencia 0.0  
Salario    0.0  
  
dtype: float64
```

```
1 df_DropNa.info()  
2  
3 # Como ya se reemplazaron todos los datos NaN, se puede observar que  
4 # el número de registros es igual al número de renglones 7,421  
5
```

```
<class 'pandas.core.frame.DataFrame'>  
Index: 6693 entries, 0 to 7420  
Data columns (total 6 columns):  
 #   Column            Non-Null Count  Dtype     
---  --  
 0   Edad              6693 non-null    float64  
 1   Genero            6693 non-null    object  
 2   Nivel_Educativo   6693 non-null    object  
 3   Titulo_Del_Trabajo 6693 non-null    object  
 4   Años_De_Experiencia 6693 non-null    object  
 5   Salario           6693 non-null    float64  
dtypes: float64(2), object(4)  
memory usage: 366.0+ KB
```

c) Eliminación de duplicados:

Identificar y eliminar filas duplicadas usando df.drop_duplicates(), garantizando que los datos sean únicos.

```
1 # Se crea df después de Reemplazar NaN
2 df_Dupli = df_DropNa
3 df_Dupli.shape
4
```

→ (6693, 6)

```
1 #Retomando la lista de todos los nombres de las columnas
2 lista_col = df_Dupli.columns
3 lista_col
4
```

→ Index(['Age', 'Gender', 'Education Level', 'Job Title', 'Years of Experience',
 'Salary'],
 dtype='object')

```
1 # c) Eliminación de duplicados:
2
3 duplicates_before = df_Dupli.duplicated().sum()
4 df_Dupli.drop_duplicates(inplace=True) # Eliminar filas duplicadas
5 duplicates_after = df_Dupli.duplicated().sum()
6 print(f"Número de Duplicados antes: {duplicates_before}, Número de Duplicados después: {duplicates_after}")
7
```

→ Número de Duplicados antes: 4092, Número de Duplicados después: 0

```
1 # Validar el número de registros después de Eliminar duplicados
2 df_Dupli.shape # 7,421 - 4,092 = 2,601
3
```

→ (2601, 6)

```
[1] 1 #validar número de registros
2 df_Dupli.info() # el número de registros es igual al número de renglones
3
```

→ <class 'pandas.core.frame.DataFrame'>
Index: 2601 entries, 0 to 7415
Data columns (total 6 columns):
 # Column Non-Null Count Dtype

 0 Edad 2601 non-null float64
 1 Genero 2601 non-null object
 2 Nivel_Educativo 2601 non-null object
 3 Titulo_Del_Trabajo 2601 non-null object
 4 Años_De_Experiencia 2601 non-null object
 5 Salario 2601 non-null float64
dtypes: float64(2), object(4)
memory usage: 142.2+ KB

d) Corrección de tipos de datos:

Asegurarse de que las columnas tengan tipos de datos adecuados, utilizando astype() para convertir tipos incorrectos, por ejemplo, asegurando que la columna de edad sea un entero.

```
1 # d) Corrección de tipos de datos:  
2  
3 # Verificar tipos de datos  
4 print(df_Dupli.dtypes)  
5
```

```
→ Edad          float64  
Genero         object  
Nivel_Educativo    object  
Titulo_Del_Trabajo    object  
Años_De_Experiencia    object  
Salario        float64  
dtype: object
```

```
[ ] 1 #validar número de registros  
2 df_Dupli.info() # el número de registros es igual al número de renglones  
3
```

```
→ <class 'pandas.core.frame.DataFrame'>  
Index: 2601 entries, 0 to 7415  
Data columns (total 6 columns):  
 #   Column      Non-Null Count  Dtype     
---  
 0   Edad        2601 non-null   float64  
 1   Genero       2601 non-null   object  
 2   Nivel_Educativo  2601 non-null   object  
 3   Titulo_Del_Trabajo  2601 non-null   object  
 4   Años_De_Experiencia  2601 non-null   object  
 5   Salario      2601 non-null   float64  
dtypes: float64(2), object(4)  
memory usage: 142.2+ KB
```

```
1 # Convertir tipos de datos  
2 # Asegurar que la columna 'Edad' tenga tipo de datos adecuado.  
3 df_Dupli['Edad'] = df_Dupli['Edad'].astype(int) # Asegurar que 'Edad' sea int  
4
```

```
1 # Convertir tipos de datos  
2 # Asegurar que las columnas 'Salario' tenga tipo de datos adecuado.  
3 df_Dupli['Salario'] = df_Dupli['Salario'].astype(float) # Asegurar que 'Salario' sea float  
4  
5 # ---->>> La columna de 'Salario' tiene la cadena 'bbb' por lo que se tiene que corregir los valores inválidos  
6
```

```
1 # Convertir tipos de datos  
2 # Asegurar que la columna 'Años_De_Experiencia' tenga tipo de datos adecuado.  
3 df_Dupli['Años_De_Experiencia'] = df_Dupli['Años_De_Experiencia'].astype(float)  
4 # Asegurar que 'Años_De_Experiencia' sea int antes verifiquemos que sea float  
5  
6 # ---->>> La columna de 'Años_De_Experiencia' tiene la cadena 'bbb'  
7 # por lo que se tiene que corregir los valores inválidos  
8
```

```
ValueError                                Traceback (most recent call last)  
<ipython-input-90-eb1b878130b2> in <cell line: 3>()  
  1 # Convertir tipos de datos  
  2 # Asegurar que la columna 'Años_De_Experiencia' tenga tipo de datos adecuado.  
----> 3 df_Dupli['Años_De_Experiencia'] = df_Dupli['Años_De_Experiencia'].astype(float)  
  4 # Asegurar que 'Años_De_Experiencia' sea int antes verifiquemos que sea float  
  5  
  6 # ---->>> La columna de 'Años_De_Experiencia' tiene la cadena 'bbb'  
  7 # por lo que se tiene que corregir los valores inválidos  
  8  
-----  
ValueError: could not convert string to float: 'bbb'
```

e) Corrección de valores "inválidos":

Identificar y corregir valores incorrectos, como cadenas erróneas ('bbb'), reemplazándolos con un valor adecuado o eliminándolos.

```
▶ 1 # Se genera df para Cambiar los datos nulos
 2 df_Final = df_Dupli
 3
```

```
▶ 1 df_Final.shape
 2 # 2,601
```

```
→ (2601, 6)
```

```
▶ 1 # e) Corrección de valores "inválidos":
 2
 3 # Reemplazar valores inválidos
 4 # df['Years of Experience'].replace('bbb', pd.NA, inplace=True) # Reemplazar 'bbb' con NaN
 5
 6 df_Final['Años_De_Experiencia'].replace('bbb', pd.NA, inplace=True) # Reemplazar 'bbb' con NaN
 7
```

```
▶ 1 # Reemplazar NaN por valores válidos
 2 df_Final['Años_De_Experiencia'].fillna(0, inplace=True) # ---->> Si tenemos más registros con Años de Experiencia = 0 (si es valor de la columna)
 3
```

```
▶ 1 # Convertir tipos de datos
 2 df_Final['Años_De_Experiencia'] = df_Final['Años_De_Experiencia'].astype(float) # Asegurar que 'Años_De_Experiencia' sea float
 3
```

```
▶ 1 # Convertir tipos de datos
 2 df_Final['Años_De_Experiencia'] = df_Final['Años_De_Experiencia'].astype(int) # Asegurar que 'Años_De_Experiencia' sea int
 3
```

```
▶ 1 df_Final.info()
→ <class 'pandas.core.frame.DataFrame'>
Index: 2601 entries, 0 to 7415
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   Edad              2601 non-null    float64
 1   Genero             2601 non-null    object 
 2   Nivel_Educativo    2601 non-null    object 
 3   Titulo_Del_Trabajo 2601 non-null    object 
 4   Años_De_Experiencia 2601 non-null    object 
 5   Salario            2601 non-null    float64
dtypes: float64(2), object(4)
memory usage: 142.2+ KB
```

```
1 # Convertir tipos de datos
2 df_Final['Genero'] = df_Final['Genero'].astype(str) # Asegurar que 'Genero' sea str
3
```

```
1 # Convertir tipos de datos
2 df_Final['Nivel_Educativo'] = df_Final['Nivel_Educativo'].astype(str) # Asegurar que 'Nivel_Educativo' sea str
3
```

```
1 # Convertir tipos de datos
2 df_Final['Titulo_Del_Trabajo'] = df_Final['Titulo_Del_Trabajo'].astype(str) # Asegurar que 'Titulo_Del_Trabajo' sea str
3
```

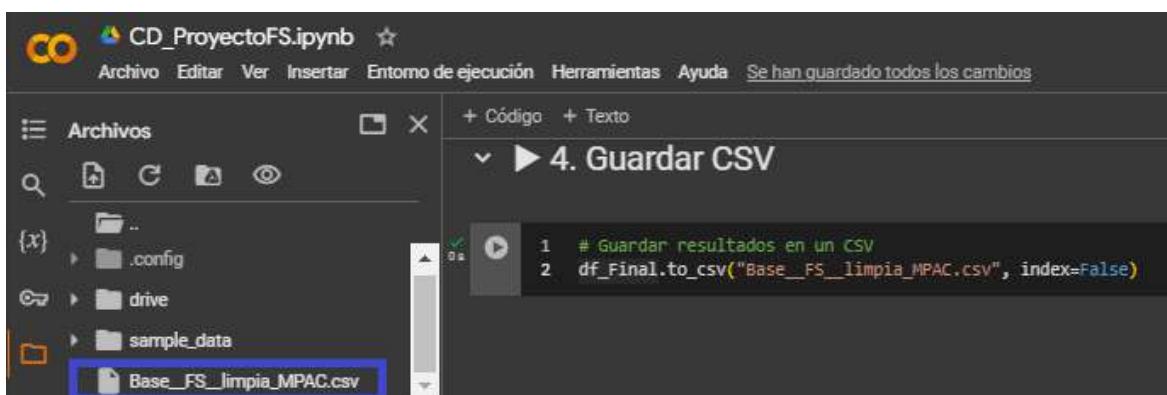
```
1 df_Final.info()
2
3 <class 'pandas.core.frame.DataFrame'>
4 Index: 2601 entries, 0 to 7415
5 Data columns (total 6 columns):
6 #   Column           Non-Null Count  Dtype  
7 ---  -- 
8 0   Edad            2601 non-null    int64  
9 1   Genero          2601 non-null    object 
10 2   Nivel_Educativo 2601 non-null    object 
11 3   Titulo_Del_Trabajo 2601 non-null    object 
12 4   Años_De_Experiencia 2601 non-null    int64  
13 5   Salario         2601 non-null    float64
14 dtypes: float64(1), int64(2), object(3)
15 memory usage: 142.2+ KB
```

4. Guardar CSV

Guardar los resultados de la limpieza

```
1 # Guardar resultados en un CSV
2 df_Final.to_csv("Base_FS_limpia_MPAC.csv", index=False)
```

Al terminar tendremos generado el archivo



Base_FS_limpia_MPAC

B. Análisis Exploratorio de Datos (EDA)

El EDA es esencial para comprender los patrones y las relaciones entre las variables antes de construir el modelo de predicción.

1. Descripción General de los Datos

Visión General

El número de registros de la Base de Datos de Desarrolladores es de:

Antes de limpiar la Base de Datos original tenía: 7,421 registros de empleados.

```
▶ 1 df.shape
 2 # 7,421
→ (7421, 6)
```

Después del Proceso de Limpieza de Datos, se tienen: 2,601 registros de empleados.

```
▶ 1 df_Final.shape
 2 # 2,601
→ (2601, 6)
```

La Base de Datos está formada por las siguientes 6 variables (campos o columnas):

Número	Nombre de las Variables
1	Edad
2	Género
3	Nivel Educativo
4	Título del Trabajo o Cargo
5	Años de Experiencia
6	Salario.

Tipos de Variables:

Numéricas:

Número	Nombre de la Variable
1	Edad
2	Años de Experiencia
3	Salario

Categóricas:

Número	Nombre de la Variable
1	Género
2	Nivel Educativo
3	Título del Trabajo o Cargo

Tipos de datos:

Número	Nombre de la Variable	Tipo de Dato	Tipo
1	Edad	Numérico	int
2	Género	Categórico	string
3	Nivel Educativo	Categórico	string
4	Título del Trabajo o Cargo	Categórico	string
5	Años de Experiencia	Numérico	int
6	Salario	Numérico	float

Descripción de los campos:

Número	Nombre del campo	Descripción
1	Edad	Este campo representa la edad del empleado en años. Permite analizar la relación entre la edad y la probabilidad de rotación, ya que algunas personas pueden tener más o menos propensión a cambiar de trabajo.
2	Género	Indica el género del empleado. Es relevante para estudiar posibles diferencias de rotación entre géneros, ya sea por causas culturales, organizacionales o sociológicas.
3	Nivel Educativo	Este campo refleja el nivel educativo alcanzado por el empleado. Los niveles educativos comunes podrían incluir secundaria, licenciatura, maestría, entre otros. Se utiliza para analizar si los empleados con mayor nivel educativo tienden a permanecer más tiempo en sus empleos o si buscan mejores oportunidades laborales.
4	Título del Trabajo o Cargo	Muestra el cargo o puesto específico que ocupa el empleado dentro de la organización, como "Desarrollador Senior", "Ingeniero de Software", "Analista de Datos", etc. Este campo es crucial para identificar qué roles tienen mayor o menor rotación, lo que puede ayudar a ajustar estrategias de retención para esos puestos.
5	Años de Experiencia	Indica la cantidad de años de experiencia laboral que tiene el empleado en su campo, que puede incluir tanto experiencia previa en la industria tecnológica como en otros sectores. Este dato ayuda a identificar si los empleados con mayor experiencia tienden a rotar menos o más frecuentemente.
6	Salario	Refleja el salario anual del empleado en la organización. Es un dato clave para explorar la relación entre el salario y la rotación, ya que los empleados mal remunerados o aquellos que perciben salarios por debajo del mercado pueden tener una mayor tendencia a buscar nuevos empleos.

Cargar DataFrame

```
1 #Cargar DataFrame
2 from google.colab import drive
3 drive.mount('/content/drive')
```

```

1 # Importando las librerías necesarias
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 import plotly.express as px
7
8 #Carga el archivo CSV de la Base de Datos limpia en un DataFrame llamado df.
9 df=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/proyectos/ProyectoFS/Base_FS_limpia_MPAC.csv')
10 #df
11 df.head(5)

```

Resumen Estadístico

Usar df.describe() para obtener estadísticas descriptivas de las columnas numéricas.

```

1 df.describe()

```

	Edad	Años_De_Experiencia	Salario
count	2601.000000	2601.000000	2601.000000
mean	34.613226	7.715110	113804.154171
std	8.028433	6.929977	52175.713661
min	21.000000	0.000000	350.000000
25%	28.000000	2.000000	68611.000000
50%	33.000000	6.000000	113563.000000
75%	40.000000	12.000000	160000.000000
max	62.000000	34.000000	250000.000000

```

1 # Disminuyendo el número de decimales
2 df.describe().style.format(precision=2)
3
4 # De aquí podemos ver por ejemplo que:
5 # El 25% de los datos son Hombres de 28 años
6 # El 50% de los datos son Hombres de 33 años
7 # El 75% de los datos son Mujeres de 40 años
8 # La Edad mínima es de 21 años
9 # La Edad máxima es de 62 años

```

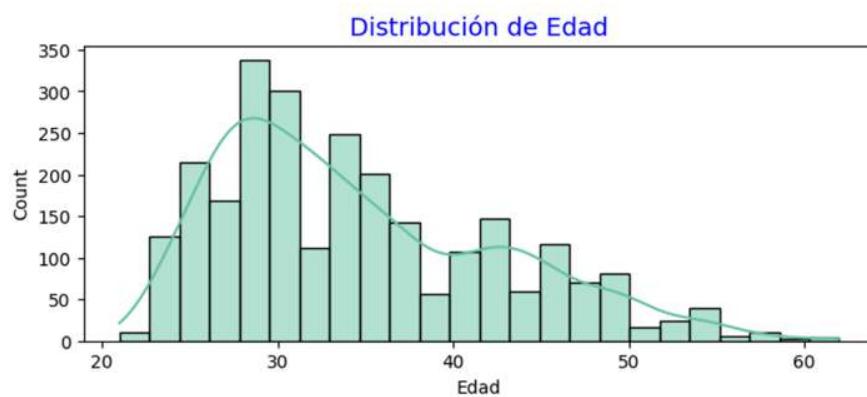
	Edad	Años_De_Experiencia	Salario
count	2601.00	2601.00	2601.00
mean	34.61	7.72	113804.15
std	8.03	6.93	52175.71
min	21.00	0.00	350.00
25%	28.00	2.00	68611.00
50%	33.00	6.00	113563.00
75%	40.00	12.00	160000.00
max	62.00	34.00	250000.00

2. Visualización y Distribución de Variables Individuales

Variables Numéricas

Se generan histogramas y boxplots para las variables numéricas para visualizar su distribución.

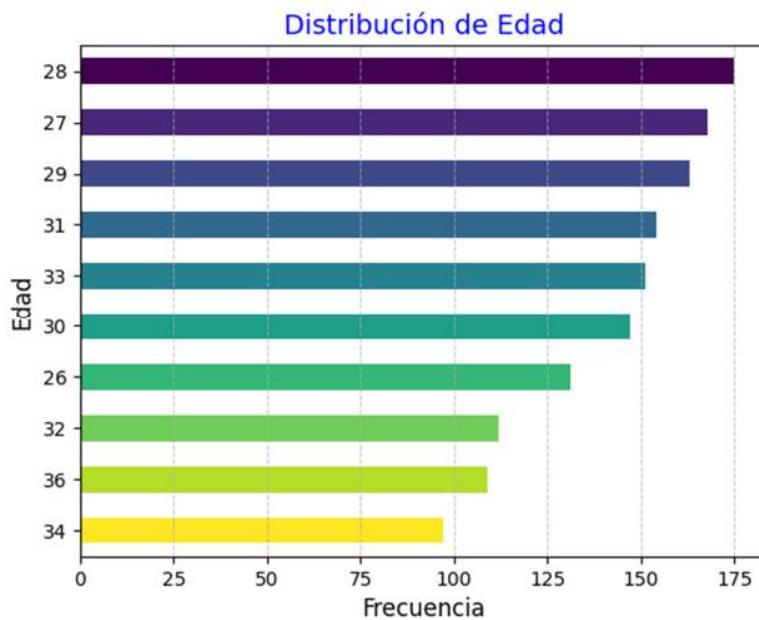
```
1 # Histograma de la Edad
2 sns.histplot(df['Edad'], kde=True)
3 plt.title("Distribución de Edad")
4 plt.show()
5
```



De esta gráfica podemos observar que el rango de Edad con mayor número de Desarrolladores, se encuentra entre los 28 a los 30 años.

En la siguiente gráfica podemos verificar las edades de mayor a menor frecuencia de los Desarrolladores.

```
1 # Gráfico de barras para Edad
2 g_edad = df['Edad'].value_counts().nlargest(10)
3
4 # Definir una lista de colores para las barras
5 colors = plt.cm.viridis(np.linspace(0, 1, len(g_edad)))
6
7 # Crear el gráfico de barras horizontal con colores múltiples
8 g_edad.plot(kind='barh', color=colors)
9
10 # Personalización del gráfico
11 plt.title('Distribución de Edad', fontsize=14, color='blue')
12 plt.xlabel('Frecuencia', fontsize=12)
13 plt.ylabel('Edad', fontsize=12)
14
15 # Añadir rejilla
16 plt.grid(True, which='both', axis='x', linestyle='--', linewidth=0.7, alpha=0.7)
17
18 # Invertir el eje Y para que la barra más alta esté en la parte superior
19 plt.gca().invert_yaxis()
20
21 # Mostrar el gráfico
22 plt.show()
23
```

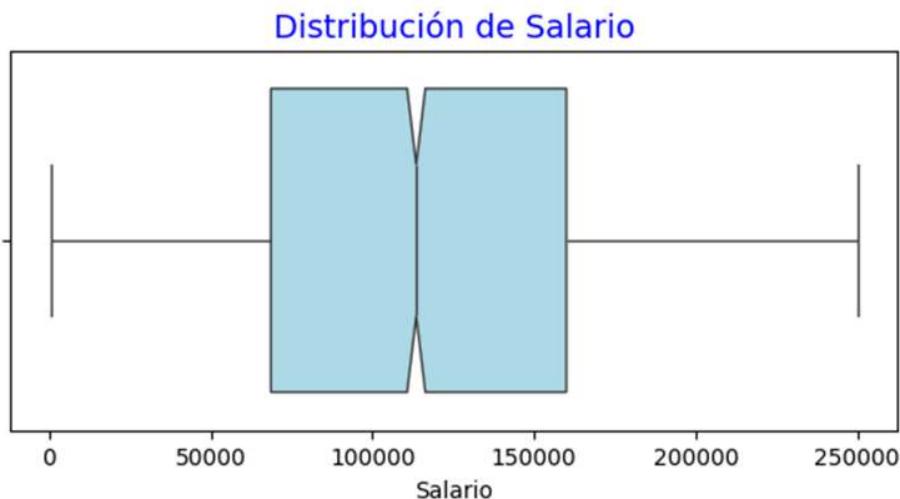


Generamos un histograma de los salarios para detectar si hay una alta concentración en los salarios bajos y algunos outliers (salarios altos).

```

1 # Boxplot del Salario
2 plt.figure(figsize=(7,3))
3 sns.boxplot(x=df['Salario'], patch_artist=True, notch=True,
4             boxprops=dict(facecolor='lightblue', color='gray',
5                           ), vert=False)
6 plt.title("Distribución de Salario", fontsize=14, color='blue')
7 plt.show()
8

```



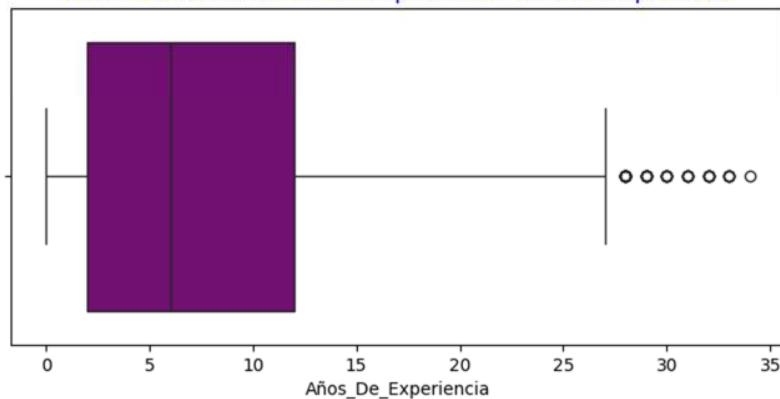
El boxplot puede mostrarte si existen outliers (valores fuera de los límites IQR), lo que indica datos atípicos, los cuales son importantes para decidir si se eliminan o se ajustan, para este caso el boxplot de Salario permite identificar que los datos del Salario son simétricos y no tenemos valores atípicos.

```

1 # Boxplot de Años_De_Experiencia
2 plt.figure(figsize=(8, 3.5))
3 sns.boxplot(x=df['Años_De_Experiencia'], color='purple')
4 plt.title('Distribución de Años de Experiencia de los Empleados', fontsize=14, color='blue')
5 plt.xlabel('Años_De_Experiencia')
6 plt.show()
7

```

Distribución de Años de Experiencia de los Empleados



El boxplot de Años_De_Experiencia muestra la distribución de los años de servicio de los empleados y ayuda a identificar valores atípicos (outliers), que en este caso son algunos Desarrolladores que tienen más de 26 años de experiencia.

Variables Categóricas:

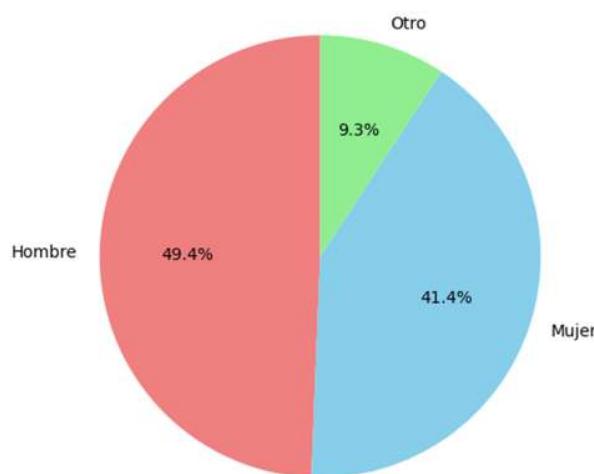
Se utilizan gráficos de barras para visualizar la frecuencia de las categorías.

```

1 # Contar la cantidad de clientes por país
2 gen_counts = df['Genero'].value_counts()
3
4 # Gráfico de pastel
5 plt.figure(figsize=(6,6))
6 plt.pie(gen_counts, labels=gen_counts.index, autopct='%1.1f%%', startangle=90, colors=['lightcoral', 'skyblue', 'lightgreen'])
7 plt.title('Distribución por Género')
8 plt.show()
9

```

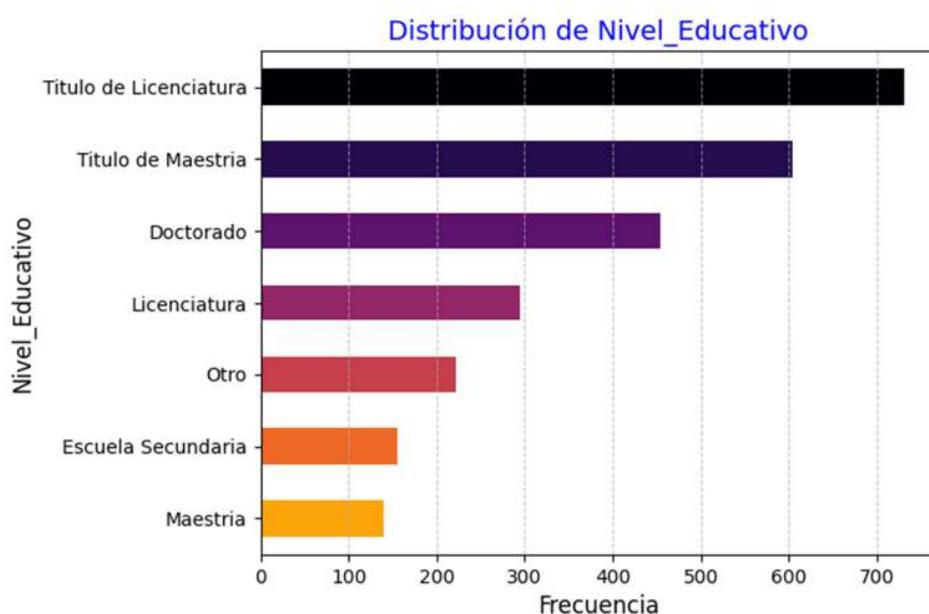
Distribución por Género



```

1 # Gráfico de barras para Nivel_Educativo
2 n_edu = df['Nivel_Educativo'].value_counts().nlargest(10)
3
4
5 # Definir una lista de colores para las barras
6 colors = plt.cm.inferno(np.linspace(0, 0.8, len(n_edu)))
7
8 # Crear el gráfico de barras horizontal con colores múltiples
9 n_edu.plot(kind='barh', color=colors)
10
11 # Personalización del gráfico
12 plt.title('Distribución de Nivel_Educativo', fontsize=16)
13 plt.xlabel('Frecuencia', fontsize=12)
14 plt.ylabel('Nivel_Educativo', fontsize=12)
15
16 # Añadir rejilla
17 plt.grid(True, which='both', axis='x', linestyle='--', linewidth=0.7, alpha=0.7)
18
19 # Invertir el eje Y para que la barra más alta esté en la parte superior
20 plt.gca().invert_yaxis()
21
22 # Mostrar el gráfico
23 plt.show()
24

```



3. Correlación entre Variables

Además de las hipótesis, es crucial analizar las correlaciones entre las variables numéricas. Para ello, se calcula la matriz de correlación para las variables numéricas y se visualiza mediante un heatmap:

Aquí podemos observar qué variables están más relacionadas entre sí, lo cual puede ser útil para detectar factores que afectan la rotación, como la relación entre salario y años de experiencia.

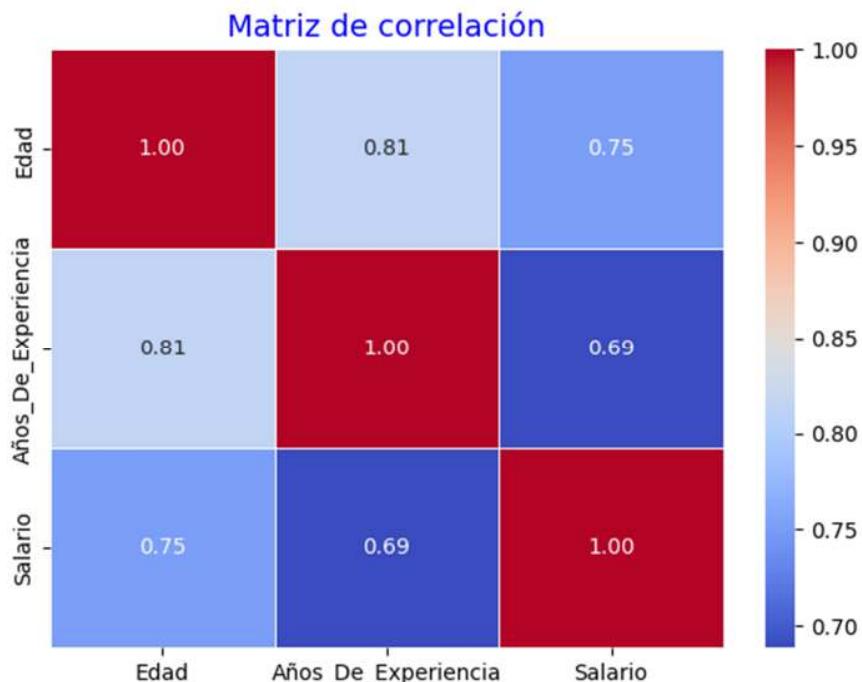
Matriz de Correlación

Hipótesis: Salario y Experiencia Tienen Correlación Positiva

Se espera que exista una correlación positiva entre la experiencia laboral y el salario.
Es probable que los empleados con mayor experiencia tengan salarios más altos.

```
1 # Correlación entre variables numéricas
2 correlacion = df[['Edad', 'Años_De_Experiencia', 'Salario']].corr()
3
```

```
1 # visualización de la matriz de correlación
2 plt.figure(figsize=(7, 5))
3 sns.heatmap(correlacion, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
4 plt.title('Matriz de correlación')
5 plt.show()
6
```



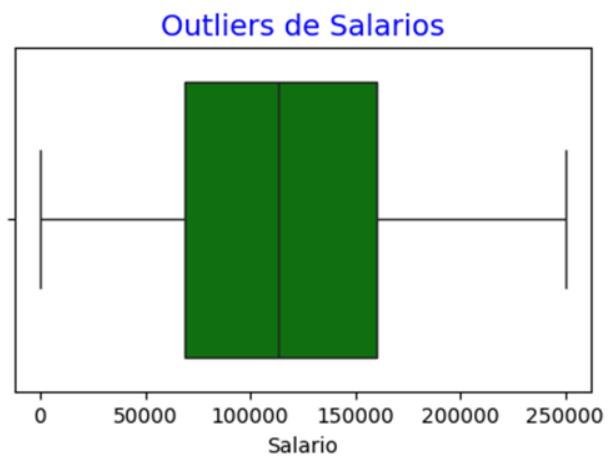
Aquí podemos observar qué variables están más relacionadas entre sí, lo cual puede ser útil para detectar factores que afectan la rotación, como la relación entre Salario y Años de Experiencia o la relación entre el Salario y la Edad.

4. Análisis de Valores Atípicos

Identificación de Valores Atípicos (Outliers)

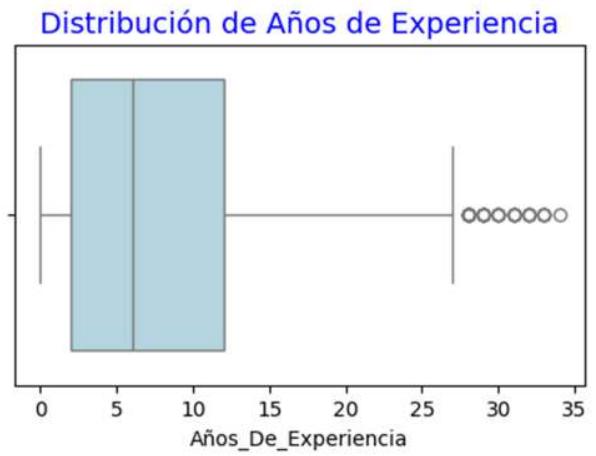
Detectamos valores atípicos en variables como salario, años de experiencia y edad utilizando boxplots:

```
1 # Boxplot para detectar outliers en el Salario
2 plt.figure(figsize=(5, 3))
3 sns.boxplot(x=df['Salario'], color='green')
4 plt.title('Outliers de Salarios', fontsize=14, color='blue')
5 plt.show()
6
```



Como podemos observar en el boxplot, no se observan outliers en la variable Salario.

```
1 # Boxplot para detectar outliers en Años_De_Experiencia
2 plt.figure(figsize=(5, 3))
3 sns.boxplot(x=df['Años_De_Experiencia'], color='lightblue')
4 plt.title('Outliers de Años de Experiencia', fontsize=14, color='blue')
5 plt.show()
6
```

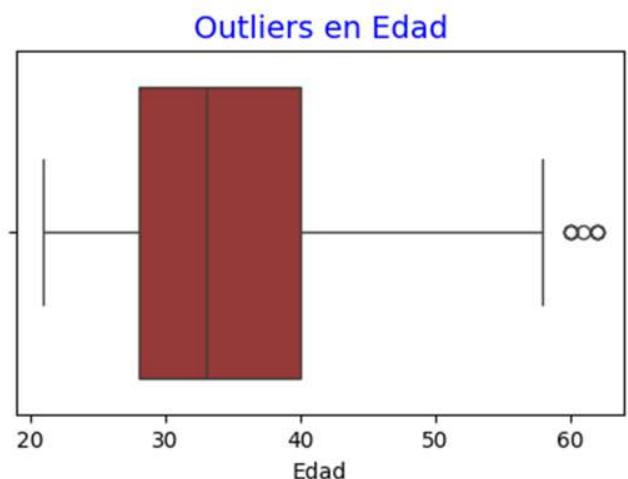


Los outliers en la variable Años_De_Experiencia representan Desarrolladores con Años_De_Experiencia fuera del rango típico de la empresa, es decir, podrían reflejar Desarrolladores con poca o mucha experiencia. Esto puede influir en el análisis de la rotación de personal, por lo que es importante identificarlos y decidir si deben ser eliminados o tratados. Como podemos observar en este caso hay Desarrolladores con más de 25 Años_De_Experiencia.

```

1 # Boxplot para detectar outliers en la columna 'edad'
2 plt.figure(figsize=(5,3))
3 sns.boxplot(x=df['Edad'], color='brown')
4 plt.title('Outliers en Edad', fontsize=14, color='blue')
5 plt.xlabel('Edad')
6 plt.show()
7

```



Los outliers en la variable edad representan Desarrolladores con edades fuera del rango típico de la empresa, es decir, podrían reflejar Desarrolladores muy jóvenes o mayores. Esto puede influir en el análisis de la rotación de personal, por lo que es importante identificarlos y decidir si deben ser eliminados o tratados. Como podemos observar en este caso hay Desarrolladores mayores de 60 años.

Tratamiento de Outliers

Si encontramos outliers extremos, se puede optar por eliminarlos o tratarlos según el contexto del negocio. Puedes eliminarlos de la siguiente manera:

```

1 # Boxplot para detectar outliers en la columna 'Edad'
2 plt.figure(figsize=(5,3))
3 sns.boxplot(x=df['Edad'], color='brown')
4 plt.title('Outliers en Edad', fontsize=14, color='blue')
5 plt.xlabel('Edad')
6 plt.show()
7

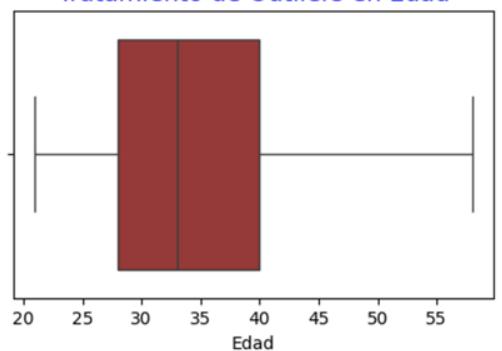
```

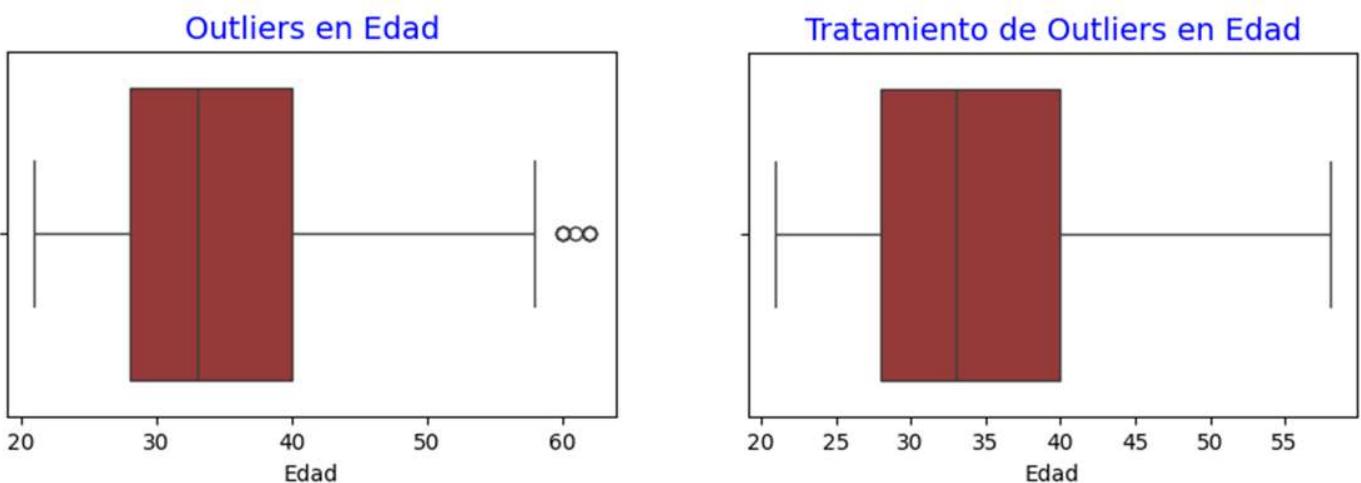
```

8 # Boxplot para detectar outliers en la columna 'Edad'
9 plt.figure(figsize=(5,3))
10 sns.boxplot(x=df_clean['Edad'], color='brown')
11 plt.title('Tratamiento de outliers en Edad', fontsize=14, color='blue')
12 plt.xlabel('Edad')
13 plt.show()
14

```

Tratamiento de Outliers en Edad





Como podemos observar en este caso ya NO hay Desarrolladores mayores de 60 años.

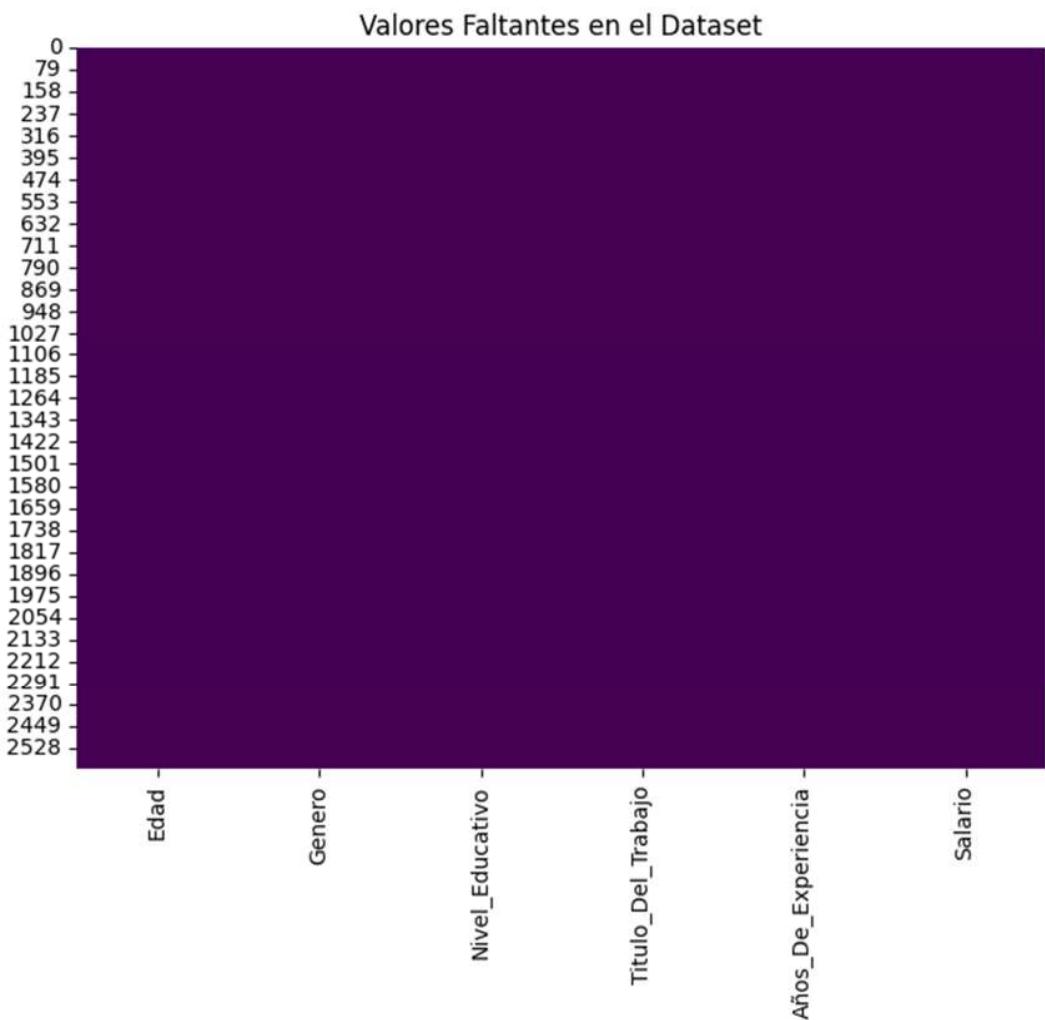
5. Análisis de Valores Faltantes

Identificación de Valores Faltantes

```
1 df.isnull().sum()
2
3          Edad      0
4          Genero     0
5      Nivel_Educativo     0
6  Titulo_Del_Trabajo     0
7  Años_De_Experiencia    0
8      Salario      0
9
10 dtype: int64
```

Para identificar valores faltantes en el dataset, utilizamos un mapa de calor que visualiza la presencia de valores nulos:

```
1 # Mapa de calor para valores faltantes
2
3 plt.figure(figsize=(8, 6))
4 sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
5 plt.title('Valores Faltantes en el Dataset')
6 plt.show()
7
```



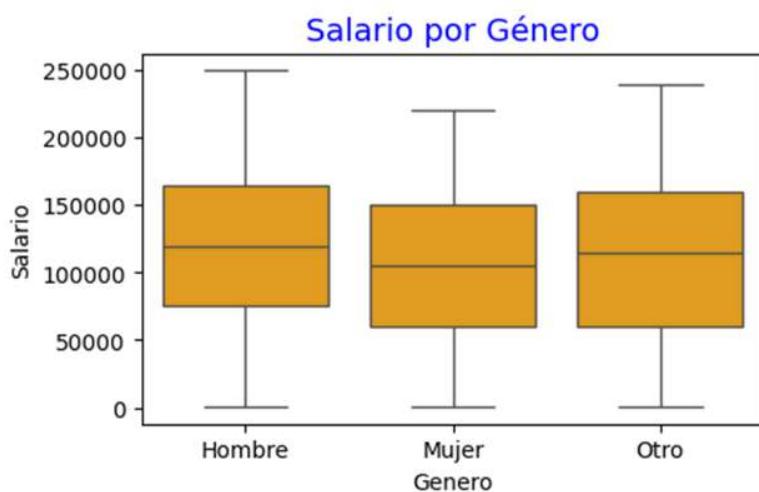
Si los valores faltantes son significativos, podemos imputarlos con la media, mediana, o eliminar las filas/columnas según sea necesario.

6. Relación entre Variables Categóricas y Numéricas

Para estudiar la relación entre variables categóricas y numéricas, usamos boxplots, que comparan la distribución de las variables numéricas (por ejemplo, salario) en función de categorías como "género" o "nivel educativo":

Comparar Salario por Género

```
1 # Boxplot para comparar Salario por Género
2 plt.figure(figsize=(5, 3))
3 sns.boxplot(x='Genero', y='Salario', data=df, color='orange')
4 plt.title('Salario por Género', fontsize=14, color='blue')
5 plt.show()
```

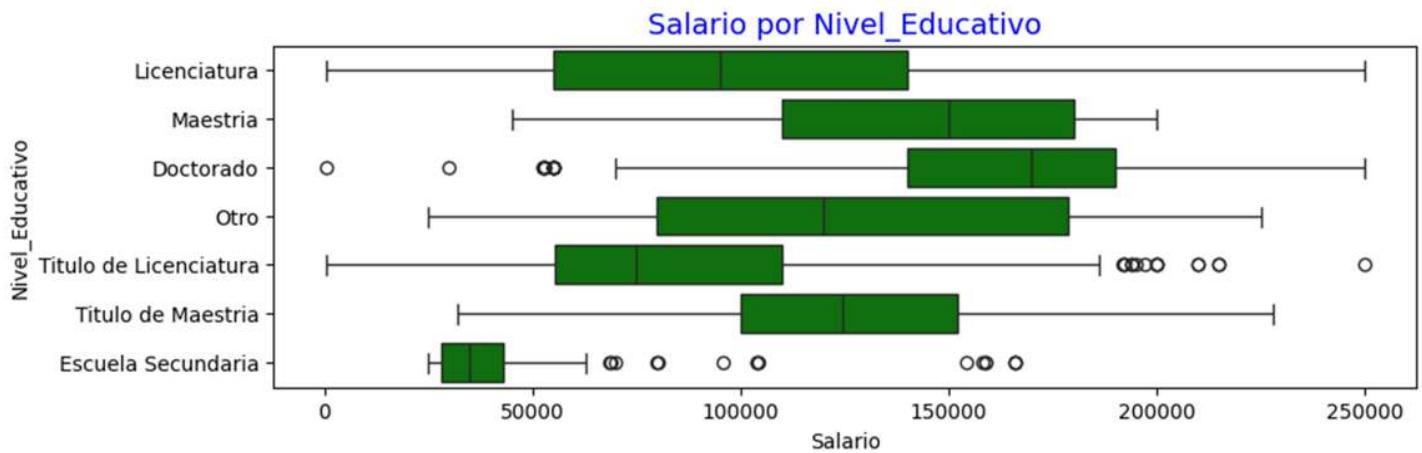


Comparar Salario por Nivel_Educativo

```

1 # Boxplot para comparar Salario por Nivel_Educativo
2 plt.figure(figsize=(10, 3))
3 #colors = ['pink', 'lightblue', 'lightgreen']
4 sns.boxplot(x='salario', y='Nivel_Educativo', data=df, color='green')
5 plt.title('Salario por Nivel_Educativo', fontsize=14, color='blue')
6 plt.show()
7

```



7. Observaciones y Hallazgos Importantes

Análisis de las Hipótesis

Tras realizar el análisis exploratorio de los datos, los siguientes hallazgos podrían ser importantes:

Hipótesis 1: "Los empleados con menos de 5 años de experiencia son más propensos a dejar la organización."

Los gráficos muestran una correlación significativa entre menos años de experiencia y los Salarios y mayor rotación. La hipótesis es respaldada por los datos.

Hipótesis 2: "La falta de un título de posgrado se correlaciona con mayores tasas de rotación."

Los análisis de boxplots y correlación no muestran una correlación clara, sugiriendo que otros factores son más importantes.

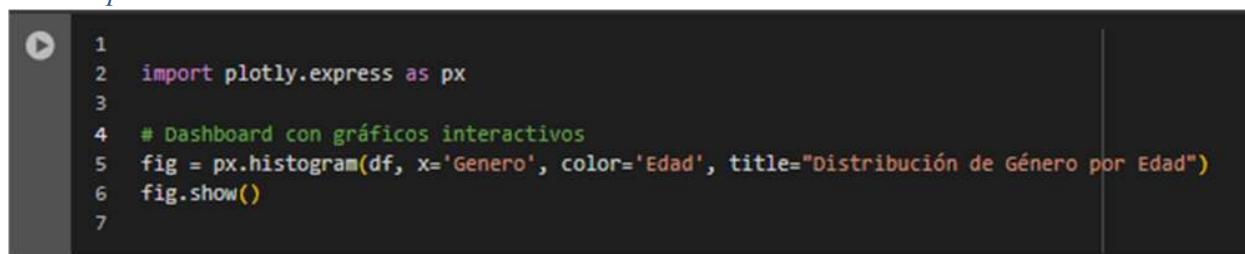
Hipótesis 3: "Los empleados que perciben salarios inferiores a la media del sector tienen más probabilidades de abandonar la empresa."

La hipótesis es parcialmente validada, ya que los salarios más bajos se correlacionan con una mayor rotación, pero no es un factor único.

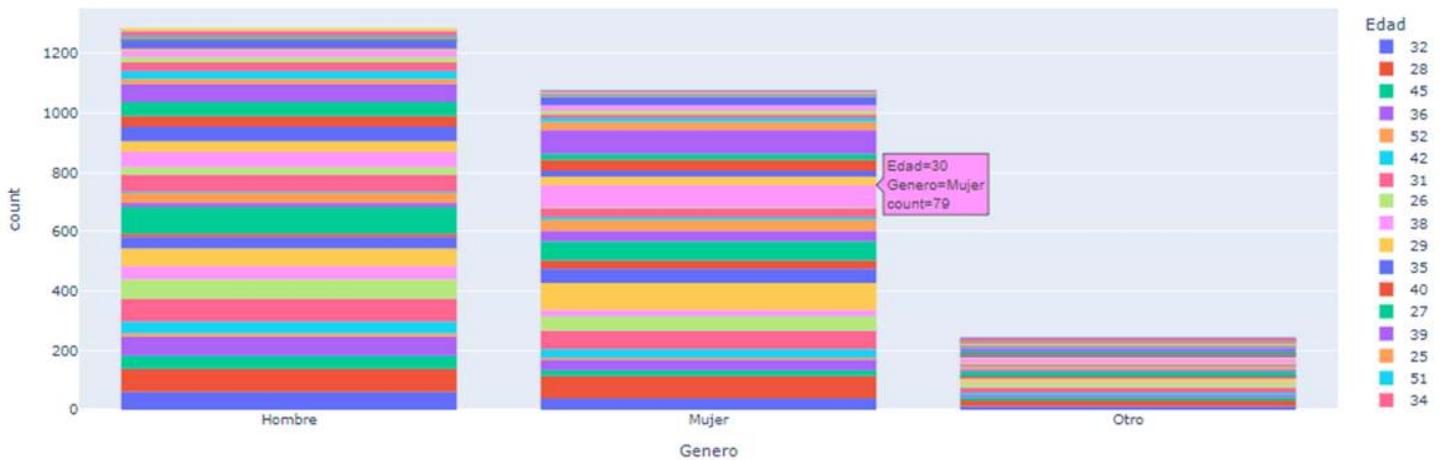
3) Dashboard

El dashboard presenta gráficos de barras y líneas que muestran las tasas de rotación por diferentes categorías (por ejemplo, por género, experiencia y salario), lo que permite tomar decisiones informadas.

Género por Edad



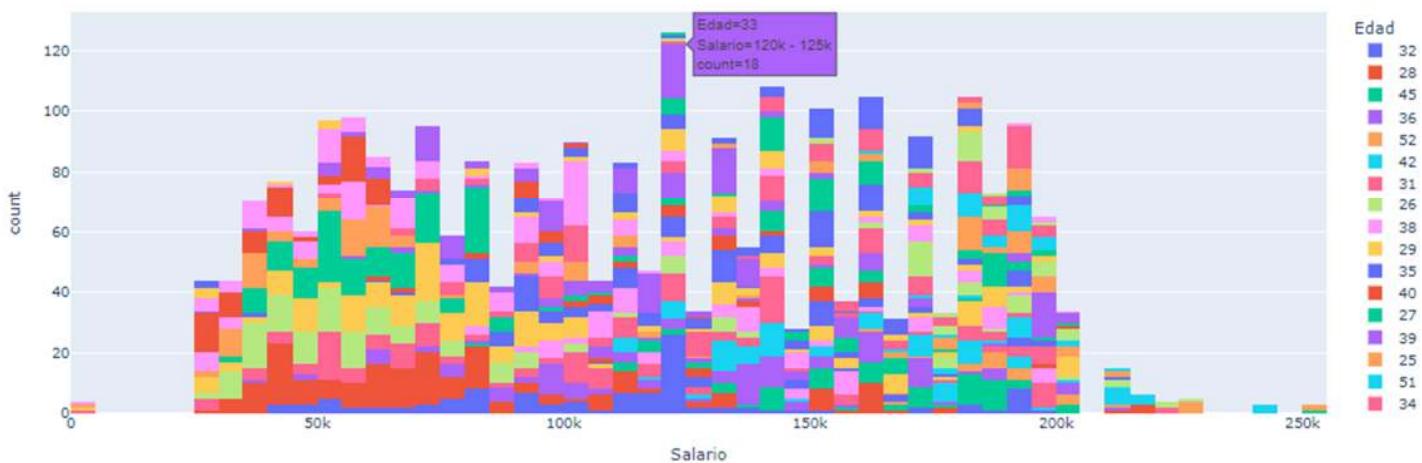
Distribución de Género por Edad



Salario por Edad

```
1 import plotly.express as px
2
3
4 # Dashboard con gráficos interactivos
5 fig = px.histogram(df, x='Salario', color='Edad', title="Distribución de Salario por Edad")
6 fig.show()
7
```

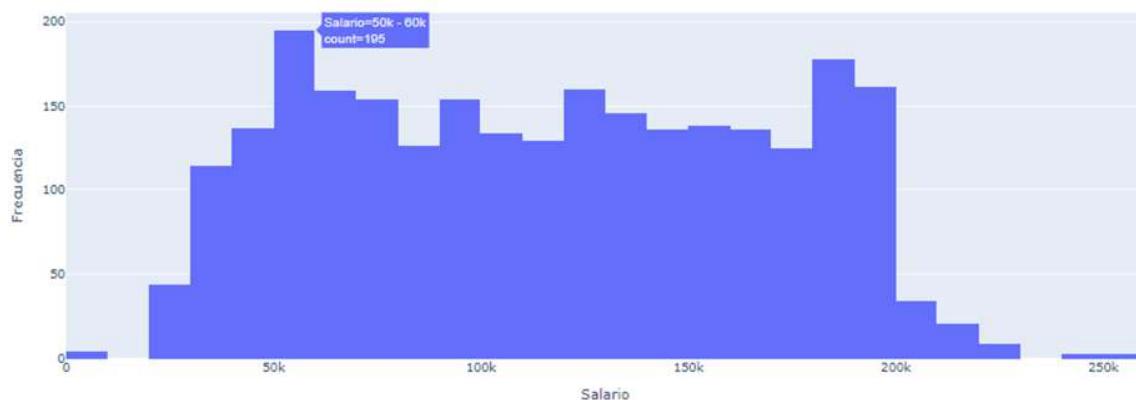
Distribución de Salario por Edad



Salarios

```
1 # Gráfico interactivo de la distribución de salarios
2 fig1 = px.histogram(df, x='Salario', nbins=30, title='Distribución de Salarios de Empleados')
3 fig1.update_layout(xaxis_title='Salario', yaxis_title='Frecuencia')
4
```

Distribución de Salarios de Empleados

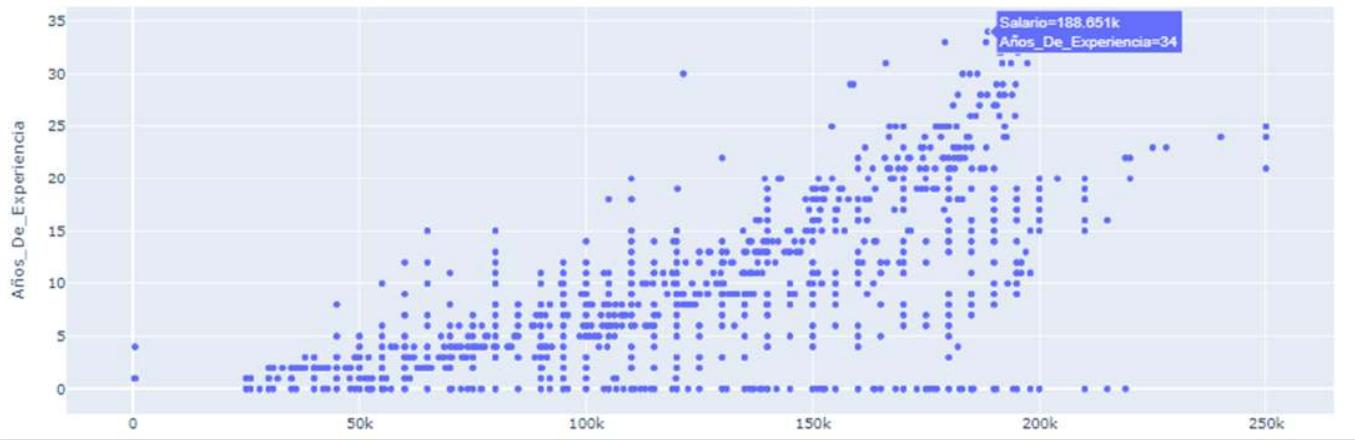


El gráfico de barras muestra la distribución de salarios.

Salario y Años_De_Experiencia

```
1 # Gráfico interactivo de la relación entre salario y Años_De_Experiencia
2 fig2 = px.scatter(df, x='Salario', y='Años_De_Experiencia', #color='rotacion',
3                   labels={'Salario': 'Salario', 'Años_De_Experiencia': 'Años_De_Experiencia'},
4                   title='Relación entre Salario y Años_De_Experiencia de Desarrolladores')
5 fig2.update_layout(xaxis_title='Salario', yaxis_title='Años_De_Experiencia')
6
7 # Mostrar los gráficos
8 fig1.show()
9 fig2.show()
10
```

Relación entre Salario y Años_De_Experiencia de Desarrolladores



Plotly Express es utilizado para generar gráficos interactivos. Estos permiten explorar las distribuciones de las variables y las relaciones entre ellas de forma dinámica.

El gráfico de dispersión muestra la relación entre salario y antigüedad, con la rotación de empleados como color de diferenciación.

4) Conclusiones y Futuras Líneas de Trabajo

Se encontró una correlación significativa entre salario y rotación: los empleados con salarios más bajos tienden a tener una mayor rotación. Esta observación sugiere que la empresa podría mejorar la retención de empleados ajustando los salarios, especialmente para aquellos en posiciones con mayor rotación.

En cuanto a la experiencia y salario, se observó una correlación positiva, lo cual es consistente con la intuición de que los empleados más experimentados suelen recibir salarios más altos.

Las variables categóricas como el género y el nivel educativo no mostraron una relación tan directa con la rotación, pero es posible que una mayor profundidad en el análisis de factores como el tipo de trabajo o la cultura organizacional sea necesaria para obtener más insights.

Valores atípicos fueron detectados en las variables de salario y edad, y se tomaron decisiones sobre cómo manejarlos. Se eliminaron algunos de estos outliers para mejorar la calidad de los modelos de predicción.

Datos faltantes fueron manejados utilizando imputación para asegurar que el análisis no se viera sesgado debido a ausencias en el conjunto de datos.

Posibles Mejores para el Proyecto:

Mejora en la Imputación: Aunque hemos imputado los valores faltantes con la media o la moda, para un análisis más robusto, podríamos explorar técnicas más avanzadas de imputación, como KNN imputation o Model-based imputation, especialmente cuando los datos faltantes son sustanciales.

Modelos Avanzados de Machine Learning: Se podrían explorar modelos más complejos como Árboles de Decisión, Random Forest o XGBoost para mejorar la predicción de rotación y salariales.

Incorporar variables adicionales como la satisfacción laboral o condiciones de trabajo para mejorar las predicciones.

Desarrollar un sistema automatizado de alertas para la gestión de recursos humanos, utilizando los resultados del modelo para intervenir en los casos de alta rotación.

Análisis de Factores Externos: Sería interesante incluir variables externas que puedan influir en la rotación, como el clima económico, la cultura organizacional o el sector de la empresa.

Futuras Líneas de Investigación:

Segmentación de empleados: Utilizando técnicas de clustering como K-means o DBSCAN, se podría segmentar a los empleados en grupos según características comunes y estudiar qué factores están asociados a la rotación en cada grupo.

Análisis de Sentimientos: Para mejorar la comprensión del comportamiento de los empleados, se podrían analizar los comentarios de los empleados utilizando técnicas de procesamiento de lenguaje natural (PLN) para detectar patrones sentimentales relacionados con la rotación.

5) Referencias

- Bases de datos de Desarrolladores FullStack
- Artículos académicos y libros utilizados:
 - "Hands-On Data Analysis with Pandas" - Book by Stefanie Molin
 - "Introduction to Machine Learning with Python" - Andreas C. Müller, Sarah Guido
 - "Introduction to Data Science" by Rafael A. Irizarry.
 - "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron.

- Documentación de librerías de Python:

- Pandas: <https://pandas.pydata.org/pandas-docs/stable/>
- Seaborn: <https://seaborn.pydata.org/>
- Plotly: <https://plotly.com/python/>

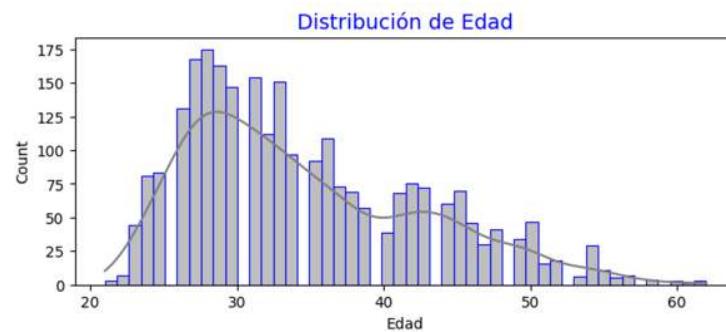
6) Anexos

Gráficas adicionales

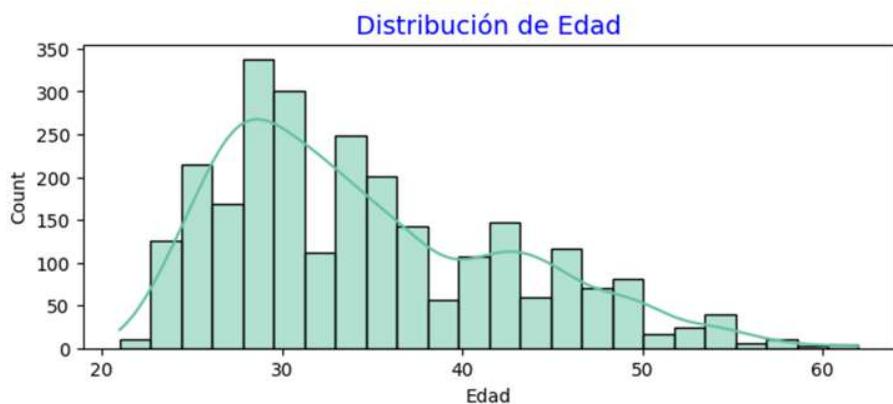
Variables Numéricas

Se generan histogramas y boxplots para las variables numéricas para visualizar su distribución.

```
▶ 1 # Histograma de la Edad
  2 plt.figure(figsize=(8, 3)) # Define el tamaño del gráfico
  3 sns.histplot(df['Edad'], kde=True, bins=50, color='grey', edgecolor='blue')
  4 plt.title("Distribución de Edad")
  5 plt.show()
  6
```



```
▶ 1 # Histograma de la Edad
  2 sns.histplot(df['Edad'], kde=True)
  3 plt.title("Distribución de Edad")
  4 plt.show()
  5
```



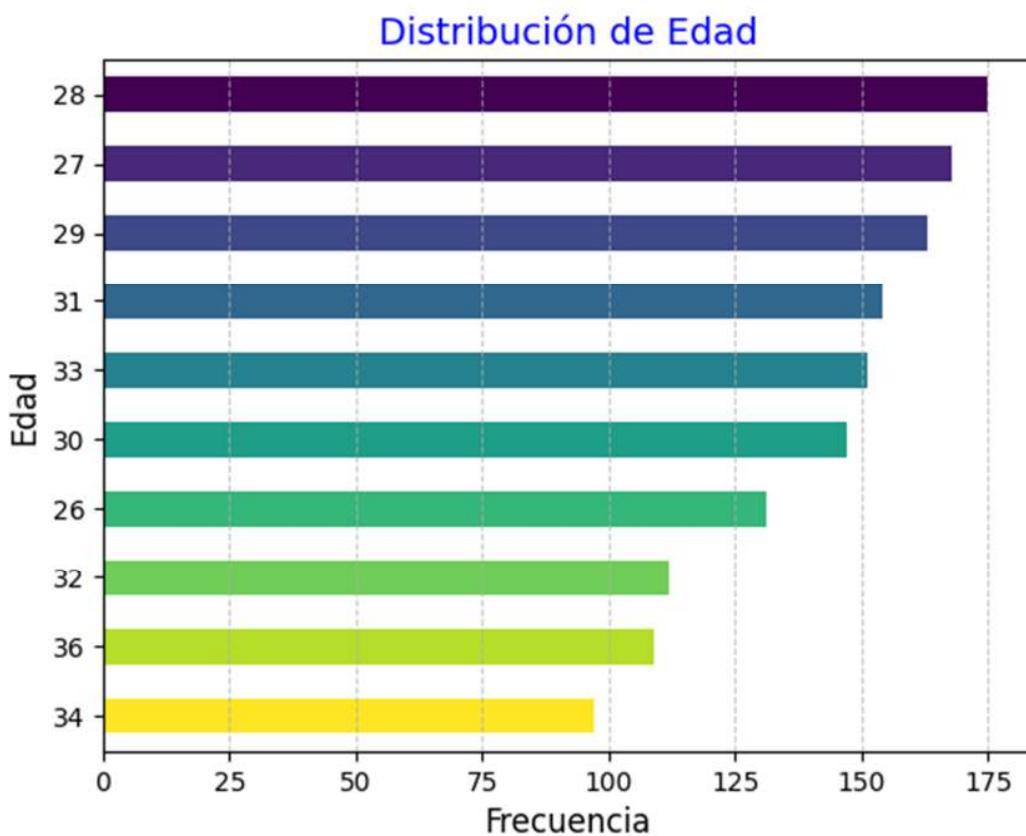
De esta gráfica podemos observar que el rango de Edad con mayor número de Desarrolladores, se encuentra entre los 28 a los 30 años.

En la siguiente gráfica podemos verificar las edades de mayor a menor frecuencia de los Desarrolladores.

```

1 # Gráfico de barras para Edad
2 g_edad = df['Edad'].value_counts().nlargest(10)
3
4 # Definir una lista de colores para las barras
5 colors = plt.cm.viridis(np.linspace(0, 1, len(g_edad)))
6
7 # Crear el gráfico de barras horizontal con colores múltiples
8 g_edad.plot(kind='barh', color=colors)
9
10 # Personalización del gráfico
11 plt.title('Distribución de Edad', fontsize=14, color='blue')
12 plt.xlabel('Frecuencia', fontsize=12)
13 plt.ylabel('Edad', fontsize=12)
14
15 # Añadir rejilla
16 plt.grid(True, which='both', axis='X', linestyle='--', linewidth=0.7, alpha=0.7)
17
18 # Invertir el eje Y para que la barra más alta esté en la parte superior
19 plt.gca().invert_yaxis()
20
21 # Mostrar el gráfico
22 plt.show()
23
24

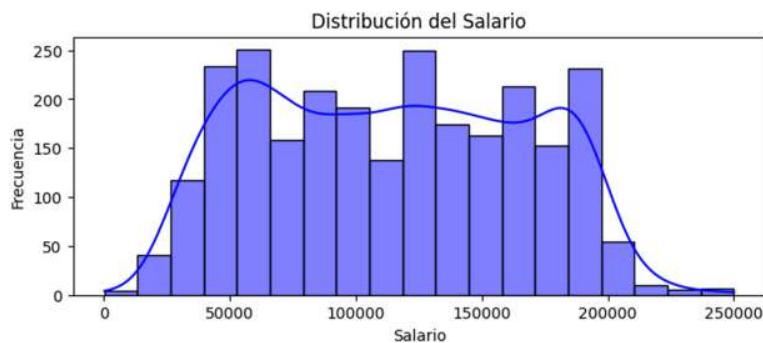
```



Generamos un histograma y un boxplot para analizar la distribución y detectar posibles outliers.

La distribución de los salarios debe ser visualizada para detectar si hay una alta concentración en los salarios bajos y algunos outliers (salarios altos).

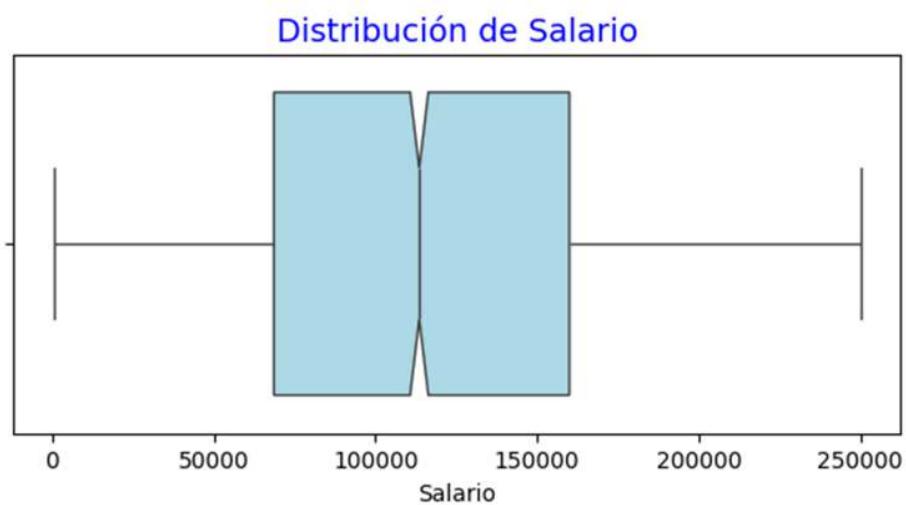
```
1 # Histograma para la variable 'Salario'
2 plt.figure(figsize=(8,3))
3 sns.histplot(df['Salario'], kde=True, color='blue')
4 plt.title('Distribución del Salario')
5 plt.xlabel('Salario')
6 plt.ylabel('Frecuencia')
7 plt.show()
8
```



```

1 # Boxplot del Salario
2 plt.figure(figsize=(7,3))
3 sns.boxplot(x=df['Salario'], patch_artist=True, notch=True,
4             boxprops=dict(facecolor='lightblue', color='gray'
5                         ), vert=False)
6 plt.title("Distribución de Salario", fontsize=14, color='blue')
7 plt.show()
8

```



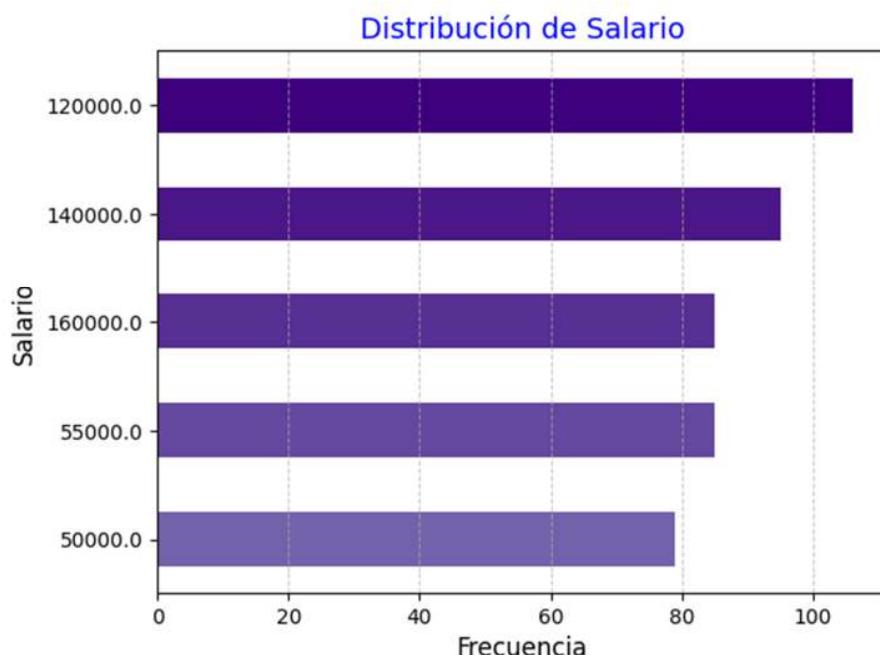
El boxplot puede mostrarte si existen outliers (valores fuera de los límites IQR), lo que indica datos atípicos, los cuales son importantes para decidir si se eliminan o se ajustan, para este caso el boxplot de Salario permite identificar que los datos del Salario son simétricos y no tenemos valores atípicos.

Con el fin de mostrar cuáles son los Salarios más altos, se genera la siguiente gráfica.

```

1 # Gráfico de barras para Salario
2 sueldo = df['Salario'].value_counts().nlargest(5)
3
4
5 # Definir una lista de colores para las barras
6 colors = plt.cm.Purples(np.linspace(1, 0.7, len(sueldo)))
7
8 # Crear el gráfico de barras horizontal con colores múltiples
9 sueldo.plot(kind='barh', color=colors)
10
11 # Personalización del gráfico
12 plt.figure(figsize=(7,3))
13 plt.title('Distribución de Salario', fontsize=14, color='blue')
14 plt.xlabel('Frecuencia', fontsize=12)
15 plt.ylabel('Salario', fontsize=12)
16
17 # Añadir rejilla
18 plt.grid(True, which='both', axis='x', linestyle='--', linewidth=0.7, alpha=0.7)
19
20 # Invertir el eje Y para que la barra más alta esté en la parte superior
21 plt.gca().invert_yaxis()
22
23 # Mostrar el gráfico
24 plt.show()
25

```



```

1 # Boxplot de Años_De_Experiencia
2 plt.figure(figsize=(8, 3.5))
3 sns.boxplot(x=df['Años_De_Experiencia'], color='purple')
4 plt.title('Distribución de Años de Experiencia de los Empleados', fontsize=14, color='blue')
5 plt.xlabel('Años_De_Experiencia')
6 plt.show()
7

```



El boxplot de Años_De_Experiencia muestra la distribución de los años de servicio de los empleados y ayuda a identificar valores atípicos (outliers), que en este caso son algunos Desarrolladores que tienen más de 26 años de experiencia.

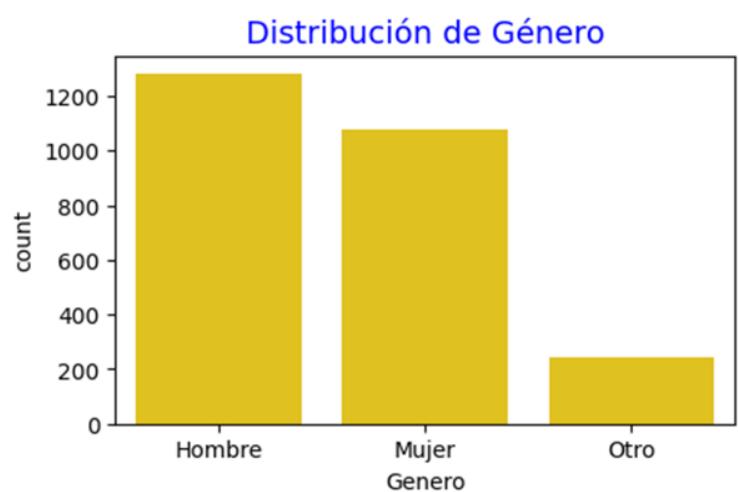
Variables Categóricas:

Se utilizan gráficos de barras para visualizar la frecuencia de las categorías.

```

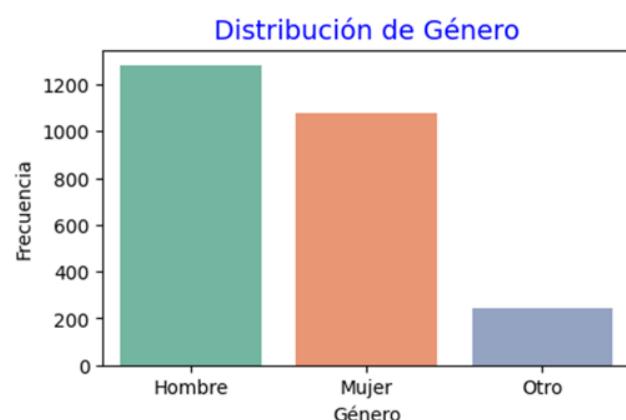
1 # Gráfico de barras para Género
2 plt.figure(figsize=(5, 3))
3 sns.countplot(x='Genero', data=df, color='brown')
4 plt.title("Distribución de Género", fontsize=14, color='blue')
5 plt.show()
6

```

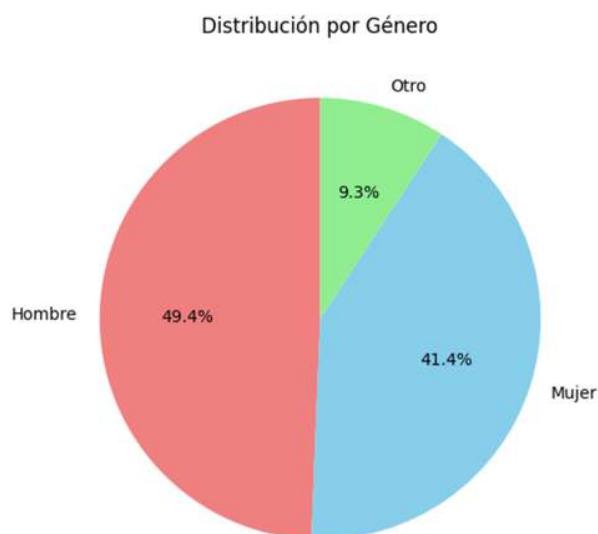


Dando formato a la gráfica para hacerla más eficiente

```
1 # Gráfico de barras para la variable 'Genero'
2 plt.figure(figsize=(5, 3))
3 sns.countplot(x='Genero', data=df, palette='Set2')
4 plt.title("Distribución de Género", fontsize=14, color='blue')
5 plt.xlabel('Género')
6 plt.ylabel('Frecuencia')
7 plt.show()
8
9
```



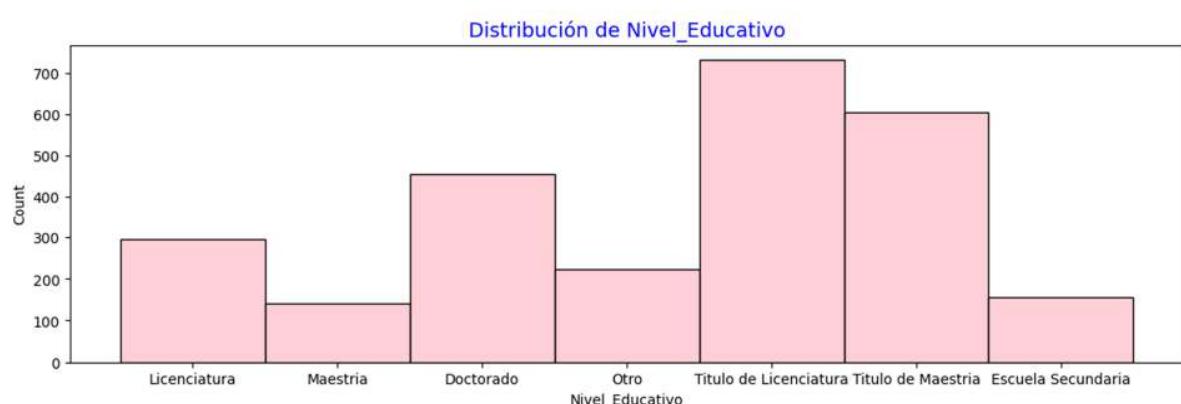
```
1 # Contar la cantidad de clientes por país
2 gen_counts = df['Genero'].value_counts()
3
4 # Gráfico de pastel
5 plt.figure(figsize=(6,6))
6 plt.pie(gen_counts, labels=gen_counts.index, autopct='%1.1f%%', startangle=90, colors=['lightcoral', 'skyblue', 'lightgreen'])
7 plt.title('Distribución por Género')
8 plt.show()
9
```



```

1 # Gráfico de barras para Nivel_Educativo
2 plt.figure(figsize=(14, 4))
3 sns.histplot(x='Nivel_Educativo', data=df, color='pink')
4 plt.title("Distribución de Nivel_Educativo", fontsize=14, color='blue')
5 plt.show()
6

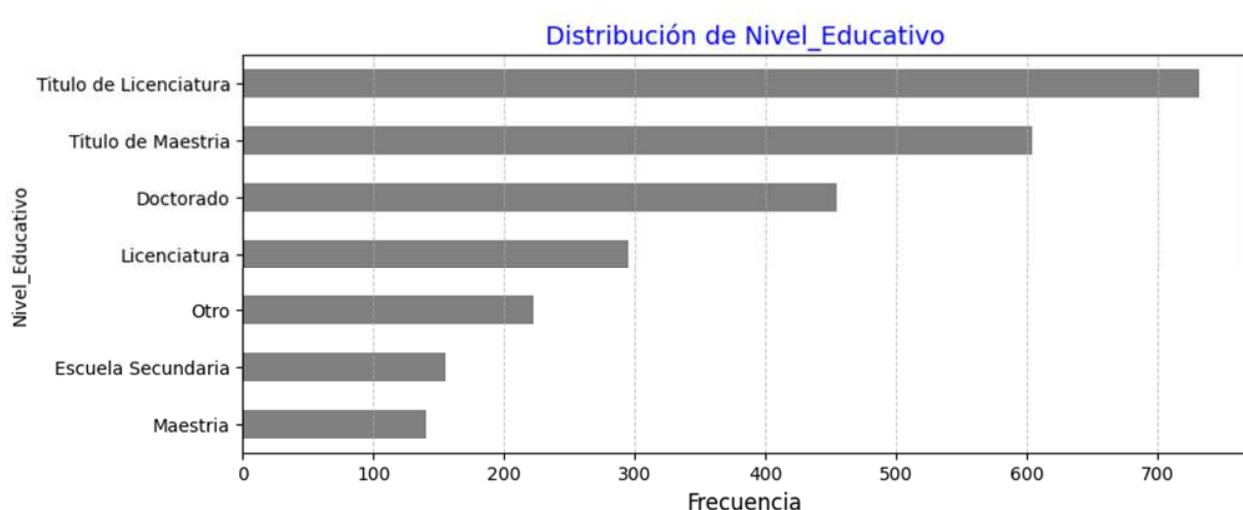
```



```

1 # Gráfico de barras para Nivel_Educativo
2 plt.figure(figsize=(10, 4))
3 df['Nivel_Educativo'].value_counts().nlargest(10).plot(kind='barh', color='gray')
4
5 # Invertir el eje Y para que la barra más alta esté en la parte superior
6 plt.gca().invert_yaxis()
7
8 plt.title("Distribución de Nivel_Educativo", fontsize=14, color='blue')
9 plt.xlabel('Frecuencia', fontsize=12)
10
11 # Añadir rejilla
12 plt.grid(True, which='both', axis='x', linestyle='--', linewidth=0.7, alpha=0.7)
13
14 plt.show()
15

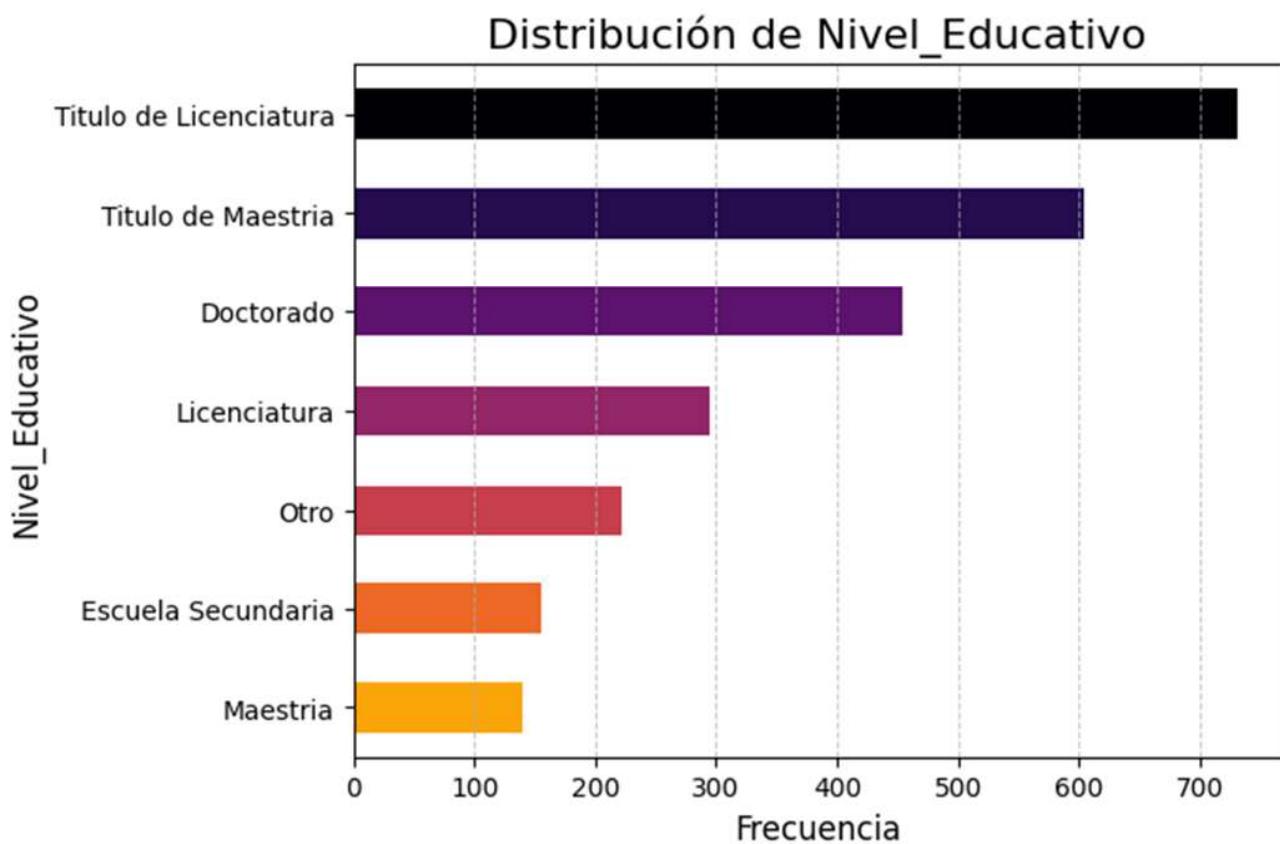
```



```

1 # Gráfico de barras para Nivel_Educativo
2 n_edu = df['Nivel_Educativo'].value_counts().nlargest(10)
3
4
5 # Definir una lista de colores para las barras
6 colors = plt.cm.inferno(np.linspace(0, 0.8, len(n_edu)))
7
8 # Crear el gráfico de barras horizontal con colores múltiples
9 n_edu.plot(kind='barh', color=colors)
10
11 # Personalización del gráfico
12 plt.title('Distribución de Nivel_Educativo', fontsize=16)
13 plt.xlabel('Frecuencia', fontsize=12)
14 plt.ylabel('Nivel_Educativo', fontsize=12)
15
16 # Añadir rejilla
17 plt.grid(True, which='both', axis='x', linestyle='--', linewidth=0.7, alpha=0.7)
18
19 # Invertir el eje Y para que la barra más alta esté en la parte superior
20 plt.gca().invert_yaxis()
21
22 # Mostrar el gráfico
23 plt.show()
24

```



Base de Datos Limpia

Se agrega Base de Datos de Desarrolladores generada después del Proceso de Limpieza.

```
1 # Guardar resultados en un CSV  
2 df_Final.to_csv("Base_FS_limpia_MPAC.csv", index=False)
```

