CIENCIA DE DATOS EN PYTHON

PROYECTO FINAL

BALMORE hERNÁNDEZ BERNAL
ANDREA HERNÁNDEZ MARROQUÍN

Estructura de Nuestro Proyecto

Consta de 4 Archivos

1. Diseño Transaccional y carga datos.ipynb
2. Diseño Dimensional.ipynb
3. ETL_dimensional con inserts a el DW.ipynb
4. Respuestas_Preguntas_del_Negocio.ipynb

1. Diseño Transaccional y carga datos.ipynb

```python
aws_rds_conn = boto3.client('rds', aws_access_key_id = config.get('IAM','ACCESS_KEY'),
                            aws_secret_access_key = config.get('IAM','SECRET_KEY'),
                            region_name = 'us-east-2')
```

```python
rds_intances_ids = []

aws_response = aws_rds_conn.describe_db_instances() # retorna un objeto [diccionario] iterable de las instancias

for response in aws_response['DBInstances']:
    rds_intances_ids.append(response['DBInstanceIdentifier'])

print(f'Instancias disponibles: {rds_intances_ids}')
```

```
Instancias disponibles: ['dbdim', 'dbtienda', 'sakila-db-pg-v']
```

```python
try:
    response = aws_rds_conn.create_db_instance(
                            DBInstanceIdentifier=config.get('DBTIENDA','DB_INSTANCE_ID'),
                            DBName=config.get('DBTIENDA','DB_NAME'),
                            MasterUsername=config.get('DBTIENDA','DB_USERNAME'),
                            MasterUserPassword=config.get('DBTIENDA','DB_PASSWORD'),
                            Port=int(config.get('DBTIENDA','DB_PORT')),
                            DBInstanceClass='db.t3.micro',
                            Engine=config.get('DBTIENDA','DB_ENGINE'),
                            PubliclyAccessible=True,
                            AllocatedStorage=20,
                            VpcSecurityGroupIds=[config.get('VPC','SECURITY_GROUP')],
                        )
```

Access point

```python
try:
    instance = aws_rds_conn.describe_db_instances(DBInstanceIdentifier=config.get('DBTIENDA','DB_INSTANCE_ID'))
    RDS_HOSTNAME = instance.get('DBInstances')[0].get('Endpoint').get('Address')
    print(RDS_HOSTNAME)
except Exception as ex:
    print('Error!!!!',ex)
```

```
dbtienda.cp6geq8ycm59.us-east-2.rds.amazonaws.com
```

Conexion a BD transaccional en mysql y creacion de tablas

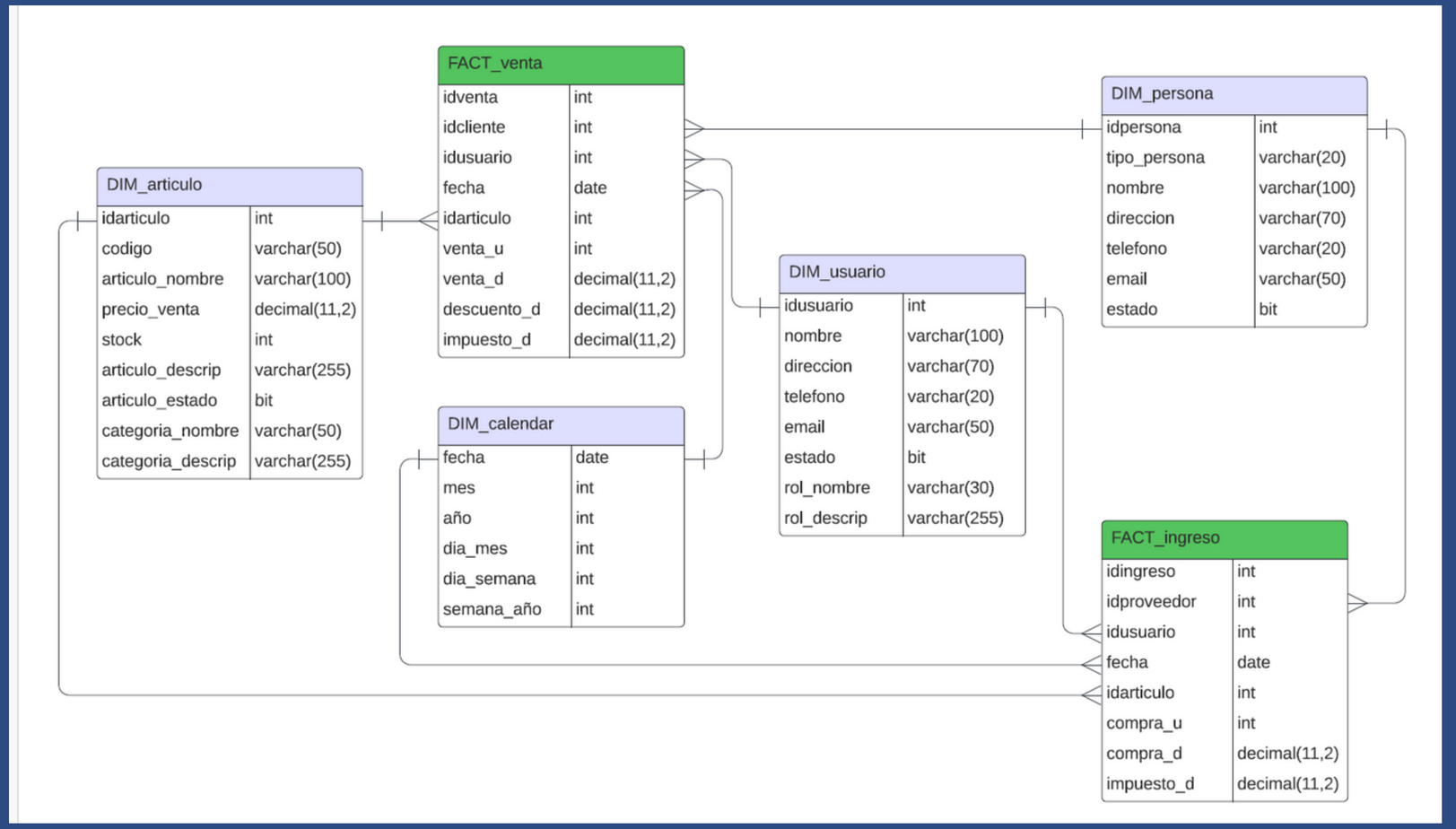
```python
import ddl_transacdb # py donde se encuentra el ddl
#ddl_transdb.ddl
```

```python
# TABLA ARTICULO
for _ in range(100):  #Insertamos 100 datos
    idcategoria = random.randint(1, 6)  # Le colocamos hasta el 5 porque solo colocamos 5 categorias
    codigo = faker.uuid4()[:6]  # Generamos un codigo uuid
    nombre_articulo = faker.word()  # Le colocamos nombre
    precio_venta = round(random.uniform(100, 1000), 2)  # Le colocamos un precio entre 100 y mil que tenga 2 decimales como maximo
    stock = random.randint(1, 100)
    descripcion_articulo = faker.text()  # Le colocamos la descripcion del articulo
    imagen = faker.file_name(category='image', extension='jpg')  # Agregamos una imagen con extension jpg
    estado_articulo = random.choice([0, 1])  # Estado aleatorio (0 o 1)

    # Insertamos los datos a la base de datos
    sql_articulo = "INSERT INTO articulo (idcategoria, codigo, nombre, precio_venta, stock, descripcion, imagen, estado) VALUES (%s, %s, %s, %s, %s, %s, %s
    val_articulo = (idcategoria, codigo, nombre_articulo, precio_venta, stock, descripcion_articulo, imagen, estado_articulo)

    # Ejecutar la consulta para la tabla "articulo"
    cursor.execute(sql_articulo, val_articulo)
```

Python


FACT_venta
idventa int
idcliente int
idusuario int
fecha date
idarticulo int
venta_u int
venta_d decimal(11,2)
descuento_d decimal(11,2)
impuesto_d decimal(11,2)
DIM_persona
idpersona int
tipo_persona varchar(20)
nombre varchar(100)
direccion varchar(70)
telefono varchar(20)
email varchar(50)
estado bit
DIM_articulo
idarticulo int
codigo varchar(50)
articulo_nombre varchar(100)
precio_venta decimal(11,2)
stock int
articulo_descrip varchar(255)
articulo_estado bit
categoria_nombre varchar(50)
categoria_descrip varchar(255)
DIM_usuario
idusuario int
nombre varchar(100)
direccion varchar(70)
telefono varchar(20)
email varchar(50)
estado bit
rol_nombre varchar(30)
rol_descrip varchar(255)
DIM_calendar
fecha date
mes int
año int
dia_mes int
dia_semana int
semana_año int
FACT_ingreso
idingreso int
idproveedor int
idusuario int
fecha date
idarticulo int
compra_u int
compra_d decimal(11,2)
impuesto_d decimal(11,2)

2. Diseño Dimensional.ipynb

Access point dimensional

```
try:
    instance = aws_rds_conn.describe_db_instances(DBInstanceIdentifier=config.get('DIM','DB_INSTANCE_ID'))
    RDS_HOST_DBDIM = instance.get('DBInstances')[0].get('Endpoint').get('Address')
    print(RDS_HOST_DBDIM)
except Exception as ex:
    print('Error!!!!',ex)
```

```
dbdim.cp6geq8ycm59.us-east-2.rds.amazonaws.com
```

Conexion a DB y creacion de tablas

```
import ddl_dimdb # py donde se encuentra el ddl
```

```
try:
    db_pg_conn = psycopg2.connect(
                        database=config.get('DIM','DB_NAME'),
                        user=config.get('DIM','DB_USERNAME'),
                        password=config.get('DIM','DB_PASSWORD'),
                        port=config.get('DIM','DB_PORT'),
                        host=RDS_HOSTNAME
                        )
    cursor = db_pg_conn.cursor()
    cursor.execute(ddl_dimdb.ddl)
    db_pg_conn.commit()
except Exception as ex:
```

3. ETL_dimensional con inserts a el DW.ipynb

Transformar tablas dimensionales e insersion de datos

+ Code + Markdown

Dimension Persona

```
#Seleccionamos los campos que necesitamos para la dimension
dim_persona = df_persona.loc[:, ["idpersona","tipo_persona","nombre","direccion","telefono","email"]]
#Le cambiamos los datos como aparecen en la tabla dimensional
dim_persona = dim_persona.rename(columns={'nombre': 'nombre_p', 'direccion': 'direccion_p', 'telefono': 'telefono_p', 'email': 'email_p'
#insertamos datos a dim_persona.
dim_persona
#dim_persona = dim_persona.to_sql('dim_persona', postgres_driver, index=False, if_exists='append')
```

	idpersona	tipo_persona	nombre_p	direccion_p	telefono_p	email_p
0	1	PERSONA	Ashley Long	731 Hall Rest Apt. 345\nNew Stephanie, TX 45230	+1-612-213-0569x885	heatherbass@example.org
1	2	PERSONA	Joseph Floyd	827 Harris Squares\nSouth Michael, OH 15445	711.625.8479	ydavis@example.net
2	3	EMPRESA	Matthew Brown	Unit 5986 Box 8060\nDPO AE 31584	293.723.3014	taraholmes@example.org
3	4	EMPRESA	Scott Mayer	824 Medina Avenue Suite 336\nPort Terri, NH 27174	001-763-708-7537	jimenezalicia@example.net
4	5	EMPRESA	Cassandra Torres	32549 Mendoza Extension Apt. 991\nEast Timothy...	5579496223	melissapeters@example.com
5	6	EMPRESA	Michael Cox	01919 Richard Common Suite 092\nSouth Anna, OR...	412.396.3173x3157	donnakelley@example.org
6	7	EMPRESA	Kathy Perez	598 Michael Forges Apt. 030\nWest Christine, N...	712.735.8251	wbrown@example.org
7	8	PERSONA	Margaret Goodwin	PSC 8756, Box 2910\nAPO AE 34076	001-856-756-5240x009	wilsonjesse@example.net
8	9	EMPRESA	Jennifer Daugherty	9702 Ingram Curve\nLake Jeffery, CT 95522	(647)397-5018x3727	johnbrown@example.net
9	10	PERSONA	Samantha Ramirez	823 Brown Fords\nCollinsberg, MH 62101	+1-363-212-6561x8889	mark54@example.com

Dimension Usuario

```python
dim_usuario['estado_u'].dtypes
```

Python

```
dtype('int64')
```

```python
#Seleccionamos los campso que necesitamos para la dimension
dim_usuario = df_usuario.loc[:, ["idusuario","nombre","direccion","telefono","email","idrol","estado"]]

#Le cambiamos el nombre a los campos que tienen el nombre diferente
dim_usuario = dim_usuario.rename(columns={'nombre': 'nombre_u', 'direccion' : 'direccion_u','telefono':'telefono_u','email':'email_u','estado':'estado_u'})

#Seleccionamos los campso que necesitamos para la dimension
dim_rol = df_rol.loc[:, ["idrol","nombre","descripcion"]]
#Le cambiamos el nombre a los campos que tienen el nombre diferente
dim_rol = dim_rol.rename(columns={'nombre': 'rol_nombre', 'descripcion' : 'rol_descrip'})

join_usuario_rol = pd.merge(dim_usuario, dim_rol, left_on='idrol', right_on='idrol', how='inner').drop_duplicates() # Eliminamos Duplicados
#Ordenamos los campos
dim_usuario = join_usuario_rol.loc[:, ["idusuario","nombre_u","direccion_u","telefono_u","email_u","rol_nombre","rol_descrip"]]
#dim_usuario['estado_u'] = dim_usuario['estado_u'].astype('bool')
dim_usuario
```

4. Respuestas_Preguntas_del_Negocio.ipynb

¿Cuál es la persona que más ha comprado?

Samantha Ramirez

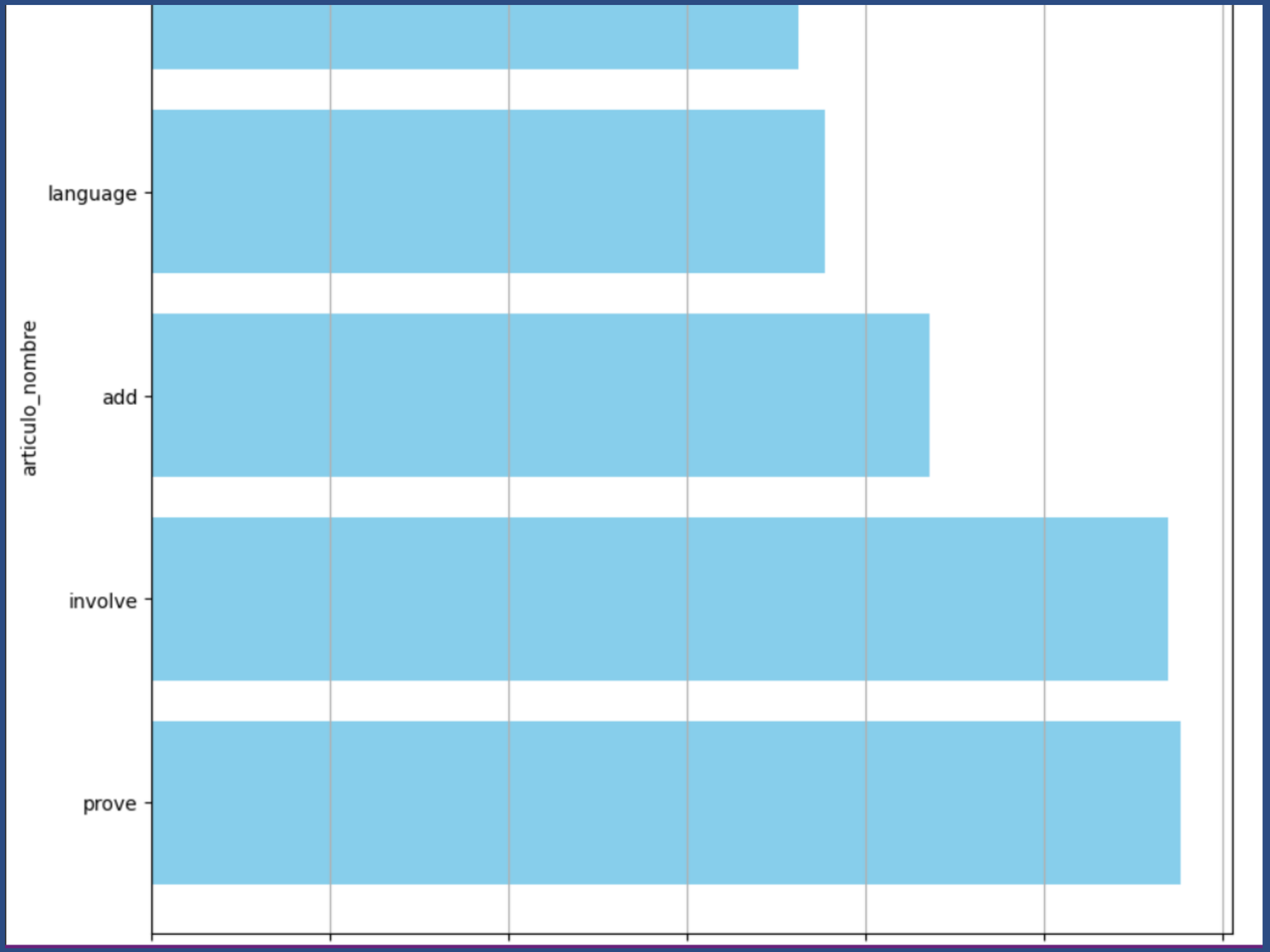
```python
join_FACT_venta_dim_persona = pd.merge(df_FACT_venta, df_dim_persona, left_on='idcliente', right_on='idpersona', how='inner').drop_duplicates() # Eliminamo
pregunta_2= join_FACT_venta_dim_persona.groupby('nombre_p')['venta_d'].sum().reset_index()
# Ordenar de mayor a menor
pregunta_2 = pregunta_2.sort_values(by='venta_d', ascending=False)
pregunta_2 = pregunta_2.head(10)
```

Python

```python
display(pregunta_2)
```

Python

	nombre_p	venta_d
8	Samantha Ramirez	190396.30
2	Jennifer Daugherty	182366.90
1	Cassandra Torres	178075.10
4	Kathy Perez	168078.03
7	Michael Cox	140611.37
5	Margaret Goodwin	125581.82
3	Joseph Floyd	120635.09
9	Scott Mayer	109964.56
6	Matthew Brown	109882.56


articulo_nombre
language
add
involve
prove

¿Qué categoría tiene más ventas?

Ropa

```
[27]
1  pregunta_3= join_FACT_venta_dim_articulo.groupby('categoria_nombre')['venta_d'].sum().reset_index()
2  # Ordenar de mayor a menor
3  pregunta_3 = pregunta_3.sort_values(by='venta_d', ascending=False)
4  pregunta_3 = pregunta_3.head(10)
```

```
[28]
1  display(pregunta_3)
```

...

	categoria_nombre	venta_d
5	ROPA	335650.56
3	HOGAR	273236.57
0	ALIMENTOS	246790.17
4	JUGUETES	243904.96
2	DEPORTES	175287.38
1	CALZADO	112887.24

¿Cual es el rol de usuario que mas transacciones ha hecho?

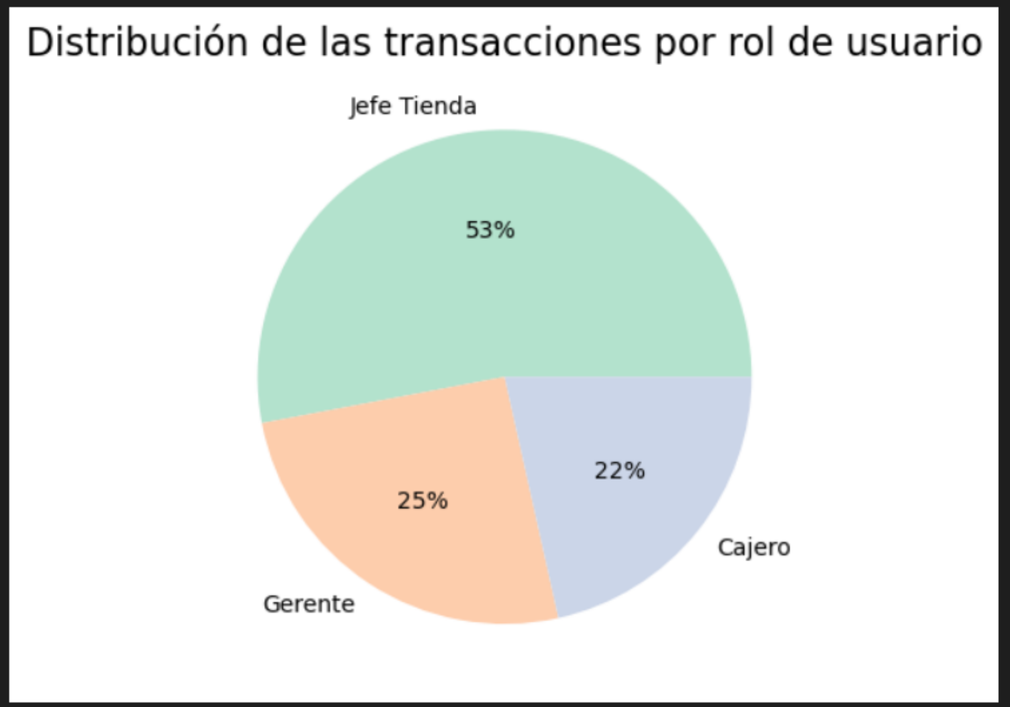
Jefe Tienda

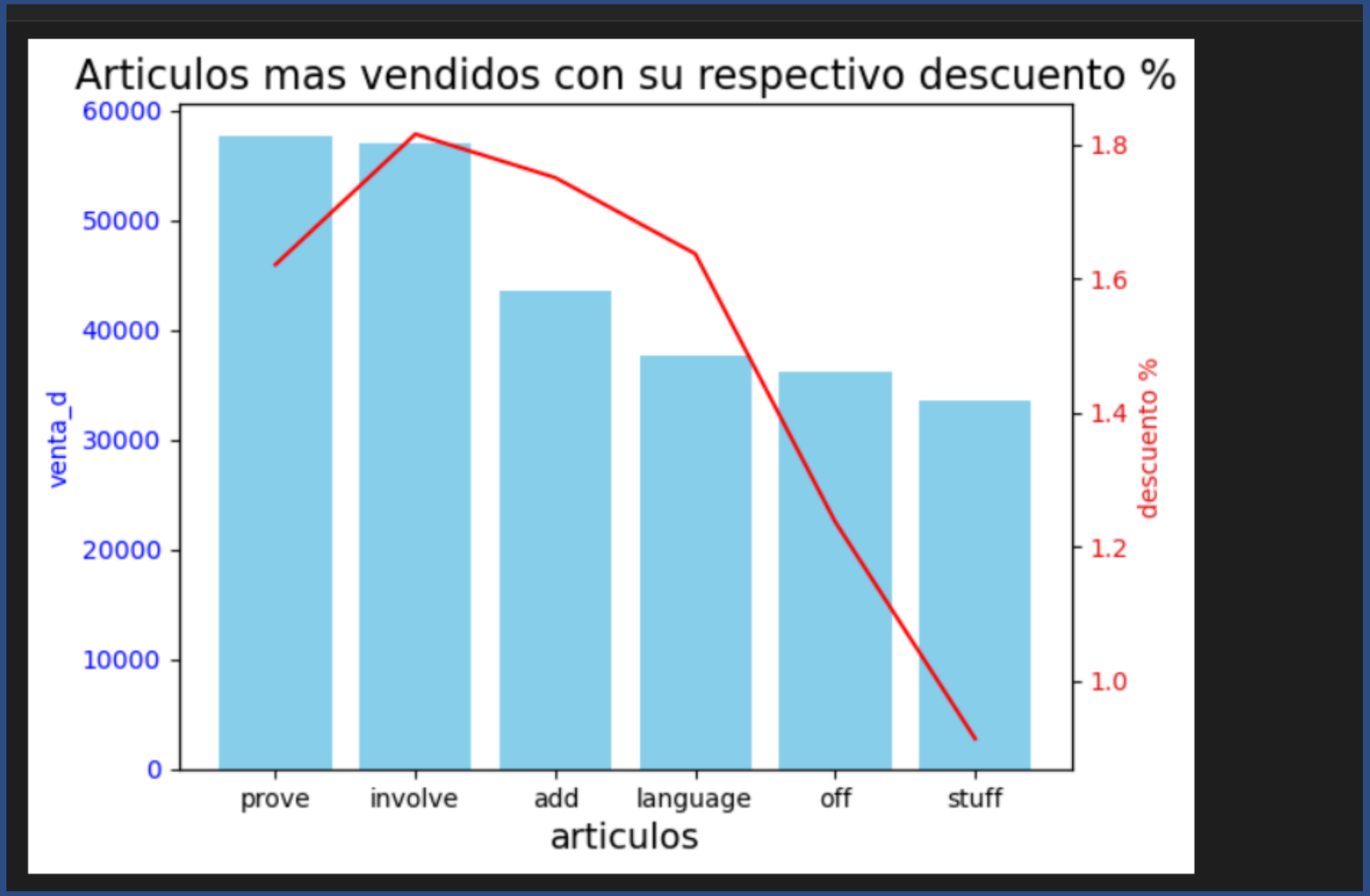
```python
# sumar los registros unicos de venta e ingresos de cada usuario y luego agregar el rol del usuario a la tabla
pregunta_4 = (df_FACT_ingreso.groupby(by='idusuario')['idingreso'].nunique() + \
              df_FACT_venta.groupby(by='idusuario')['idventa'].nunique()).reset_index().\
              rename({0:'Transacciones'}, axis=1).\
              merge(df_dim_usuario[['idusuario','rol_nombre']], on='idusuario', how='left')
# agrupar por roles y ordenar de mayor a menor
pregunta_4 = pregunta_4.groupby(by='rol_nombre', as_index=False)[['Transacciones']].sum().\
                    sort_values('Transacciones', ascending=False)
pregunta_4
```

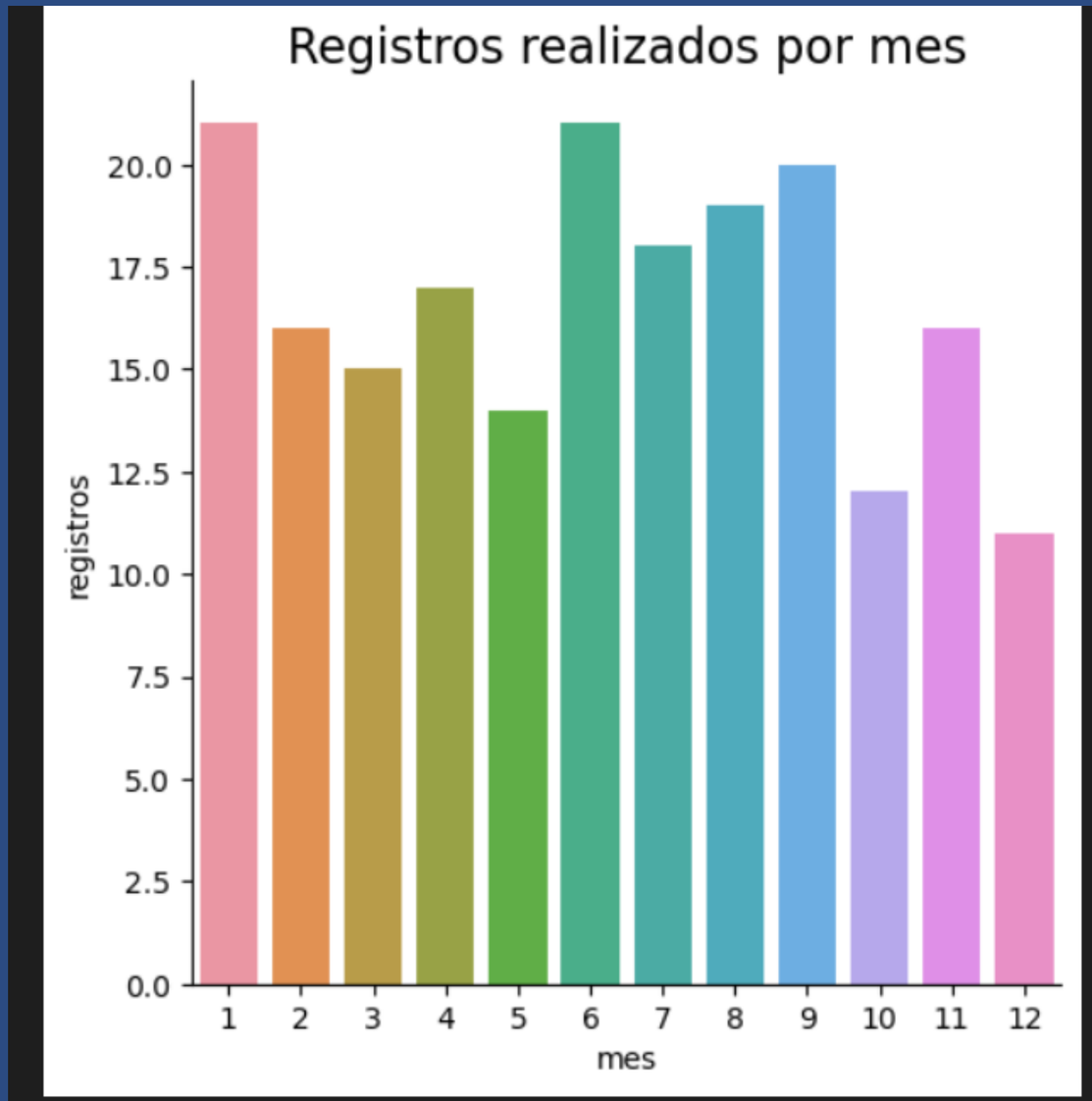
	rol_nombre	Transacciones
2	Jefe Tienda	106
1	Gerente	51
0	Cajero	43

```python
palette_color = sns.color_palette('Pastel2')
plt.pie(pregunta_4['Transacciones'], labels=pregunta_4['rol_nombre'], colors=palette_color, autopct='%.0f%%')
plt.title('Distribución de las transacciones por rol de usuario', size=16)
```

```
Text(0.5, 1.0, 'Distribución de las transacciones por rol de usuario')
```




Articulos mas vendidos con su respectivo descuento %
60000
50000
40000
30000
20000
10000
0
venta_d
1.8
1.6
1.4
1.2
1.0
descuento %
prove
involve
add
language
off
stuff
articulos


Registros realizados por mes
registros
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
1
2
3
4
5
6
7
8
9
10
11
12
mes