

Tarea 2 _ Documentación del modelado de datos

Presentado Por: Paula Guarín

Visualización y Storytelling

Base: Ask A Manager Salary Survey 2021

1. Variables en la base de datos original

A continuación se describen las variables originales encontradas en la base de datos. Los nombres se mantienen en inglés.

Variable original (inglés)	Tipo	Descripción
Timestamp	Fecha/hora	Fecha y hora en la que la persona respondió la encuesta.
How old are you?	Texto	Rango de edad reportado por la persona encuestada.
What industry do you work in?	Texto	Industria o sector económico en el que trabaja la persona.
Job title	Texto	Cargo o título del puesto de trabajo.
If your job title needs additional context, please clarify here:	Texto	Información adicional para contextualizar el cargo.
What is your annual salary? (...)	Texto	Salario anual reportado.
How much additional monetary compensation do you get, if any (...)	Numérico	Compensación monetaria adicional anual (bonos, horas extra, etc.).
Please indicate the currency	Texto	Moneda en la que se reporta el salario.
If "Other," please indicate the currency here:	Texto	Detalle de la moneda cuando se selecciona "Other".
If your income needs additional context, please provide it here:	Texto	Comentarios adicionales sobre el ingreso.
What country do you work in?	Texto	País donde trabaja la persona.
If you're in the U.S., what state do you work in?	Texto	Estado dentro de EE. UU. (si aplica).
What city do you work in?	Texto	Ciudad donde trabaja la persona.
How many years of professional work experience do you have overall?	Texto	Años totales de experiencia profesional.
How many years of professional work experience do you have in your field?	Texto	Años de experiencia en el área actual.
What is your highest level of education completed?	Texto	Nivel educativo más alto alcanzado.
What is your gender?	Texto	Género reportado por la persona.
What is your race? (Choose all that apply.)	Texto	Raza o grupo étnico reportado.

2. Variables después del modelado

Durante el modelado, las variables fueron **renombradas al español** y estandarizadas para facilitar el análisis y la visualización. Adicionalmente se ingresaron nuevas variables para poder realizar análisis homogéneos sobre los ingresos totales de las personas encuestadas.

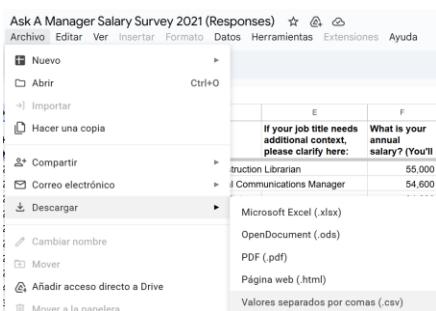
Variable modelada	Tipo	Descripción
fecha_registro	Fecha	Fecha de registro de la respuesta, convertida a tipo fecha.
rango_edad	Texto	Rango de edad estandarizado.
industria	Texto	Industria laboral reportada.
cargo	Texto	Título del cargo.
cargo_contexto	Texto	Información adicional del cargo.
salario_anual	Numérico	Salario anual reportado en la moneda original.
compensacion_adicional	Numérico	Compensación adicional monetaria anual.
moneda	Texto	Moneda principal del salario.
moneda_otro	Texto	Detalle de moneda alternativa.
ingreso_contexto	Texto	Comentarios adicionales sobre el ingreso.
pais	Texto	País original reportado.
pais_estandar	Texto	País estandarizado tras limpieza (homologación de nombres).
estado_usa	Texto	Estado de EE. UU. cuando aplica.
ciudad	Texto	Ciudad original reportada.
ciudad_final	Texto	Ciudad estandarizada y agrupada para análisis.
anios_experiencia_total	Texto	Años totales de experiencia profesional.
anios_experiencia_area	Texto	Años de experiencia en el área.
nivel_educativo	Texto	Nivel educativo original.
educacion_estandar	Texto	Nivel educativo estandarizado.
genero	Texto	Género original reportado.
genero_estandar	Texto	Género estandarizado.
salario_anual_cop	Numérico	Salario anual convertido a pesos colombianos (COP).
compensacion_adicional_cop	Numérico	Compensación adicional convertida a COP.
ingreso_total_cop	Numérico	Suma de salario anual y compensación adicional en COP.

3. Paso a paso para actualizar los datos y aplicar el modelado

Esta sección describe el procedimiento que se debe seguir para replicar el modelado realizado, asumiendo que la estructura de la base original no cambia.

Paso 1. Extracción de datos

- Descargar el archivo original desde el Google Sheets de Ask A Manager Salary Survey 2021.
- Exportar la hoja como archivo .csv.



Paso 2. Carga del archivo en Python

- Importar el archivo CSV usando pandas.
- Validar nombres de columnas originales y tipos de datos.

Análisis Salarios de Gerentes - 2021

Análisis Salarios de Gerentes - 2021														
[1]:	#Importar librerías import pandas as pd import numpy as np													
[4]:	#Cargar la base de datos df = pd.read_csv("AskAManagerSalarySurvey2021.csv") df.head()													
Timestamp	How old are you?	What industry do you work in?	Job title	If your job title needs additional context, please clarify here:	What is your annual salary? (You'll indicate the currency in a later question. If you are part-time or hourly, please enter an annualized equivalent -- what you would earn if you worked the job 40 hours a week, 52 weeks a year.)	How much additional monetary compensation do you get, if any (for example, bonuses or overtime in an average year)? Please only include monetary compensation here, not the value of benefits.	If "Other," please indicate the currency here:	If your income needs additional context, please provide it here:	What country do you work in?	If you're in the U.S., what state do you work in?	What city do you work in?	How many years professional work experience do you have overall?	How many years of professional work experience do you have in your field?	What is your highest level of education completed?
0	4/27/2021 11:02:10	25- 34	Education (Higher Education)	Research and Instruction Librarian	NaN	55,000	0.0	USD	NaN	NaN	United States	Massachusetts	Boston	5-7
1	4/27/2021 11:02:22	25- 34	Computing or Tech	Change & Internal Communications Manager	NaN	54,600	4000.0	GBP	NaN	NaN	United Kingdom	NaN	Cambridge	8 - 1C
2	4/27/2021 11:02:38	25- 34	Accounting Banking & Finance	Marketing Specialist	NaN	34,000	NaN	USD	NaN	NaN	US	Tennessee	Chattanooga	2 - 4
3	4/27/2021 11:02:41	25- 34	Nonprofits	Program Manager	NaN	62,000	3000.0	USD	NaN	NaN	USA	Wisconsin	Milwaukee	8 - 1C
4	4/27/2021 11:02:42	25- 34	Accounting, Banking & Finance	Accounting Manager	NaN	60,000	7000.0	USD	NaN	NaN	US	South Carolina	Greenville	8 - 1C

Análisis preliminar de la base

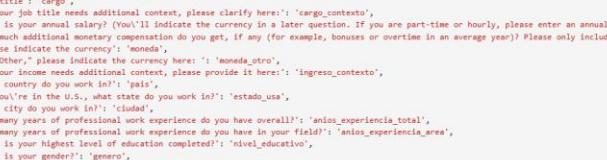
```
[5]: df.dtypes
[6]: df.info()
[6]: 
Timestamp: object
How old are you?: object
What industry do you work in?: object
Job title: object
If your job title needs additional context, please clarify here: object
What is your annual salary? (You'll indicate the currency in a later question. If you are part-time or hourly, please enter an annualized equivalent -- what you would earn if you worked the job 40 hours a week, 52 weeks a year.): object
How much additional monetary compensation do you get, if any (for example, bonuses or overtime in an average year)? Please only include monetary compensation here, not the value of benefits.: float64
Please indicate the currency here: object
If "Other," please indicate the currency here: object
If your income needs additional context, please provide it here: object
What country do you work in?: object
If you're in the U.S., what state do you work in?: object
What city do you work in?: object
How many years of professional work experience do you have overall?: object
How many years of professional work experience do you have in your field?: object
What is your highest level of education completed?: object
What is your gender?: object
What is your race? (Choose all that apply.): object
dtype: object
[8]: #Validar nombre de las columnas originales
df.columns
```

[8]: Index(['Timestamp', 'How old are you?', 'What industry do you work in?', 'Job title', 'If your job title needs additional context, please clarify here.', 'What is your annual salary? (You'll indicate the currency in a later question. If you are part-time or hourly, please enter an annualized equivalent -- what you would earn if you worked the job 40 hours a week, 52 weeks a year.)', 'How much additional monetary compensation do you get, if any (for example, bonuses or overtime in an average year)? Please only include monetary compensation here, not the value of benefits.', 'Please indicate the currency here.', 'If "Other," please indicate the currency here.', 'If your income needs additional context, please provide it here.', 'What country do you work in?', 'If you're in the U.S., what state do you work in?', 'What city do you work in?', 'How many years of professional work experience do you have overall?', 'How many years of professional work experience do you have in your field?', 'What is your highest level of education completed?'], dtype='object')

Paso 3. Renombrar variables

1. Renombrar las columnas originales al español para facilitar el análisis.

```
[10]: #Renombrar nombres de columnas para tener un mejor entendimiento
rename_columns = {
    'Timestamp': 'fecha_registro',
    'How old are you?': 'rango_edad',
    'What industry do you work in?': 'industria',
    'Job title': 'cargo',
    'If your job title needs additional context, please clarify here!': 'cargo_contexto',
    'What is your annual salary? (You'll indicate the currency in a later question. If you are part-time or hourly, please enter an annualized equivalent)': 'salario_anual',
    'How much additional monetary compensation do you get, if any (for example, bonuses or overtime in an average year)? Please only include monetary compensation': 'compensacion_adicional',
    'Please indicate the currency': 'moneda',
    'If "Other", please indicate the currency here!': 'moneda_otro',
    'If your income needs additional context, please provide it here!': 'ingreso_contexto',
    'What country do you work in?': 'pais',
    'If you're in the U.S., what state do you work in?': 'estado_usa',
    'What city do you work in?': 'ciudad',
    'How many years of professional work experience do you have overall?': 'anios_experiencia_total',
    'How many years of professional work experience do you have in your field?': 'anios_experiencia_area',
    'What is your highest level of education completed?': 'nivel_educativo',
    'What is your gender?': 'genero',
    'What is your race? (Choose all that apply.)': 'raza',
    'What does your job involve?': 'objeto_trabajo'
}
df = df.rename(columns=rename_columns)
df.columns
```



```
[10]: Index(['fecha_registro', 'rango_edad', 'industria', 'cargo', 'cargo_contexto',
       'salario_anual', 'compensacion_adicional', 'moneda', 'moneda_otro',
       'ingreso_contexto', 'pais', 'estado_usa', 'ciudad',
       'anios_experiencia_total', 'anios_experiencia_area', 'nivel_educativo',
       'genero', 'raza'],
      dtype='object')
```

```
[11]: #Validar que haya quedado bien la base ajustada
df.head()
```

	fecha_registro	rango_edad	industria	cargo	cargo_contexto	salario_anual	compensacion_adicional	moneda	moneda_otro	ingreso_contexto	pais	
0	4/27/2021 11:02:10	25-34	Education (Higher Education)	Research and Instruction Librarian		NaN	55,000	0.0	USD	NaN	NaN	United States
1	4/27/2021 11:02:22	25-34	Computing or Tech	Change & Internal Communications Manager		NaN	54,600	4000.0	GBP	NaN	NaN	United Kingdom
2	4/27/2021 11:02:38	25-34	Accounting, Banking & Finance	Marketing Specialist		NaN	34,000	NaN	USD	NaN	NaN	US
3	4/27/2021 11:02:41	25-34	Nonprofits	Program Manager		NaN	62,000	3000.0	USD	NaN	NaN	USA
4	4/27/2021 11:02:42	25-34	Accounting, Banking & Finance	Accounting Manager		NaN	60,000	7000.0	USD	NaN	NaN	US

Paso 4. Limpieza y estandarización de variables

- **Variable País (pais):**

- Homologar variaciones de escritura de diferentes países (ej. “US”, “USA”, “United States”) al nombre correcto del país.

▼ Variable país

```

    Valida variable de pais
    """
    df['pais_limpio'] = df['pais'].astype(str).str.strip().str.title()

[16]: df['pais_estandar'] = df['pais_limpio']

[18]: # United States
df.loc[df['pais_limpio'].str.contains(
    'United|Us|America|U.S.|U.S.|U.S.|Us|Usa|U.S.|'
    'United States|United States|United States|United States|'
    'Virginia|Us|United States|United States|'
    'United States|United States|U.S.|United States|'
    'United States|California|United States Of America|United States|United States|'
    'United States|United States|United States|United States|United States|'
    'San Francisco|us|Hartford|us|U.S.|'
    'na=False
), 'pais_estandar'] = 'United States'

# United Kingdom
df.loc[df['pais_limpio'].str.contains(
    'United Kingdom|England|Scotland|Wales|Northern Ireland|UK|U.K.|United Kingdom|UK|'
    'Englang|Britain|London|Englang|Britain',
    na=False
), 'pais_estandar'] = 'United Kingdom'

# Canada
df.loc[df['pais_limpio'].str.contains('Canda|Canada|Canad|Can|Canada|Canad|Csanda', na=False), 'pais_estandar'] = 'Canada'

# Mexico
df.loc[df['pais_limpio'].str.contains('Mexico|México', na=False), 'pais_estandar'] = 'Mexico'

# Italy
df.loc[df['pais_limpio'].str.contains('Italy \\(South\\)|Italia', na=False), 'pais_estandar'] = 'Italy'

# New Zealand
df.loc[df['pais_limpio'].str.contains('Nz|NI|Aotearoa New Zealand|New Zealand Aotearoa', na=False), 'pais_estandar'] = 'New Zealand'

# Netherlands
df.loc[df['pais_limpio'].str.contains('The Netherlands|Nederland', na=False), 'pais_estandar'] = 'Netherlands'

# Australia
df.loc[df['pais_limpio'].str.contains('Australia|Australian', na=False), 'pais_estandar'] = 'Australia'

# India
df.loc[df['pais_limpio'].str.contains('Ibdia', na=False), 'pais_estandar'] = 'India'

# Bosnia
df.loc[df['pais_limpio'].str.contains('Bosnia And Herzegovina', na=False), 'pais_estandar'] = 'Bosnia'

# Panama
df.loc[df['pais_limpio'].str.contains('Panama', na=False), 'pais_estandar'] = 'Panama'

# Ukraine
df.loc[df['pais_limpio'].str.contains('U.A.|Ua|Ucrania', na=False), 'pais_estandar'] = 'Ukraine'

# United Arab Emirates
df.loc[df['pais_limpio'].str.contains('Uae', na=False), 'pais_estandar'] = 'United Arab Emirates'

# China
df.loc[df['pais_limpio'].str.contains('Mainland China|Hong Konh|Hong Kongleng|Hong Kong', na=False), 'pais_estandar'] = 'China'

# Austria
df.loc[df['pais_limpio'].str.contains('Austria, But I Work Remotely For A Dutch|British Company', na=False), 'pais_estandar'] = 'Austria'

# Argentina
df.loc[df['pais_limpio'].str.contains('Argentinian But My Org Is In Thailand', na=False), 'pais_estandar'] = 'Argentina'

# Pakistan
df.loc[df['pais_limpio'].str.contains('Company In German. I Work From Pakistan', na=False), 'pais_estandar'] = 'Pakistan'

# Luxembourg
df.loc[df['pais_limpio'].str.contains('Luxemburg', na=False), 'pais_estandar'] = 'Luxembourg'

# Denmark
df.loc[df['pais_limpio'].str.contains('Denmark', na=False), 'pais_estandar'] = 'Denmark'

# Brazil
df.loc[df['pais_limpio'].str.contains('Brasil', na=False), 'pais_estandar'] = 'Brazil'

```

- Identificar textos largos o respuestas inválidas y marcarlas como “No aplica”.

```
[20]: Patron_ruido = (
    'Remote|Wherever|Worldwide|Bonus|Commission|Salary|'
    'Finance|Policy|Contract|N/A|No$|Work|Raises|Globally|Projects|'
    'Overseas|Contracts|Sales|Company|Y|Uxz|Ss|NI|International|Global|#f|'
    'Dbfem|j|'
)
df.loc[
    (df['pais_limpio'].str.len() > 40) |
    (df['pais_limpio'].str.contains(patron_ruido, na=False)),
] = 'pais_estandar'
] = 'No aplica'
```

- Por último, verificar ajustes realizados en la variable final país_estandar.

```
[22]: df['pais_estandar'].value_counts().head(20)
[22]: pais_estandar
United States      23983
Canada            3691
United Kingdom     1599
Australia          392
Germany           198
New Zealand        131
Ireland             125
Netherlands         98
France              69
Spain                50
Sweden              41
Switzerland          38
No aplica           37
Belgium              35
Japan                29
India                26
Denmark              24
Singapore             20
South Africa          19
Italy                 16
Name: count, dtype: int64
[24]: len(df['pais_estandar'].unique())
[24]: 101
[26]: df[df['pais_estandar'] == 'No aplica'].shape
[26]: (37, 20)
```

• Variable Ciudad (ciudad):

- Normalizar mayúsculas/minúsculas y unificar variaciones comunes (ej. “NY”, “New York City”, “NY Suburb”).

Variable ciudad

```
[28]: """
Validar variable de ciudad
"""
df['ciudad limpia'] = df['ciudad'].astype(str).str.strip().str.title()
df['ciudad estandar'] = df['ciudad limpia']

[30]: #Washington
df.loc[
    df['ciudad limpia'].str.contains(
        'Washington',
        case=False,
        na=False
    ),
    'ciudad estandar'
] = 'Washington'

# NY
df.loc[
    df['ciudad limpia'].str.contains(
        'New York Ny',
        case=False,
        na=False
    ),
    'ciudad estandar'
] = 'New York'
```

- Agrupar ciudades con muy baja frecuencia como “No especificada”.

```
[34]: df.loc[
    df['ciudad limpia'].str.contains(
        'Remote|Work From Home|Worldwide|Rural|Declined|N/A|Area',
        case=False,
        na=False
    ),
    'ciudad estandar'
] = 'No especificada'
```

- Definir la frecuencia de cada valor de ciudad, y se definió un umbral, y con esto las ciudades con frecuencia menor al umbral se categorizaron bajo “Otras ciudades” y las que sobrepasan el umbral conservan su nombre.

```
[38]: UMBRAL = 100
df['ciudad_final'] = df['ciudad_estandar'].where(
    df['ciudad_estandar'].map(df['ciudad_estandar'].value_counts()) >= UMBRAL,
    'Otras ciudades'
)
```

- Por último, verificar ajustes realizados en la variable final ciudad_final.

```
[40]: df['ciudad_final'].value_counts().head(20)
```

```
[40]: ciudad_final
Otras ciudades    13607
New York          1385
Washington        884
Chicago           834
Boston            832
Seattle           743
No especificada   661
London            626
San Francisco     590
Los Angeles       540
Portland          487
Toronto           444
Minneapolis      389
Atlanta           362
Austin             356
Philadelphia      351
Denver             335
Houston           279
Dallas             242
Pittsburgh         241
Name: count, dtype: int64
```

Nota: Debido a la alta dispersión del campo ciudad, se conservaron únicamente las ciudades con mayor frecuencia y se agruparon las restantes en categorías genéricas para facilitar la visualización sin perder información relevante.

- **Variable Género (genero):**

- Unificar respuestas similares y agrupar valores nulos o ambiguos en “Otro / No especificado”.

```
[44]: # 1. Crear versión Limpia
df['genero_limpio'] = (
    df['genero']
    .astype(str)
    .str.strip()
    .str.title()
)

# 2. Crear género estandar
df['genero_estandar'] = 'Otro / No especificado'

# Mujer
df.loc[df['genero_limpio'] == 'Woman', 'genero_estandar'] = 'Mujer'

# Hombre
df.loc[df['genero_limpio'] == 'Man', 'genero_estandar'] = 'Hombre'

# No binario
df.loc[df['genero_limpio'] == 'Non-Binary', 'genero_estandar'] = 'No binario'

# Preferir no responder (incluye variantes)
df.loc[
    df['genero_limpio'].str.contains(
        'Prefer', case=False, na=False
    ),
    'genero_estandar'
] = 'Otro / No especificado'

# NaN explicitos
df.loc[df['genero_limpio'] == 'Nan', 'genero_estandar'] = 'Otro / No especificado'

# 3. Validación final
df['genero_estandar'].value_counts()
```



```
[44]: genero_estandar
Mujer                21424
Hombre               5538
No binario            747
Otro / No especificado 481
Name: count, dtype: int64
```

- **Nivel educativo (nivel_educativo):**

- Homologar niveles educativos similares y crear una variable estandarizada (educacion_estandar).

```
[48]: # 1. Crear versión limpia
df['educacion_limpia'] = (
    df['nivel_educativo']
    .astype(str)
    .str.strip()
    .str.title()
)

# 2. Crear variable estandar con valor por defecto
df['educacion_estandar'] = 'Otro / No especificado'

# Secundaria
df.loc[
    df['educacion_limpia'].str.contains('High School', na=False),
    'educacion_estandar'
] = 'Secundaria'

# Educación técnica / incompleta
df.loc[
    df['educacion_limpia'].str.contains('Some College', na=False),
    'educacion_estandar'
] = 'Educación técnica / incompleta'

# Pregrado
df.loc[
    df['educacion_limpia'].str.contains('College Degree', na=False),
    'educacion_estandar'
] = 'Pregrado'

# Maestría
df.loc[
    df['educacion_limpia'].str.contains('Master', na=False),
    'educacion_estandar'
] = 'Maestría'

# Doctorado
df.loc[
    df['educacion_limpia'].str.contains('Phd', case=False, na=False),
    'educacion_estandar'
] = 'Doctorado'

# Grado profesional (MD, JD, etc.)
df.loc[
    df['educacion_limpia'].str.contains('Professional Degree', na=False),
    'educacion_estandar'
] = 'Grado profesional'
df['educacion_estandar'].value_counts()

[48]: educacion_estandar
Pregrado           13557
Maestría          8892
Educación técnica / incompleta   3096
Doctorado          1438
Grado profesional  1328
Secundaria          657
Otro / No especificado  236
Name: count, dtype: int64
```

Paso 5. Exportación

- Por último, se descarga la base de python estandarizada en formato .csv y se carga en LookerStudio.

Descargar base ajustada para trabajar poder cargar en LookerStudio

```
[60]: df.to_csv("base_salarios_final.csv", index=False)
```

The screenshot shows the Looker Studio interface with the following details:

- Header:** Tarea_2, Archivo, Editar, Vista, Insertar, Página, Organizar, Recurso, Ayuda, Compartir.
- Toolbar:** Restablecer, Compartir, Añadir página, Añadir datos, Combinar, Añadir un gráfico, Añadir un control.
- Message Bar:** 1,99 GB disponibles de 2 GB.
- CSV Upload Panel:**
 - Subida de archivos CSV De Google
 - Puedes importar datos a Looker Studio subiendo archivos CSV (valores separados por comas) desde tu ordenador. Se pueden combinar varios archivos CSV siempre que los nombres sean diferentes.
 - Más información Notificar un problema
- Data Management:**
 - Conjuntos de datos: Salarios, Tarea_1.
 - Archivos: 8 feb 2026, 13:46, 1 archivos (7 MB/100 MB), base_salarios_final.
 - Esquema: fecha_registro, rango_edad, industria.

Paso 6. Conversión monetaria

- Variable Moneda (moneda):** validar posibles valores para poder realizar la búsqueda y conversión de acuerdo con TRM.

```
[54]: ##Moneda
df["moneda"].value_counts(dropna=False)

[54]: moneda
USD      23473
CAD      1676
GBP      1594
EUA      654
AUD/NZD  505
Other     170
CHF      37
SEK      37
JPY      23
ZAR      17
HKD      4
Name: count, dtype: int64
```

- Validamos que todas las variables relacionadas con dinero estén efectivamente como tipo de dato numérico.

salario_anual	123	Número	Total
---------------	-----	--------	-------

compensacion_adicional	123	Número	Ninguna
------------------------	-----	--------	---------

- Convertir salario_anual y compensacion_adicional a COP usando la TRM del día de la elaboración de la base de datos estandarizada. Directamente en LookerStudio, seleccionamos la opción añadir un campo. Verificar que las variables creadas sean tipo numérico.

- salario_anual_cop

Nombre del campo por ejemplo, Nuevo campo calculado salario_anual_cop	ID de campo ID de campo calc_nu324v9q0d
---	--

Fórmula

```

1 CASE
2 WHEN moneda = "USD" THEN salario_anual * 3668
3 WHEN moneda = "CAD" THEN salario_anual * 2682
4 WHEN moneda = "GBP" THEN salario_anual * 4993
5 WHEN moneda = "EUR" THEN salario_anual * 4335
6 WHEN moneda = "AUD/NZD" THEN salario_anual * 2572
7 WHEN moneda = "CHF" THEN salario_anual * 4728
8 WHEN moneda = "SEK" THEN salario_anual * 406
9 WHEN moneda = "JPY" THEN salario_anual * 23
10 WHEN moneda = "ZAR" THEN salario_anual * 228
11 WHEN moneda = "HKD" THEN salario_anual * 469
12 ELSE salario_anual
13 END

```

DAR FORMATO A FÓRMULA

- compensacion_adicional_cop

Nombre del campo por ejemplo, Nuevo campo calculado compensaciones_cop	ID de campo ID de campo calc_6b7guw9q0d
--	--

Fórmula

```

1 CASE
2 WHEN moneda = "USD" THEN compensacion_adicional * 3668
3 WHEN moneda = "CAD" THEN compensacion_adicional * 2682
4 WHEN moneda = "GBP" THEN compensacion_adicional * 4993
5 WHEN moneda = "EUR" THEN compensacion_adicional * 4335
6 WHEN moneda = "AUD/NZD" THEN compensacion_adicional * 2572
7 WHEN moneda = "CHF" THEN compensacion_adicional * 4728
8 WHEN moneda = "SEK" THEN compensacion_adicional * 406
9 WHEN moneda = "JPY" THEN compensacion_adicional * 23
10 WHEN moneda = "ZAR" THEN compensacion_adicional * 228
11 WHEN moneda = "HKD" THEN compensacion_adicional * 469
12 ELSE compensacion_adicional
13 END

```

DAR FORMATO A FÓRMULA

NOTA: Las tasas de cambio fueron obtenidas de la plataforma XE Currency Exchange, utilizando los valores vigentes en la fecha de realización del ejercicio.

- Crear ingreso_total_cop como la suma de salario y compensación adicional en COP.

Nombre del campo por ejemplo, Nuevo campo calculado ingreso_total_cop	ID de campo ID de campo calc_90lmx9q0d
---	---

Fórmula

```

1 salario_anual_cop + compensaciones_cop

```

DAR FORMATO A FÓRMULA

Paso 7. Ajustar fecha

- Debido a que la variable fecha tiene problemas en su exportación y queda guardada como tipo Texto en LookerStudio, se procede a convertir el tipo a datetime. Verificar que tipo sea Fecha.

Nombre del campo por ejemplo, Nuevo campo calculado fecha_registro_limpia	ID de campo ID de campo calc_4ipetu9q0d
---	--

Fórmula

```

1 DATE(
2 PARSE_DATETIME("%m/%d/%Y %H:%M:%S", fecha_registro )
3 )

```

DAR FORMATO A FÓRMULA