MÓDULO

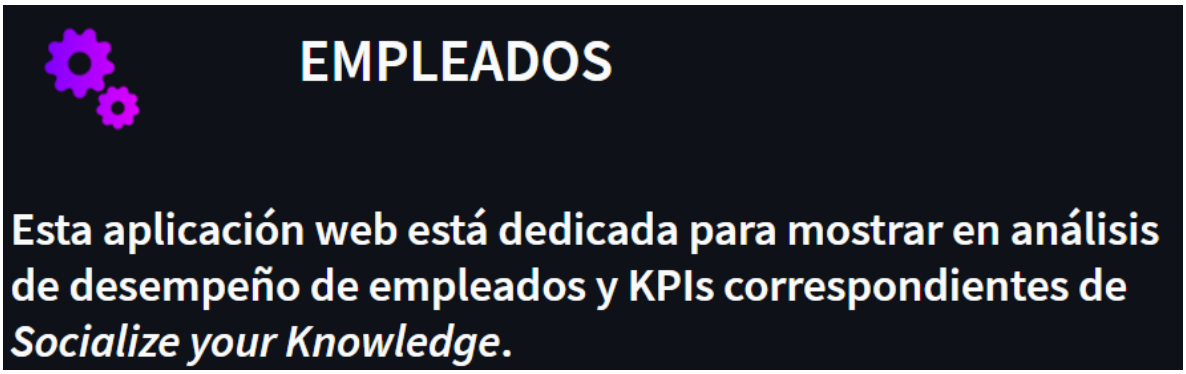
Aplicación web de ciencia de datos

NOMBRE

Genaro Salgado Alarcón

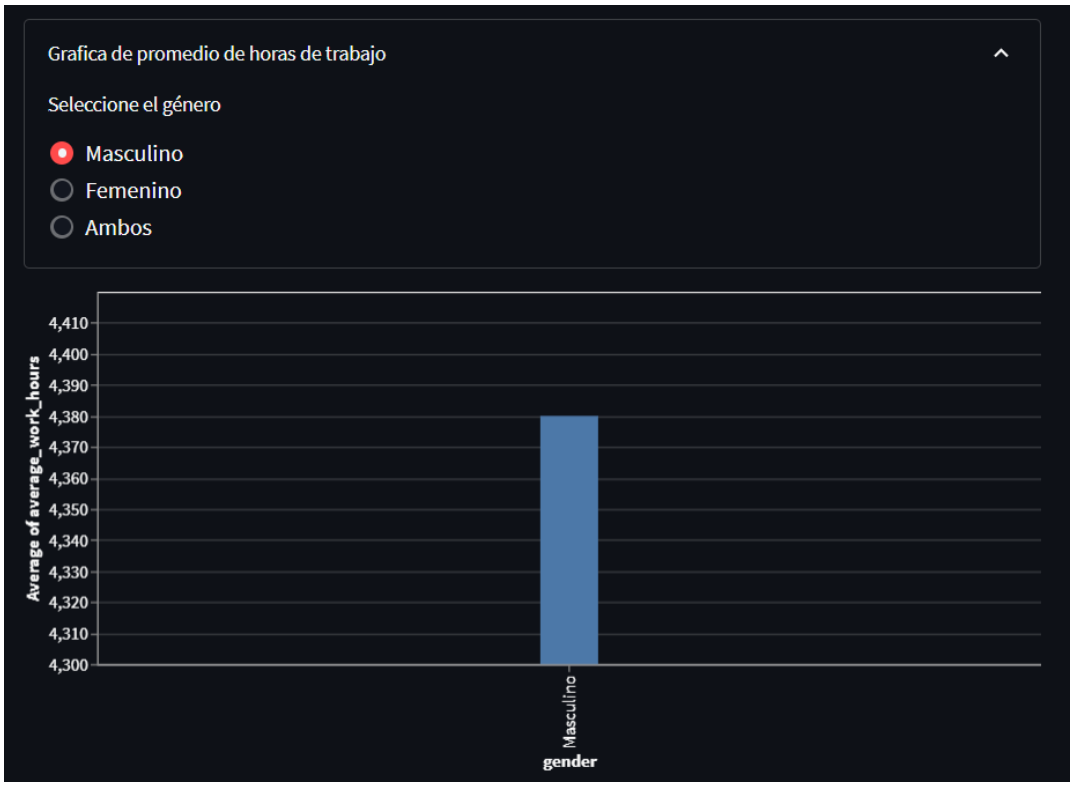
EVALUACIÓN FINAL

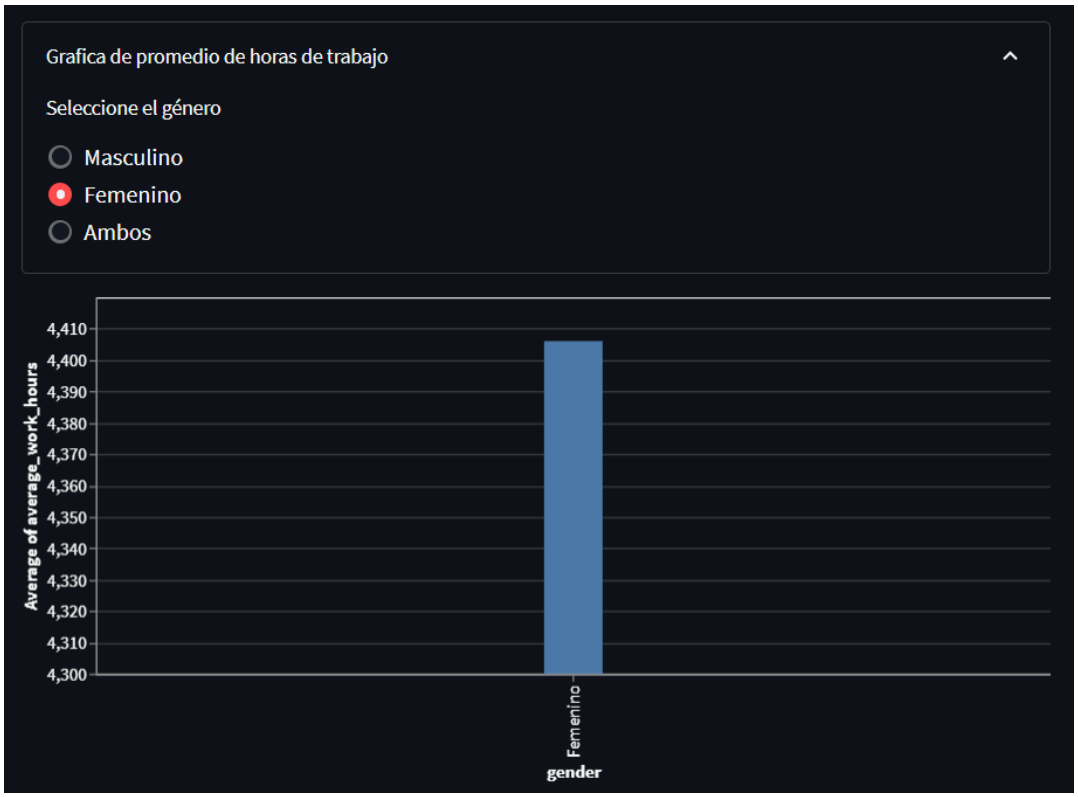
Agregando el título, logo y descripción:

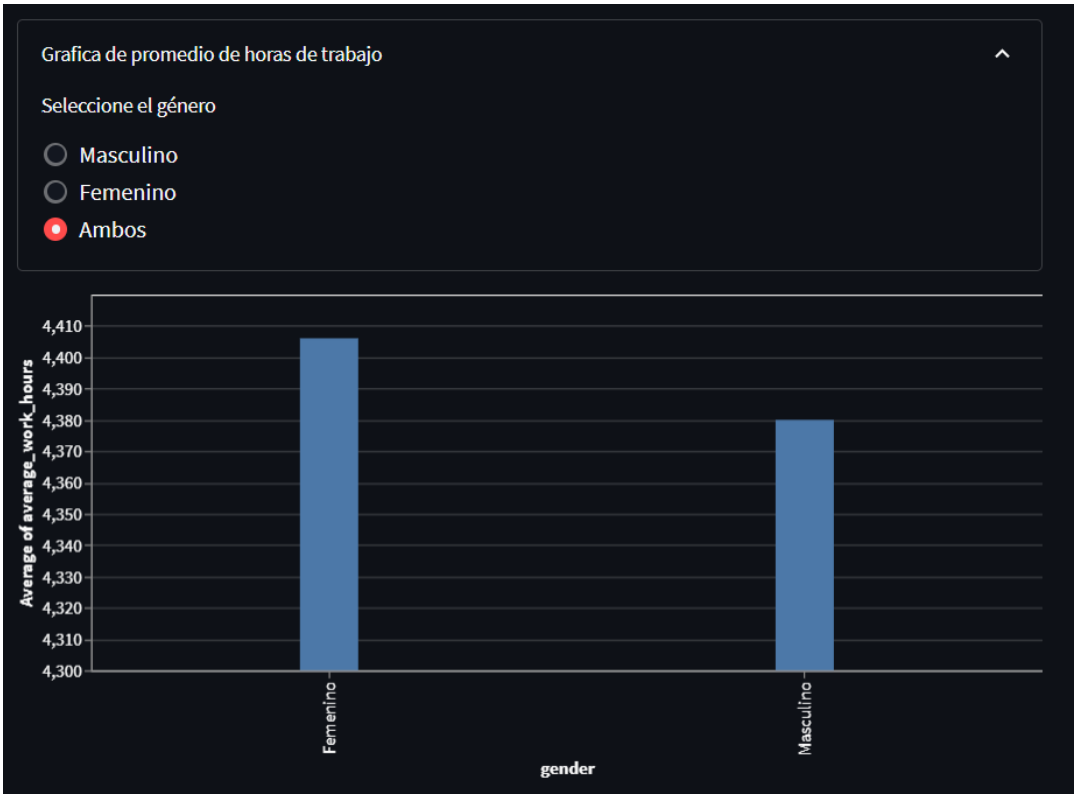



```
# Titulo, logo y descripcion
col1, mid, col2 = st.columns([4,2,20])
with col1:
    engranes = Image.open('ajuste.png')
    logo = engranes.resize((100, 100))
    st.image(logo)
with col2:
    st.header('EMPLEADOS')

st.subheader('Esta aplicación web está dedicada para mostrar en
análisis de desempeño de empleados y KPIs correspondientes de
_Socialize your Knowledge_.')
```

Creando el radioButton del género y la gráfica correspondiente a las horas de trabajo según la selección:



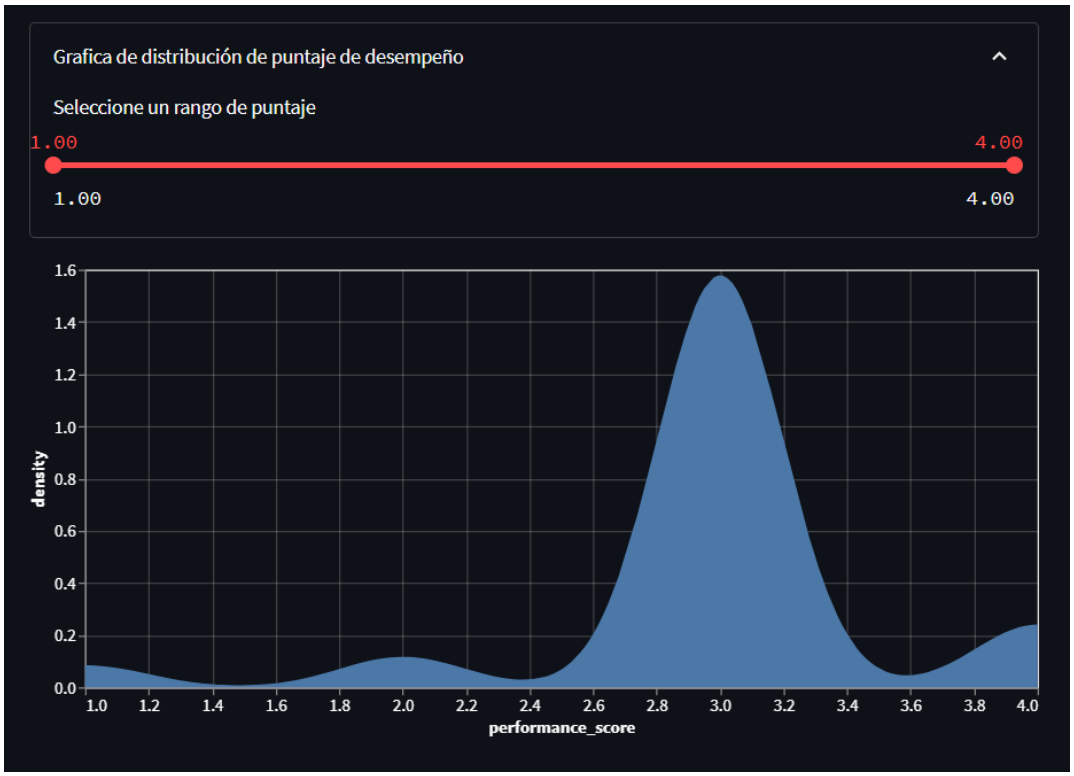


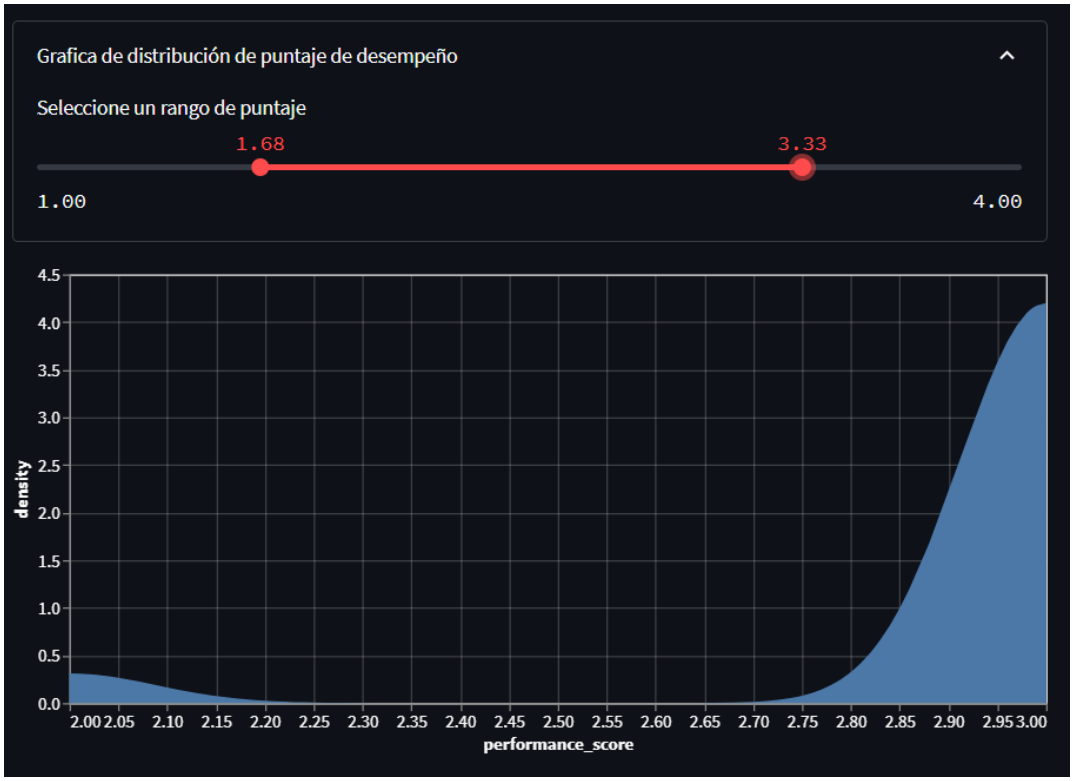



```
# Cargar datos
empleados = pd.read_csv("Empleados.csv")
empleados['gender'] = empleados['gender'].replace('M ','Masculino')
empleados['gender'] = empleados['gender'].replace('F','Femenino')

# Radio de genero
generoAux = empleados['gender'].unique()
generoAux = np.append(generoAux, "Ambos")

confGenero = st.expander("Grafica de promedio de horas de trabajo",
True)
genero = confGenero.radio("Seleccione el género", generoAux)

# Grafica de barras sobre el promedio
if genero == 'Ambos':
    graphPromedio = alt.Chart(empleados).mark_bar(size=40).encode(
        x = 'gender:N',
        y = alt.Y('average_work_hours:Q', aggregate='average',
            scale=alt.Scale(domain=(4300, 4420)),
            axis=alt.Axis(values=list(range(4300, 4420, 10)))
        )
    )
    st.altair_chart(graphPromedio, use_container_width=True)
```

```python
# Caso femenino(0)
elif genero == 'Femenino':
    auxF = empleados[empleados['gender'] == 'Femenino']

    graphPromedio = alt.Chart(auxF).mark_bar(size=40).encode(
        x = 'gender:N',
        y = alt.Y('average_work_hours:Q', aggregate='average',
            scale=alt.Scale(domain=(4300, 4420)),
            axis=alt.Axis(values=list(range(4300, 4420, 10)))
        )
    )
    st.altair_chart(graphPromedio, use_container_width=True)



# Caso masculino(1)
elif genero == 'Masculino':
    auxM = empleados[empleados['gender'] == 'Masculino']

    graphPromedio = alt.Chart(auxM).mark_bar(size=40).encode(
        x = 'gender:N',
        y = alt.Y('average_work_hours:Q', aggregate='average',
            scale=alt.Scale(domain=(4300, 4420)),
            axis=alt.Axis(values=list(range(4300, 4420, 10)))
        )
    )
    st.altair_chart(graphPromedio, use_container_width=True)
```

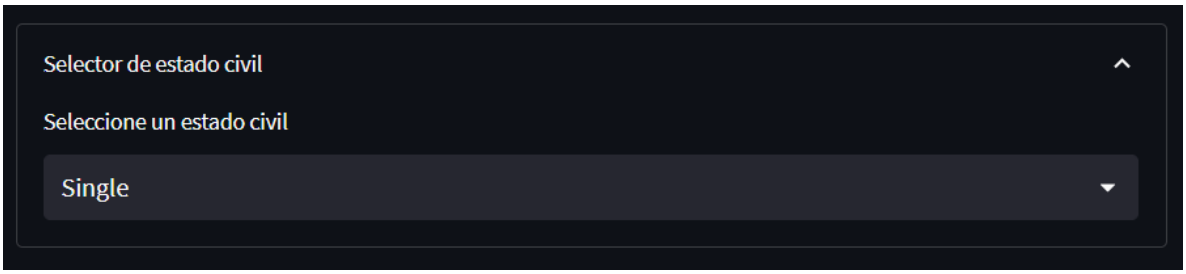
Creando el slider del puntaje de desempeño y la gráfica correspondiente a la distribución de los puntajes seleccionados:

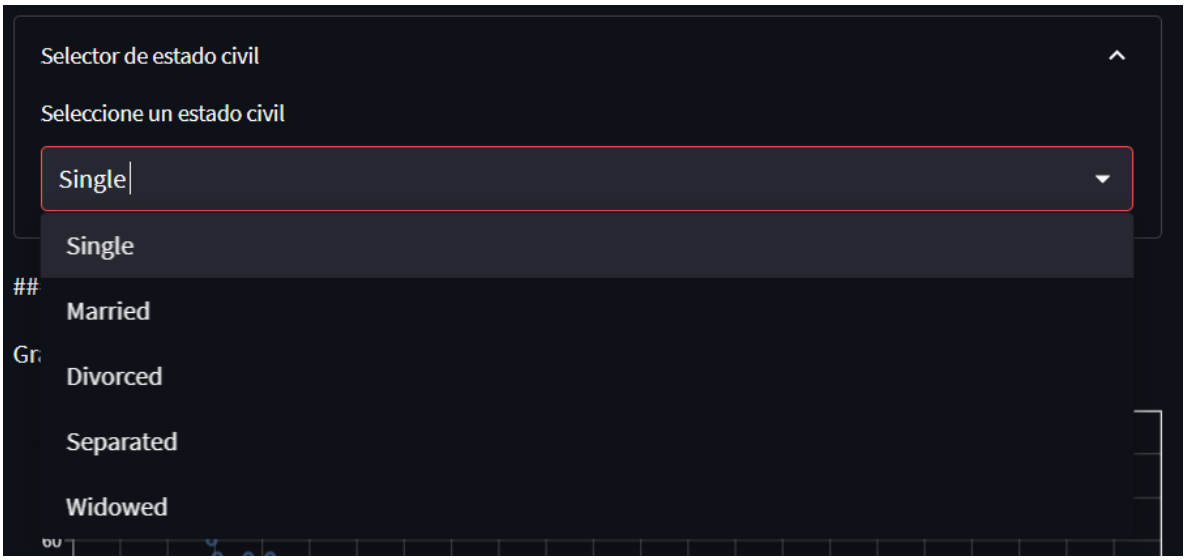




```
# Slider de puntaje de desempeño
empleados['performance_score'] =
pd.to_numeric(empleados['performance_score'])

confSlider = st.expander("Grafica de distribución de puntaje de
desempeño", True)
puntaje = confSlider.slider(
    "Seleccione un rango de puntaje",
        value = [float(empleados['performance_score'].min()),
float(empleados['performance_score'].max())],
        min_value = float(empleados['performance_score'].min()),
        max_value = float(empleados['performance_score'].max())
)
puntajeSelect =
empleados[empleados['performance_score'].between(puntaje[0],
puntaje[1])] #0 es min, 1 es max

# Grafica de distribucion del puntaje de desempeño
graphPuntaje = alt.Chart(puntajeSelect).transform_density(
    'performance_score',
    as_=['performance_score', 'density'],
).mark_area().encode(
    x="performance_score:Q",
    y='density:Q',
)
st.altair_chart(graphPuntaje, use_container_width=True)
```

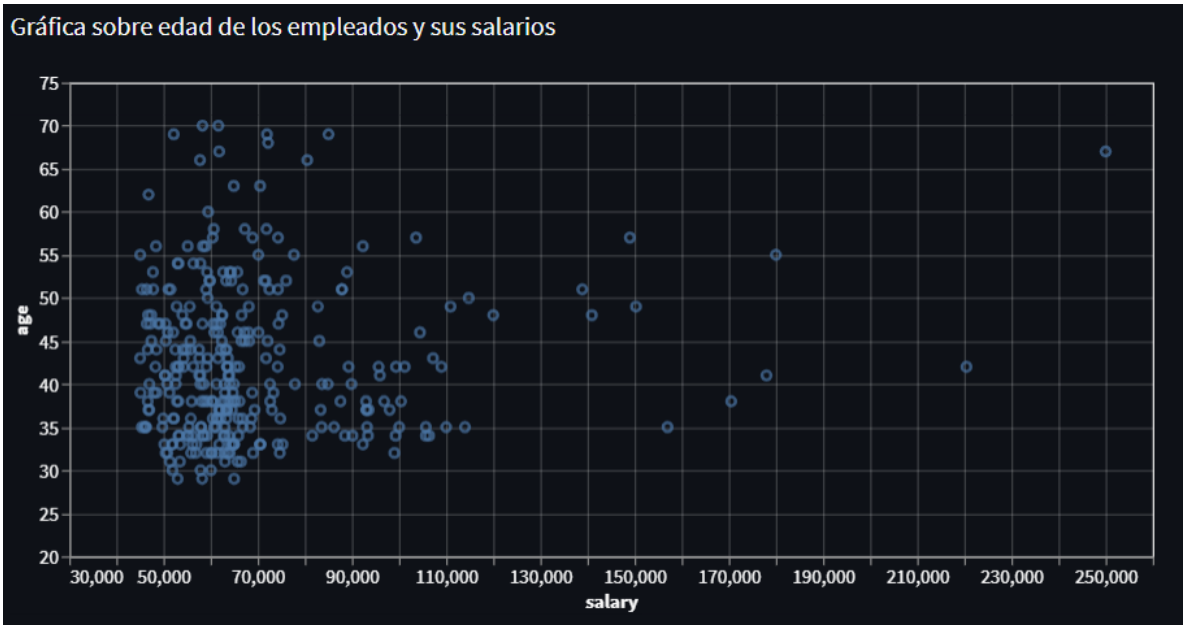
Creando el selectBox del estado civil; este elemento no cuenta con su propia gráfica:






```
# Select de estado civil
confCivil = st.expander("Selector de estado civil", True)
civil = confCivil.selectbox("Seleccione un estado civil",
empleados['marital_status'].unique())
```

Creando la gráfica correspondiente a la edad de los empleados y su salario correspondiente:

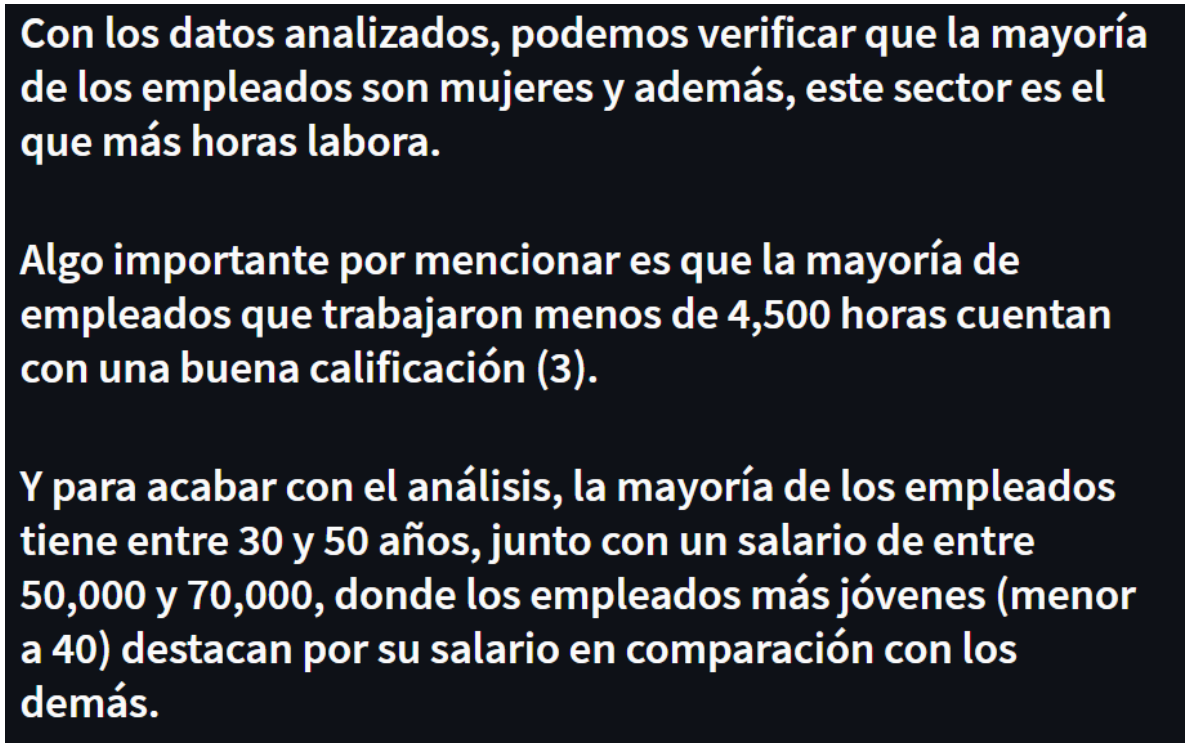



```
# Grafica de edad / salario
st.write("Gráfica sobre edad de los empleados y sus salarios")

graphEdad = alt.Chart(empleados).mark_point().encode(
    x = alt.X('salary', scale=alt.Scale(domain=(30000, 260000))),
    y = alt.Y('age', scale=alt.Scale(domain=(20, 75)))
)
st.altair_chart(graphEdad, use_container_width=True)
```

Creando la gráfica correspondiente a las horas de trabajo y el puntaje de desempeño:


```
# Grafica de horas / puntaje
st.write("Gráfica sobre las horas trabajadas de los empleados y su
puntuación de desempeño")

graphHoras = alt.Chart(empleados).mark_tick().encode(
    x = alt.X('average_work_hours',
        scale=alt.Scale(domain=(3900, 5200))
    ),
    y = alt.Y('performance_score',
        scale=alt.Scale(domain=(0, 5)),
        axis=alt.Axis(values=list(range(0, 6)))
    )
)
st.altair_chart(graphHoras, use_container_width=True)
```

Escribiendo las conclusiones del análisis:




```
# Conclusiones
st.subheader('Con los datos analizados, podemos verificar que la mayoría de los empleados son mujeres y además, este sector es el que más horas labora.')
st.subheader('Algo importante por mencionar es que la mayoría de empleados que trabajaron menos de 4,500 horas cuentan con una buena calificación (3).')
st.subheader('Y para acabar con el análisis, la mayoría de los empleados tiene entre 30 y 50 años, junto con un salario de entre 50,000 y 70,000, donde los empleados más jóvenes (menor a 40) destacan por su salario en comparación con los demás.')
```