Examen final de Desarrollo de Aplicaciones para la Visualización de Datos

Nombre: Juan

Apellidos: de Egaña Marin

Tiempo: 2 horas y 30 minutos

Contexto del ejercicio

Una gran compañía aseguradora de salud desea entender con mayor profundidad los patrones de coste médico, el uso de servicios sanitarios y los factores que determinan el riesgo clínico de sus asegurados.

La empresa cree que ciertos hábitos de vida, condiciones crónicas y características del seguro tienen un fuerte impacto sobre:

- Los costes médicos anuales (annual_medical_cost)
- El total pagado en reclamaciones médicas (total_claims_paid)
- La utilización de servicios médicos, incluyendo consultas, hospitalizaciones y procedimientos
- De que depende de que un asegurado sea clasificado como de alto riesgo (is_high_risk)

La aseguradora sospecha que estos resultados están influenciados por variables relacionadas con:

- Hábitos de vida (bmi, smoker, alcohol, ejercicio…)
- Enfermedades crónicas (diabetes, hipertensión, cardiopatías…)
- Características del seguro (tipo de plan, deducible, copago, calidad del proveedor)

Tu rol será actuar como Analista de Datos, llevando a cabo un análisis que permita identificar patrones significativos y generar recomendaciones accionables para la aseguradora.

Tareas obligatorias a realizar:

1. Análisis exploratorio con clusterización (mínimo 6 gráficos)
2. Entrena un modelo y explicable, a elegir entre:
   - Clasificación para predecir **is_high_risk**
   - Regresión para predecir **annual_medical_cost**
3. Dashboard con mínimo 4 visualizaciones (3 gráficas + 1 coeficientes)
4. Informe ejecutivo de dos páginas con:

- Los hallazgos más relevantes del análisis exploratorio
- Insights principales del modelo predictivo
- El dashboard con los 4 gráficos más representativos
- Recomendaciones accionables basadas en los datos

Entregable y puntuación

Se entregará un informe con la siguiente estructura:

1. Resumen ejecutivo (3 puntos) + dashboard (2 puntos)
2. Gráficas del análisis exploratorio y breve explicación de cada una (3 puntos)
3. Modelo predictivo explicado (2 puntos)

Entrega del exámen

Subir al siguiente enlace tu informe en **formato PDF**.

https://forms.office.com/e/zPHcGS22x9

En el repo existe un .docx con el formato para entregar el informe.

Juego de datos

Para realizar este análisis se provee el fichero **medical_insurance.csv** con las siguientes variables:

Rows: 100,000 **Columns:** 54+

Categoría	Variable	Tipo (esperado)	Descripción / Significado
Demographics & Socioeconomic			
	person_id	string/int	Identificador único de la persona.
	age	int	Edad del individuo.
	sex	category	Sexo (male/female/other).
	region	category	Región geográfica donde reside.
	urban_rural	category	Tipo de zona (urbana vs rural).
	income	float	Nivel de ingresos anuales.
	education	category	Nivel educativo alcanzado.
	marital_status	category	Estado civil (single, married, divorced...).

Categoría	Variable	Tipo (esperado)	Descripción / Significado
	employment_status	category	Situación laboral (employed, unemployed, retired…).
	household_size	int	Número total de personas en el hogar.
	dependents	int	Número de dependientes económicos.
Lifestyle & Habits			
	bmi	float	Índice de masa corporal.
	smoker	category/bool	Si la persona es fumadora.
	alcohol_freq	category	Frecuencia de consumo de alcohol.
	exercise_frequency	category/int	Frecuencia de ejercicio semanal.
	sleep_hours	float	Promedio de horas de sueño diario.
	stress_level	int	Nivel de estrés percibido (escala ordinal).
Health & Clinical			
	hypertension	bool	Diagnóstico de hipertensión.
	diabetes	bool	Diagnóstico de diabetes.
	copd	bool	Enfermedad pulmonar obstructiva crónica.
	cardiovascular	bool	Enfermedad cardiovascular.
	cancer_history	bool	Antecedente personal de cáncer.
	kidney_disease	bool	Enfermedad renal crónica.
	liver_disease	bool	Enfermedad hepática crónica.
	arthritis	bool	Diagnóstico de artritis.
	mental_health	bool/category	Algún trastorno de salud mental reportado.
	chronic_count	int	Número total de enfermedades crónicas diagnosticadas.
	systolic_bp	int	Presión arterial sistólica.
	diastolic_bp	int	Presión arterial diastólica.

Categoría	Variable	Tipo (esperado)	Descripción / Significado
	ldl	float	Colesterol LDL.
	hba1c	float	Hemoglobina glicosilada, indicador de diabetes.
	risk_score	float	Puntuación compuesta de riesgo clínico.
	is_high_risk	bool	Indicador de alto riesgo clínico.
Healthcare Utilization & Procedures			
	visits_last_year	int	Visitas médicas en el último año.
	hospitalizations_last_3yrs	int	Número de hospitalizaciones en 3 años.
	days_hospitalized_last_3yrs	int	Días totales hospitalizado en 3 años.
	medication_count	int	Cantidad de medicamentos activos.
	proc_imaging	int	Cantidad de estudios de imagen realizados.
	proc_surgery	int	Número de procedimientos quirúrgicos.
	proc_psycho	int	Número de consultas/procedimientos de psicología.
	proc_consult_count	int	Número total de consultas médicas.
	proc_lab	int	Número de exámenes de laboratorio.
	had_major	bool	Si tuvo un procedimiento mayor (cirugía importante).
Insurance & Policy			
	plan_type	category	Tipo de plan (HMO, PPO, etc.).
	network_tier	category	Nivel de red (gold/silver/bronze).
	deductible	float	Deducible anual del seguro.
	copay	float	Copago por servicio.
	policy_term_years	int	Duración del contrato del seguro.

Categoría	Variable	Tipo (esperado)	Descripción / Significado
	policy_changes_last_2yrs	int	Cambios realizados a la póliza en 2 años.
	provider_quality	float	Índice de calidad del proveedor.
Medical Costs & Claims			
	annual_medical_cost	float	Coste médico anual real del paciente.
	annual_premium	float	Prima anual pagada por el paciente.
	monthly_premium	float	Prima mensual pagada.
	claims_count	int	Número de reclamaciones realizadas.
	avg_claim_amount	float	Valor promedio por reclamación.
	total_claims_paid	float	Total pagado por la compañía aseguradora.

Ejemplos de preguntas que se pueden realizar a los datos:

1. ¿Cómo varía el coste médico anual según edad, sexo y región?
2. ¿Qué relación existe entre hábitos de vida (bmi, smoker, alcohol_freq...) y el coste médico anual?
3. ¿Cuánto más cuesta un paciente de alto riesgo respecto a uno de bajo riesgo?
4. ¿Qué enfermedades crónicas generan mayores costes promedio?
5. ¿Existe relación entre hospitalizaciones y coste total en los últimos 3 años?
6. ¿Qué parámetros clínicos tienen mayor correlación con el coste médico?
7. ¿Qué tipos de seguro están asociados con mayor gasto?
8. ¿Cómo se relacionan los procedimientos médicos con el total pagado en reclamaciones?
9. ¿Qué variables explican mejor los costes según un modelo predictivo?
10. ¿Qué grupos de pacientes comparten patrones similares de riesgo y coste (clustering)?

Nota:

- Se valorará la creatividad en las hipótesis, soluciones y limpieza del código y visualizaciones.

Carga tus librerías

In [14]:
```
!pip install plotly jupyter-dash dash dash-bootstrap-components scikit-learn
```

```
Requirement already satisfied: plotly in c:\users\juane\anaconda3\lib\site-packag
es (5.22.0)
Collecting jupyter-dash
  Downloading jupyter_dash-0.4.2-py3-none-any.whl.metadata (3.6 kB)
Collecting dash
  Downloading dash-3.3.0-py3-none-any.whl.metadata (11 kB)
Collecting dash-bootstrap-components
  Downloading dash_bootstrap_components-2.0.4-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: scikit-learn in c:\users\juane\anaconda3\lib\site-
packages (1.6.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\juane\anaconda3\lib\si
te-packages (from plotly) (8.2.2)
Requirement already satisfied: packaging in c:\users\juane\anaconda3\lib\site-pac
kages (from plotly) (23.2)
Requirement already satisfied: requests in c:\users\juane\anaconda3\lib\site-pack
ages (from jupyter-dash) (2.32.2)
Requirement already satisfied: flask in c:\users\juane\anaconda3\lib\site-package
s (from jupyter-dash) (3.0.3)
Collecting retrying (from jupyter-dash)
  Downloading retrying-1.4.2-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: ipython in c:\users\juane\anaconda3\lib\site-packa
ges (from jupyter-dash) (8.25.0)
Requirement already satisfied: ipykernel in c:\users\juane\anaconda3\lib\site-pac
kages (from jupyter-dash) (6.28.0)
Collecting ansi2html (from jupyter-dash)
  Downloading ansi2html-1.9.2-py3-none-any.whl.metadata (3.7 kB)
Requirement already satisfied: nest-asyncio in c:\users\juane\anaconda3\lib\site-
packages (from jupyter-dash) (1.6.0)
Requirement already satisfied: Werkzeug<3.2 in c:\users\juane\anaconda3\lib\site-
packages (from dash) (3.0.3)
Requirement already satisfied: importlib-metadata in c:\users\juane\anaconda3\lib
\site-packages (from dash) (7.0.1)
Requirement already satisfied: typing_extensions>=4.1.1 in c:\users\juane\anacond
a3\lib\site-packages (from dash) (4.11.0)
Requirement already satisfied: setuptools in c:\users\juane\anaconda3\lib\site-pa
ckages (from dash) (69.5.1)
Requirement already satisfied: numpy>=1.19.5 in c:\users\juane\anaconda3\lib\site
-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\juane\anaconda3\lib\site-
packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\juane\anaconda3\lib\site
-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\juane\anaconda3\l
ib\site-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\juane\anaconda3\lib\site
-packages (from flask->jupyter-dash) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\juane\anaconda3\li
b\site-packages (from flask->jupyter-dash) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\juane\anaconda3\lib\site-
packages (from flask->jupyter-dash) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\juane\anaconda3\lib\sit
e-packages (from flask->jupyter-dash) (1.6.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\juane\anaconda3\lib
\site-packages (from Werkzeug<3.2->dash) (2.1.3)
Requirement already satisfied: zipp>=0.5 in c:\users\juane\anaconda3\lib\site-pac
kages (from importlib-metadata->dash) (3.17.0)
Requirement already satisfied: comm>=0.1.1 in c:\users\juane\anaconda3\lib\site-p
ackages (from ipykernel->jupyter-dash) (0.2.1)
Requirement already satisfied: debugpy>=1.6.5 in c:\users\juane\anaconda3\lib\sit
e-packages (from ipykernel->jupyter-dash) (1.6.7)
```

```
Requirement already satisfied: jupyter-client>=6.1.12 in c:\users\juane\anaconda3
\lib\site-packages (from ipykernel->jupyter-dash) (8.6.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in c:\users\juane\anaco
nda3\lib\site-packages (from ipykernel->jupyter-dash) (5.7.2)
Requirement already satisfied: matplotlib-inline>=0.1 in c:\users\juane\anaconda3
\lib\site-packages (from ipykernel->jupyter-dash) (0.1.6)
Requirement already satisfied: psutil in c:\users\juane\anaconda3\lib\site-packag
es (from ipykernel->jupyter-dash) (5.9.0)
Requirement already satisfied: pyzmq>=24 in c:\users\juane\anaconda3\lib\site-pac
kages (from ipykernel->jupyter-dash) (25.1.2)
Requirement already satisfied: tornado>=6.1 in c:\users\juane\anaconda3\lib\site-
packages (from ipykernel->jupyter-dash) (6.4.1)
Requirement already satisfied: traitlets>=5.4.0 in c:\users\juane\anaconda3\lib\s
ite-packages (from ipykernel->jupyter-dash) (5.14.3)
Requirement already satisfied: decorator in c:\users\juane\anaconda3\lib\site-pac
kages (from ipython->jupyter-dash) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\juane\anaconda3\lib\site-pa
ckages (from ipython->jupyter-dash) (0.18.1)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in c:\users\juane\an
aconda3\lib\site-packages (from ipython->jupyter-dash) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in c:\users\juane\anaconda3\lib\si
te-packages (from ipython->jupyter-dash) (2.15.1)
Requirement already satisfied: stack-data in c:\users\juane\anaconda3\lib\site-pa
ckages (from ipython->jupyter-dash) (0.2.0)
Requirement already satisfied: colorama in c:\users\juane\anaconda3\lib\site-pack
ages (from ipython->jupyter-dash) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\juane\anacond
a3\lib\site-packages (from requests->jupyter-dash) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\juane\anaconda3\lib\site-
packages (from requests->jupyter-dash) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\juane\anaconda3\lib
\site-packages (from requests->jupyter-dash) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\juane\anaconda3\lib
\site-packages (from requests->jupyter-dash) (2024.8.30)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\juane\anaconda3\li
b\site-packages (from jedi>=0.16->ipython->jupyter-dash) (0.8.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\juane\anaconda3
\lib\site-packages (from jupyter-client>=6.1.12->ipykernel->jupyter-dash) (2.9.0.
post0)
Requirement already satisfied: platformdirs>=2.5 in c:\users\juane\anaconda3\lib
\site-packages (from jupyter-core!=5.0.*,>=4.12->ipykernel->jupyter-dash) (3.10.
0)
Requirement already satisfied: pywin32>=300 in c:\users\juane\anaconda3\lib\site-
packages (from jupyter-core!=5.0.*,>=4.12->ipykernel->jupyter-dash) (305.1)
Requirement already satisfied: wcwidth in c:\users\juane\anaconda3\lib\site-packa
ges (from prompt-toolkit<3.1.0,>=3.0.41->ipython->jupyter-dash) (0.2.5)
Requirement already satisfied: executing in c:\users\juane\anaconda3\lib\site-pac
kages (from stack-data->ipython->jupyter-dash) (0.8.3)
Requirement already satisfied: asttokens in c:\users\juane\anaconda3\lib\site-pac
kages (from stack-data->ipython->jupyter-dash) (2.0.5)
Requirement already satisfied: pure-eval in c:\users\juane\anaconda3\lib\site-pac
kages (from stack-data->ipython->jupyter-dash) (0.2.2)
Requirement already satisfied: six>=1.5 in c:\users\juane\anaconda3\lib\site-pack
ages (from python-dateutil>=2.8.2->jupyter-client>=6.1.12->ipykernel->jupyter-das
h) (1.16.0)
Downloading jupyter_dash-0.4.2-py3-none-any.whl (23 kB)
Downloading dash-3.3.0-py3-none-any.whl (7.9 MB)
   ---------------------------------------- 0.0/7.9 MB ? eta -:--:--
   - -------------------------------------- 0.3/7.9 MB 7.0 MB/s eta 0:00:02
   ---- ----------------------------------- 0.9/7.9 MB 11.3 MB/s eta 0:00:01
```

```
   -------- ------------------------------- 1.7/7.9 MB 13.4 MB/s eta 0:00:01
   ------------ --------------------------- 2.6/7.9 MB 13.6 MB/s eta 0:00:01
   ---------------- ----------------------- 3.3/7.9 MB 14.9 MB/s eta 0:00:01
   -------------------- ------------------- 4.2/7.9 MB 15.6 MB/s eta 0:00:01
   -------------------------- ------------- 5.2/7.9 MB 15.8 MB/s eta 0:00:01
   ------------------------------- -------- 6.2/7.9 MB 16.5 MB/s eta 0:00:01
   ------------------------------------ --- 7.2/7.9 MB 17.0 MB/s eta 0:00:01
   ---------------------------------------  7.9/7.9 MB 17.5 MB/s eta 0:00:01
   ---------------------------------------- 7.9/7.9 MB 16.9 MB/s eta 0:00:00
Downloading dash_bootstrap_components-2.0.4-py3-none-any.whl (204 kB)
   ---------------------------------------- 0.0/204.0 kB ? eta -:--:--
   ---------------------------------------- 204.0/204.0 kB 12.9 MB/s eta 0:00:00
Downloading ansi2html-1.9.2-py3-none-any.whl (17 kB)
Downloading retrying-1.4.2-py3-none-any.whl (10 kB)
Installing collected packages: retrying, ansi2html, dash, dash-bootstrap-componen
ts, jupyter-dash
Successfully installed ansi2html-1.9.2 dash-3.3.0 dash-bootstrap-components-2.0.4
jupyter-dash-0.4.2 retrying-1.4.2
```

In [190…

```
import pandas as pd
import plotly.express as px
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from dash import Dash, html, dcc
from dash.dependencies import Input, Output
```

Escribe tu código

In [73]:

```
# Cargar CSV
df = pd.read_csv("medical_insurance.csv")
df.head()
```

Out[73]:

	person_id	age	sex	region	urban_rural	income	education	marital_status	em
0	75722	52	Female	North	Suburban	22700.0	Doctorate	Married	
1	80185	79	Female	North	Urban	12800.0	No HS	Married	
2	19865	68	Male	North	Rural	40700.0	HS	Married	
3	76700	15	Male	North	Suburban	15600.0	Some College	Married	
4	92992	53	Male	Central	Suburban	89600.0	Doctorate	Married	

5 rows × 54 columns

In [99]:

```
# 1) Distribucion coste medico
fig = px.histogram(df, x="annual_medical_cost", nbins=300)
fig.show()
```

In [107…

```
# 2) Coste medico por edad
fig = px.scatter(df, x="age", y="annual_medical_cost", opacity=0.5)
fig.show()
```

In [81]:
```
# 3) Boxplot por fumador
fig = px.box(df, x="smoker", y="annual_medical_cost")
fig.show()
```

In [109…

```python
# 4) Relación BMI-coste
fig = px.scatter(df, x="bmi", y="annual_medical_cost", color="smoker")
fig.show()
```

In [85]:
```
# 5) Correlación simple visitas → coste
fig = px.scatter(df, x="visits_last_year", y="annual_medical_cost")
fig.show()
```

In [111…

```
# 6) Coste por enfermedades cronicas
fig = px.box(df, x="chronic_count", y="annual_medical_cost")
fig.show()
```

In [167…

```
#figura a parte a ver si veo como esta relacionado bien
numeric_cols = df.select_dtypes(include=[np.number]).columns
corr_matrix = df[numeric_cols].corr()

fig = px.imshow(
    corr_matrix,
    zmin=-1,
    zmax=1,
    color_continuous_scale="RdBu",
    title="Matriz de correlación entre variables numéricas",
    labels=dict(x="Variables", y="Variables", color="Correlación"),
)

fig.update_xaxes(side="bottom")
fig.show()
```

In [221…

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import plotly.express as px

cols_cluster = ["age","bmi","chronic_count","visits_last_year","hospitalizations

X = df[cols_cluster].dropna()

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=3, random_state=42)
df.loc[X.index, "cluster"] = kmeans.fit_predict(X_scaled)

fig = px.scatter(
    df,
    x="age",
    y="annual_medical_cost",
    color="cluster",
    opacity=0.6,
    title="Clustering (KMeans)"
)
fig.show()
```

Vamos a intentar clusterizar por lo que hemos visto que tiene sentido (cronicas)

In [208…

```
cols = ["age", "bmi", "chronic_count", "visits_last_year", "hospitalizations_las

df_model = df[cols + ["is_high_risk"]].dropna()

X = df_model[cols]
y = df_model["is_high_risk"]
#split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

rf = Pipeline([
    ("scaler", StandardScaler()),
    ("model", RandomForestClassifier(n_estimators=300, max_depth=10, random_stat

# entreno
rf.fit(X_train, y_train)

# prediccion
y_pred = rf.predict(X_test)
y_prob = rf.predict_proba(X_test)[:, 1]

# metricas
print("matriz de confusion:\n")
print(classification_report(y_test, y_pred))

# Matriz de confusión
```

```
cm = confusion_matrix(y_test, y_pred)

fig_cm = px.imshow(
    cm,
    text_auto=True,
    color_continuous_scale="Blues",
    labels=dict(x="Predicción", y="Real"),
    title="Matriz de confusión"
)
fig_cm.show()

# Importancia de variables
model = rf.named_steps["model"]
importances = model.feature_importances_

feature_imp = pd.DataFrame({
    "var": cols,
    "importance": importances
}).sort_values("importance", ascending=False)

fig_imp = px.bar(
    feature_imp,
    x="var",
    y="importance",
    title="Importancia de variables"
)
fig_imp.show()
```

matriz de confusion:

	precision	recall	f1-score	support
0	0.93	1.00	0.96	12644
1	1.00	0.87	0.93	7356
accuracy			0.95	20000
macro avg	0.96	0.93	0.94	20000
weighted avg	0.95	0.95	0.95	20000

In [ ]:

In [ ]:

In [211…

```
target_col = "annual_medical_cost"
feature_cols = ["age", "bmi", "chronic_count", "visits_last_year", "is_high_risk

X = df[feature_cols]
y = df[target_col]

# Opcional: partimos train/test usando el 80% de fechas primeras como train (spl
n = len(df)
split_index = int(n * 0.8)

X_train, X_test = X.iloc[:split_index, :], X.iloc[split_index:, :]
y_train, y_test = y.iloc[:split_index], y.iloc[split_index:]

# Entrenamos la regresión lineal
linreg = LinearRegression()
linreg.fit(X_train, y_train)

# Evaluamos con R² en train y test
r2_train = linreg.score(X_train, y_train)
r2_test = linreg.score(X_test, y_test)

print(f"R² (train): {r2_train:.3f}")
print(f"R² (test): {r2_test:.3f}")
```

```
R² (train): 0.594
R² (test): 0.579
```

In [213…

```python
coef_df = pd.DataFrame({
    "variable": feature_cols,
    "coeficiente": linreg.coef_
})

coef_df["abs_coef"] = coef_df["coeficiente"].abs()
coef_df = coef_df.sort_values("abs_coef", ascending=False).drop(columns="abs_coe

coef_df
```

Out[213…

	variable	coeficiente
5	hospitalizations_last_3yrs	1108.593361
3	visits_last_year	-348.895795
4	is_high_risk	289.657506
2	chronic_count	275.767968
1	bmi	13.063848
6	medication_count	-6.299540
0	age	5.785484
7	total_claims_paid	1.049533

In [223…

```python
app = Dash(__name__)

app.layout = html.Div([
    html.H1("Dashboard Medical Insurance"),

    dcc.Dropdown(
        id="var",
        options=[{"label": c, "value": c} for c in ["age","bmi","chronic_count",
        value="bmi"
    ),

    dcc.Graph(id="g1"),
    dcc.Graph(id="g2"),
    dcc.Graph(id="g3"),

    html.H3("Importancia del modelo"),
    dcc.Graph(
        figure=px.bar(feature_imp, x="var", y="importance")
    )
])

@app.callback(
    Output("g1", "figure"),
    Input("var", "value")
)
def update_graph(v):
    return px.histogram(df, x=v)

@app.callback(
    Output("g2", "figure"),
```

```
    Input("var", "value")
)
def update_graph2(v):
    return px.box(df, y=v)

@app.callback(
    Output("g3", "figure"),
    Input("var", "value")
)
def update_graph3(v):
    return px.scatter(df, x=v, y="annual_medical_cost")

app.run(debug=False)
```

Loading...