**MÓDULO**

Aplicación web de ciencia de datos

**NOMBRE**

Genaro Salgado Alarcón

**EVALUACIÓN FINAL**

**Agregando el título, logo y descripción:**

Texto

Descripción generada automáticamente

*# 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:**

Gráfico

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

*# 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)

*# 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:**

Gráfico

Descripción generada automáticamente

Imagen que contiene Gráfico

Descripción generada automáticamente

*# 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:**

Patrón de fondo

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

*# 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:**

Imagen que contiene Diagrama

Descripción generada automáticamente

*# 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:**

Imagen que contiene Diagrama

Descripción generada automáticamente

*# 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:**

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

*# 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.')