



ASISTENTE PERSONAL DE SALUD MENTAL

Presentación de:

Estudiantes:

Jhoan Sebastian Garcia Reyes
Daniel Tarazona Sanchez
Juan Tabares

Código:

2202045
2210097
2191712

CONTEXTO Y PROBLEMÁTICA

- **Prevalencia de trastornos mentales:**

Casi mil millones de personas afectadas por algún trastorno mental en 2019.

- **Impacto de la pandemia:**

Incremento de casos de ansiedad y depresión.

- **Barreras para el tratamiento:**

Estigma y falta de recursos accesibles.

[Fuente: Organización Mundial de la Salud](#)

Depresión, estrés y ansiedad: ¿qué tan comunes son?

Encuestados que afirman haber experimentado problemas de salud mental en los últimos doce meses (en %)



De 1.600 a 10.000 encuestados (18-64 años) online por país entre julio de 2023 y junio de 2024.
Países seleccionados.
Fuente: Statista Consumer Insights

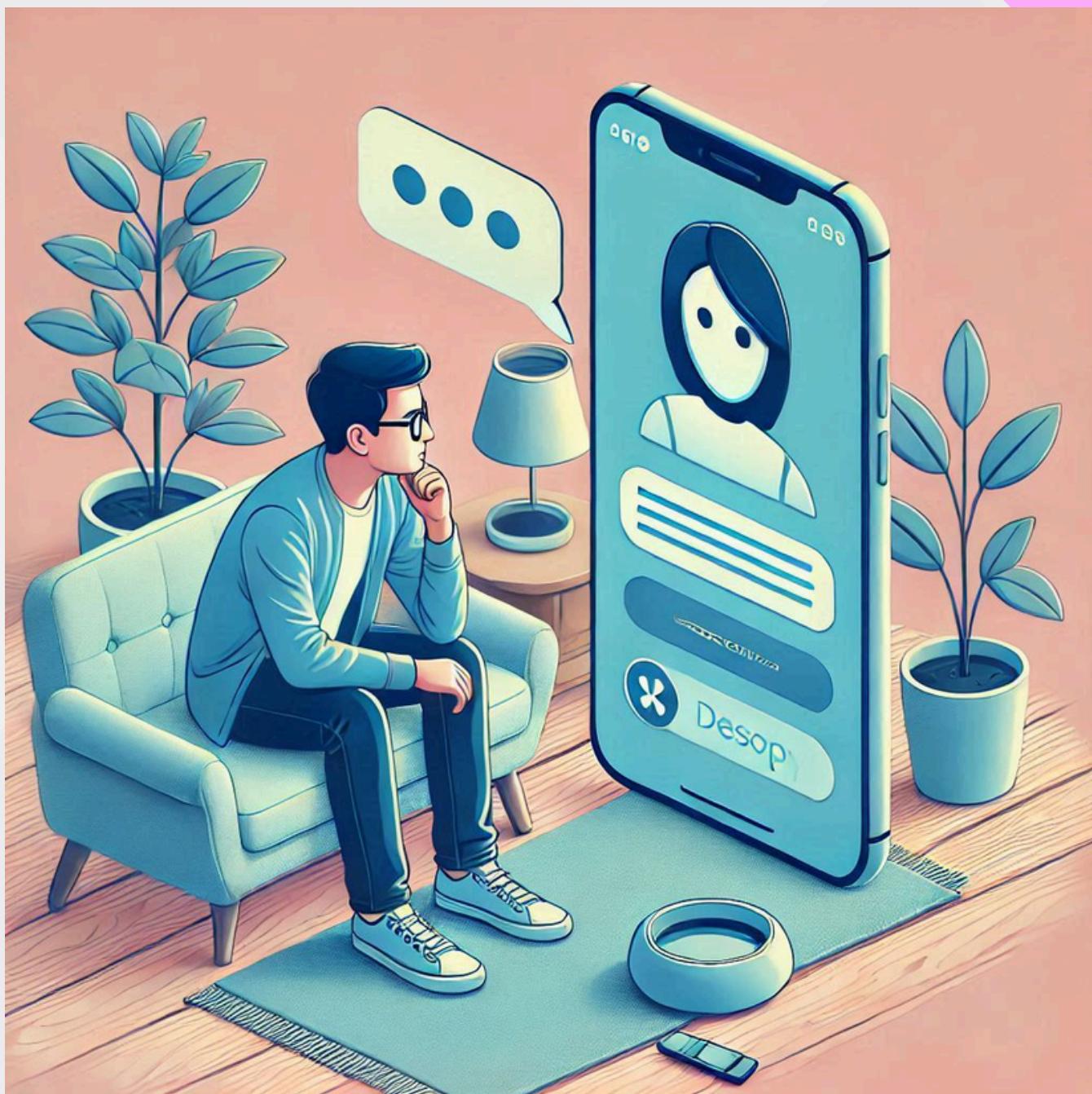


statista

Fuente: Statista

OBJETIVO DEL PROYECTO

- Asistente personal de salud mental basado en IA.
- Funciones:
 - Apoyo emocional.
 - Sugerencias personalizadas.
 - Detección temprana de trastornos.
 - Recursos de autocuidado.
 - Recordatorios de prácticas saludables.
 - Seguimiento del progreso emocional.



Fuente: [IACopilot](#)

PREPARACIÓN DE DATOS

- **Contenido:**

- **Fuente de datos:** Openpsychometrics.org
- **Características del dataset:** Casi 40,000 respuestas a la Escala DASS (2017-2019)
- **Preprocesamiento:** Limpieza, normalización y codificación de datos

[Fuente: Organización Mundial de la Salud](#)

Codebook

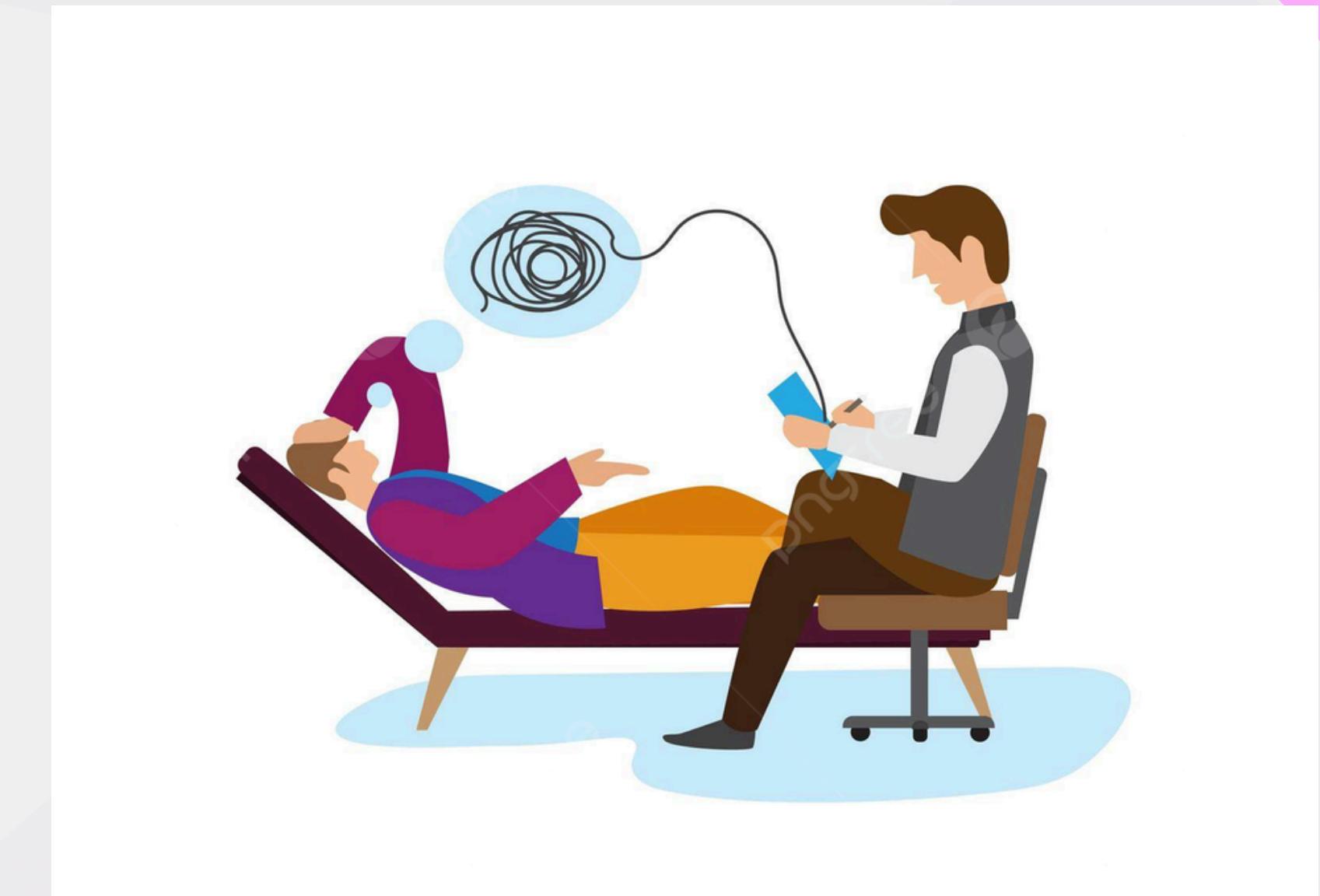
For reference, the following is a list of the survey questions as well as notes on how data was stored. This information was taken from the codebook downloaded alongside the CSV:

- Q1 I found myself getting upset by quite trivial things.
- Q2 I was aware of dryness of my mouth.
- Q3 I couldn't seem to experience any positive feeling at all.
- Q4 I experienced breathing difficulty (eg, excessively rapid breathing, breathlessness in the absence of physical exertion).
- Q5 I just couldn't seem to get going.
- Q6 I tended to over-react to situations.
- Q7 I had a feeling of shakiness (eg, legs going to give way).
- Q8 I found it difficult to relax.
- Q9 I found myself in situations that made me so anxious I was most relieved when they ended.
- Q10 I felt that I had nothing to look forward to.
- Q11 I found myself getting upset rather easily.
- Q12 I felt that I was using a lot of nervous energy.
- Q13 I felt sad and depressed.
- Q14 I found myself getting impatient when I was delayed in any way (eg, elevators, traffic lights, being kept waiting).
- Q15 I had a feeling of faintness.
- Q16 I felt that I had lost interest in just about everything.
- Q17 I felt I wasn't worth much as a person.
- Q18 I felt that I was rather touchy.
- Q19 I perspired noticeably (eg, hands sweaty) in the absence of high temperatures or physical exertion.
- Q20 I felt scared without any good reason.
- Q21 I felt that life wasn't worthwhile.
- Q22 I found it hard to wind down.
- Q23 I had difficulty in swallowing.
- Q24 I couldn't seem to get any enjoyment out of the things I did.
- Q25 I was aware of the action of my heart in the absence of physical exertion (eg, sense of heart rate increase, h heart missing a beat).
- Q26 I felt down-hearted and blue.
- Q27 I found that I was very irritable.
- Q28 I felt I was close to panic.
- Q29 I found it hard to calm down after something upset me.
- Q30 I feared that I would be "thrown" by some trivial but unfamiliar task.

[Fuente: Open-Source Psychometrics Project](#)

DESAFÍOS IDENTIFICADOS

- Limitaciones como sustituto profesional
- Confiabilidad en entornos reales
- Falta de interacción multimodal



Fuente: [PNGTree](#)

DATOS

¿Qué hace esto?

- Dice que pregunta y cuál es la característica más importante utilizando como columna objetivo la pregunta que encuentra mas importante según el Random Forest. Se aplica también PCA.

```
tamano_subconjunto = 20000
if len(data) > tamano_subconjunto:
    data = data.sample(n=tamano_subconjunto, random_state=42)

columnas_validas = data.columns[data.isnull().mean() < 0.5]
data_filtrado = data[columnas_validas].dropna()

columnas_categoricas = data_filtrado.select_dtypes(include=['object']).columns[:5]
data_procesada = pd.get_dummies(data_filtrado, columns=columnas_categoricas, drop_first=True)

columnas_preguntas = [col for col in data_procesada.columns if col.startswith('Q')]

X_base = data_procesada.drop(columns=columnas_preguntas)
importancias_preguntas = []

for pregunta in columnas_preguntas:
    try:
        y = data_procesada[pregunta]
        X_train, X_test, y_train, y_test = train_test_split(X_base, y, test_size=0.3, random_state=42)
        modelo_rf = RandomForestClassifier(random_state=42, n_estimators=10)
        modelo_rf.fit(X_train, y_train)
        score = modelo_rf.score(X_test, y_test)
        importancias_preguntas.append((pregunta, score))
    except Exception as e:
        print(f"Error al procesar la pregunta {pregunta}: {e}")

pregunta_relevante = max(importancias_preguntas, key=lambda x: x[1])[0]
print(f"La pregunta más relevante identificada como objetivo es: {pregunta_relevante}")

y = data_procesada[pregunta_relevante]
X = data_procesada.drop(columns=[pregunta_relevante])

modelo_rf = RandomForestClassifier(random_state=42, n_estimators=10)
modelo_rf.fit(X, y)
importancias = pd.DataFrame({
    'Característica': X.columns,
    'Importancia': modelo_rf.feature_importances_
}).sort_values(by='Importancia', ascending=False)

print("\nTop 10 características más importantes:")
print(importancias.head(10))

plt.figure(figsize=(10, 6))
sns.barplot(x='Importancia', y='Característica', data=importancias.head(10), hue='Característica', palette='viridis', dodge=False)
plt.title("Características más importantes según Random Forest")
plt.xlabel("Importancia")
plt.ylabel("Característica")
plt.legend([], [], frameon=False)
```

DATOS

```
plt.show()

pca = PCA(n_components=0.95)
X_pca = pca.fit_transform(X)

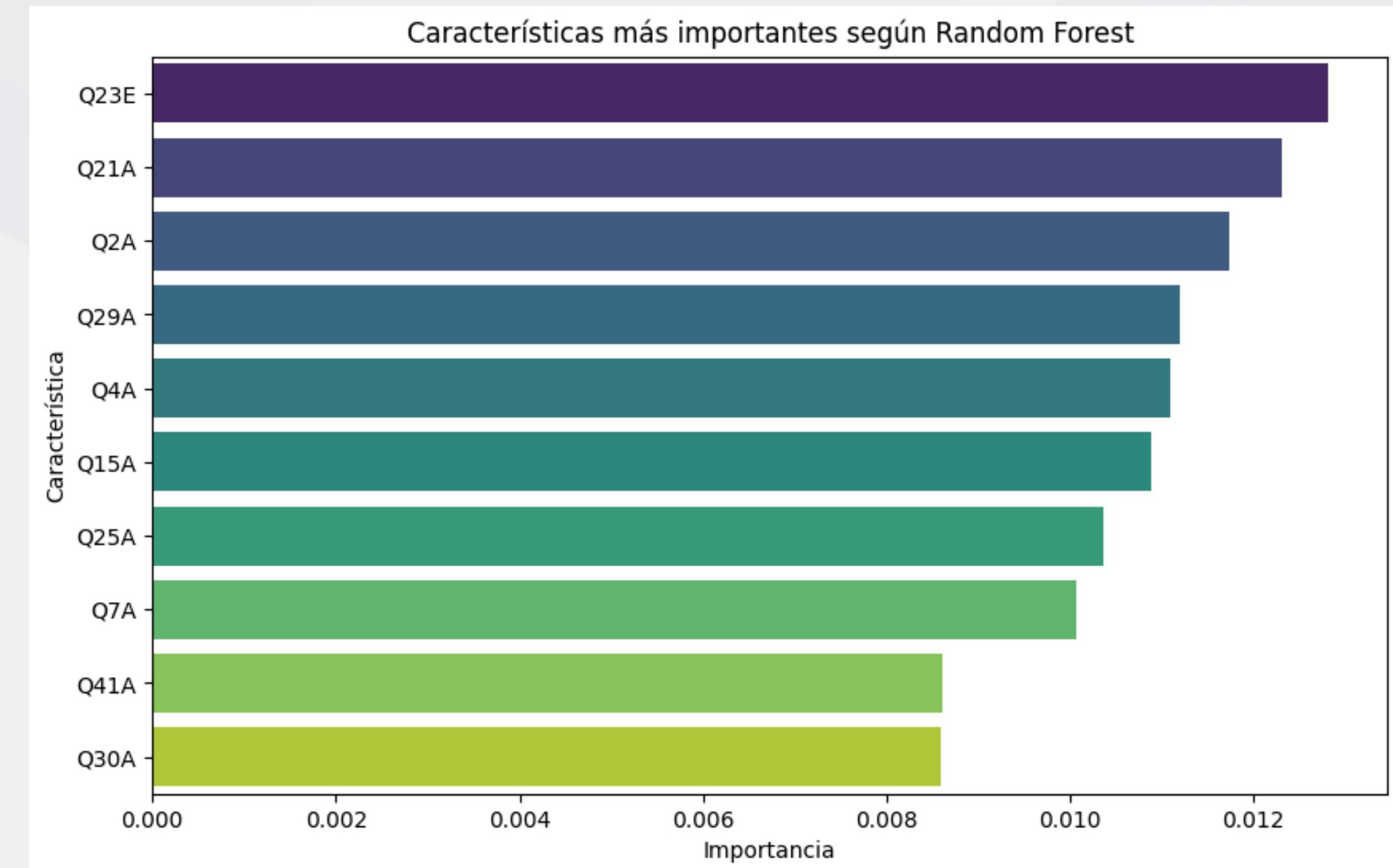
print("\nForma de los datos después de aplicar PCA:", X_pca.shape)
print("Varianza explicada por cada componente principal:", pca.explained_variance_ratio_)

plt.figure(figsize=(8, 6))
plt.plot(np.cumsum(pca.explained_variance_ratio_), marker='o', linestyle='--')
plt.title("Varianza explicada acumulada por PCA")
plt.xlabel("Número de componentes")
plt.ylabel("Varianza explicada acumulada")
plt.show()
```

METODOLOGÍA

Gráficas:

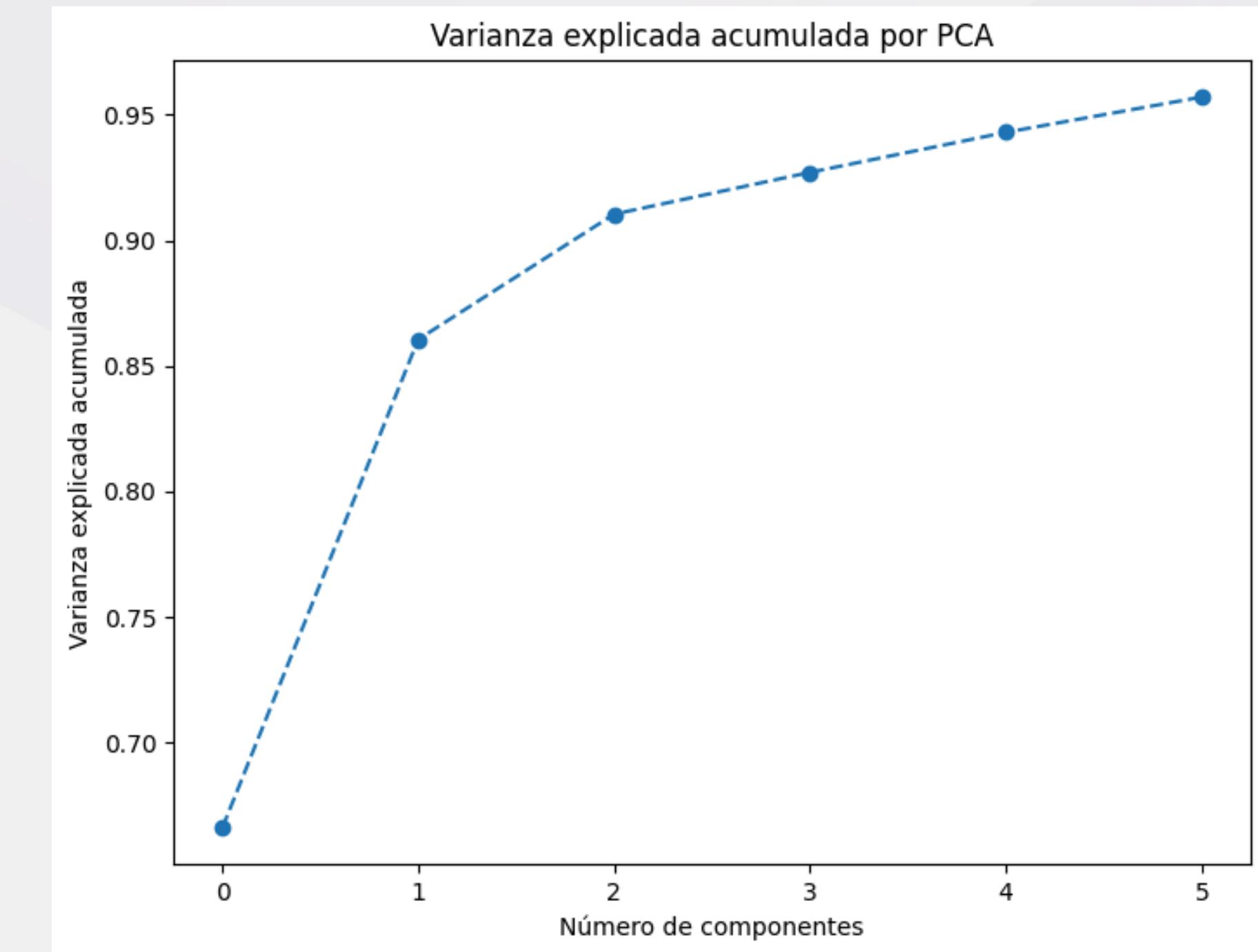
- Identifica las características más importantes que afectan la variable objetivo utilizando nuevamente, en la comparación de las métricas de desempeño, Random Forest, mostrando las 10 características principales a través de un gráfico.



METODOLOGÍA

Gráficas:

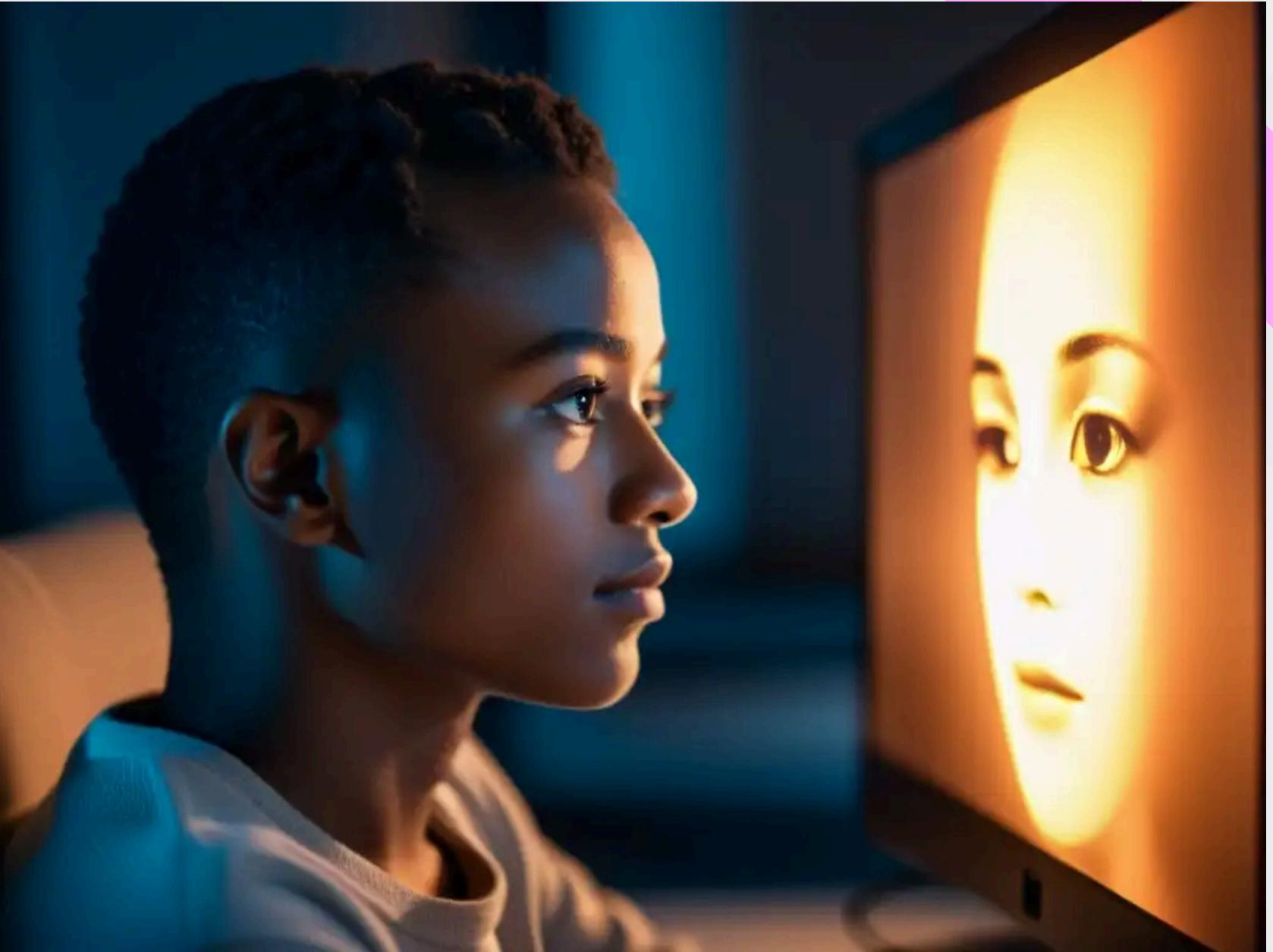
- Finalmente, aplica Análisis de Componentes Principales (PCA) para reducir la dimensionalidad del conjunto de datos, reteniendo el 95% de la varianza original, y visualiza la varianza explicada acumulada por los componentes principales.



CONCLUSIÓN

- **Innovación:** Asistente de salud mental basado en IA.
- **Resultados:** Avances en accesibilidad y personalización.
- **Futuro:** Ampliación de datos y fortalecimiento de medidas éticas.

[Fuente: Barcelona School Of Management](#)



[Fuente: Defensores Humanos](#)