

# Google\_Colab



## Análisis y visualización de datos con **Google Colab**

### ¿Qué es Colab? ▾

**Google Colab** (abreviatura de *Google Colaboratory*) es una herramienta **gratuita** basada en la nube que permite escribir y ejecutar código en Python (preferentemente) desde un navegador web **sin necesidad de configurar nada en tu máquina local**.

Es especialmente útil para:

- Proyectos de análisis de datos
- Aprendizaje automático (*machine learning*)
- Ciencia de datos en general

Características más relevantes:

- Acceso gratuito a **GPUs y TPUs**
- Colaboración en tiempo real
- Integración con Google Drive

### ¿Para qué se usa Google Colab? ▾

Con Google Colab podemos explorar un sin fin de posibilidades referentes a la Ciencia de Datos:

- **Análisis de datos**
- **Visualización de datos**
- **Desarrollo y entrenamiento de modelos de *machine learning***
- **Pruebas de código Python**
- **Educación en programación y ciencia de datos**

### **Novedades en 2025** ▾

Desde **Google I/O 2025**, Colab ha implementado un enfoque innovador centrado en la inteligencia artificial, presentando un *Compañero de Codificación* especializado en tareas de análisis de datos. Las novedades principales son:

- ☆ Se posiciona como una plataforma con enfoque en IA: compañero inteligente de codificación, bajo el concepto **"AI-first"**.
- 🧠 Incorpora *Gemini 2.5 Flash*, potenciando así su Agente de Ciencia de Datos (DSA).

La característica destacada es el **DSA**, que automatiza procesos completos desde la carga de datos hasta la creación de gráficos mediante instrucciones en lenguaje natural.

## 🧠 El Agente de Ciencia de Datos (DSA) ▾

El **DSA** automatiza la creación de notebooks Jupyter completos. Los usuarios pueden cargar un archivo y usar *prompts* en lenguaje natural para generar análisis completos.

### 🔗 Funcionalidades clave >

- **Carga de Datos:** Importa archivos CSV, JSON y .txt (hasta 1GB) en un dataframe.
- **Exploración de Datos (EDA):** Estadísticas descriptivas, tipos de datos y verificación de valores perdidos.
- **Limpieza de Datos:** Manejo de valores perdidos, duplicados e inconsistencias; genera código o limpia directamente el archivo.
- **Análisis de Datos:** Identificación de tendencias, promedios y cambios significativos.
- **Generación de Visualizaciones:** Código en Matplotlib y otras librerías para visualizaciones rápidas.

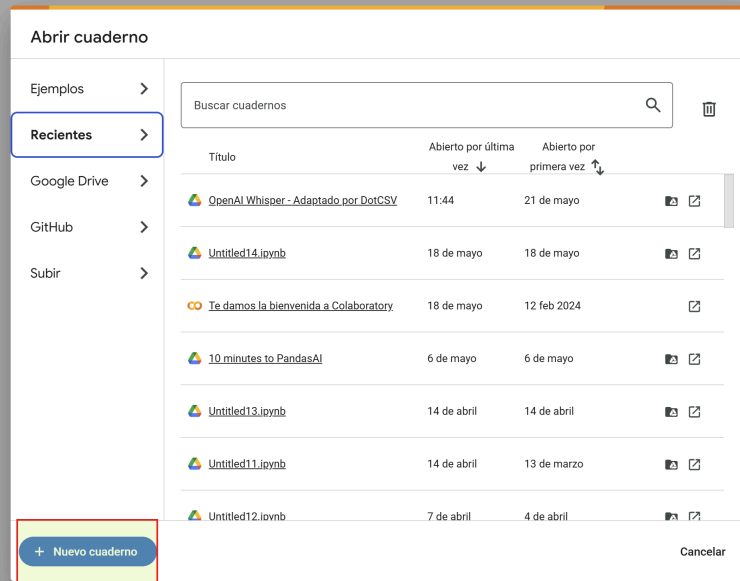
El DSA **opera a nivel de celda y de notebook completo**. Puede **auto-depurar errores** y explicar el problema junto con soluciones sugeridas. Una característica avanzada es la generación de **flujos analíticos autónomos**, donde planea, ejecuta, interpreta y presenta resultados con posibilidad de retroalimentación.

## 🧪 Tutorial: Analiza tus datos usando el agente de Colab

☰ Sigue estos pasos para comenzar a analizar tus datos ▾

### 📌 Paso 1: Accede a Google Colab

1. Visita 🖱️ <https://colab.research.google.com>
2. Inicia sesión con tu cuenta de Google
3. Haz clic en **Nuevo cuaderno**



## Paso 2: Abre el agente de Google Colab

### ¿Dónde está el agente?

Untitled1.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Comandos + Código + Texto

Volver a conectar

**Task**

Haz un análisis exploratorio de datos

Here is all the data you need: "Student Depression Dataset.csv"

**Data loading**

Subtask:

Load the "Student Depression Dataset.csv" file.

Reasoning: Load the "Student Depression Dataset.csv" file into a pandas DataFrame and display the first few rows.

```
[ ] import pandas as pd
df = pd.read_csv("Student Depression Dataset.csv")
display(df.head())
```

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Financial Stress	Family History of Mental Illness	Depression
0	2	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.0	0.0	5-6 hours	Healthy	B.Pharm	Yes	3.0	1.0	No	1
1	8	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.0	0.0	5-6 hours	Moderate	BSc	No	3.0	2.0	Yes	0
2	26	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.0	0.0	Less than 5 hours	Healthy	BA	No	9.0	1.0	Yes	0
3	30	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.0	0.0	7-8 hours	Moderate	BCA	Yes	4.0	5.0	Yes	1

Variables Terminal

12:11

## Paso 3: Carga tus datos

The screenshot shows the Google Colab interface. On the left, a Jupyter Notebook titled 'Untitled1.ipynb' is open. The 'Task' section contains the instruction: 'Haz un análisis exploratorio de datos. Here is all the data you need: "Student Depression Dataset.csv"'. The 'Data loading' section has a subtask: 'Load the "Student Depression Dataset.csv" file.' and a reasoning prompt: 'Load the "Student Depression Dataset.csv" file into a pandas DataFrame and display the first few rows.' Below this, a code cell contains the following Python code:

```
[ ] import pandas as pd
df = pd.read_csv("Student Depression Dataset.csv")
display(df.head())
```

The code cell is executed, and the output shows the first few rows of the 'Student Depression Dataset.csv' file. The table has columns: id, Gender, Age, City, Profession, Academic Pressure, Work Pressure, CGPA, Study Satisfaction, Job Satisfaction, and Sleep Duration. The first few rows are:

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration
0	2	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.0	0.0	5-6 hours
1	8	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.0	0.0	5-6 hours
2	26	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.0	0.0	Less than 5 hours
3	30	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.0	0.0	7-8 hours

On the right side of the interface, the Gemini chat window is open. It displays the Gemini logo and a message: 'Gemini es una potente herramienta de IA desarrollada por Google que te ayuda a usar Colab. ¿No sabes qué preguntar? Prueba con una de las peticiones que aparecen a continuación'. Below this, there are three buttons: 'How do I filter a Pandas DataFrame?', 'How can I create a plot in Colab?', and 'Show me a list of publicly available datasets'. At the bottom of the chat window, there is a text input field with the placeholder 'Introduce una petición aquí' and a 'Subir' button. The 'Subir' button is highlighted with a red box.

## Paso 4: Escribe un prompt sobre los análisis que quieres realizarle a tus datos e ingresalo

Por ejemplo (ejemplo básico):

Haz un análisis de los datos que acabo de cargar, basado en los siguientes puntos:

- Carga el archivo en un DataFrame de pandas y muestres las primeras filas para entender la estructura.
- Verifiques si hay valores faltantes y ajustes los tipos de datos según corresponda (fechas y numéricos).
- Obtengas estadísticas descriptivas básicas para las columnas numéricas.
- Genere algunas visualizaciones iniciales (como histogramas o gráficos de barras) para ver la distribución y tendencias.
- Resumas los hallazgos principales y sugieras posibles próximos pasos o áreas de interés para profundizar.

## Tutorial: Usando el DSA a nivel de celda

 DSA a nivel de celda: Control granular 


 ¿Cuándo usar DSA a nivel de celda?

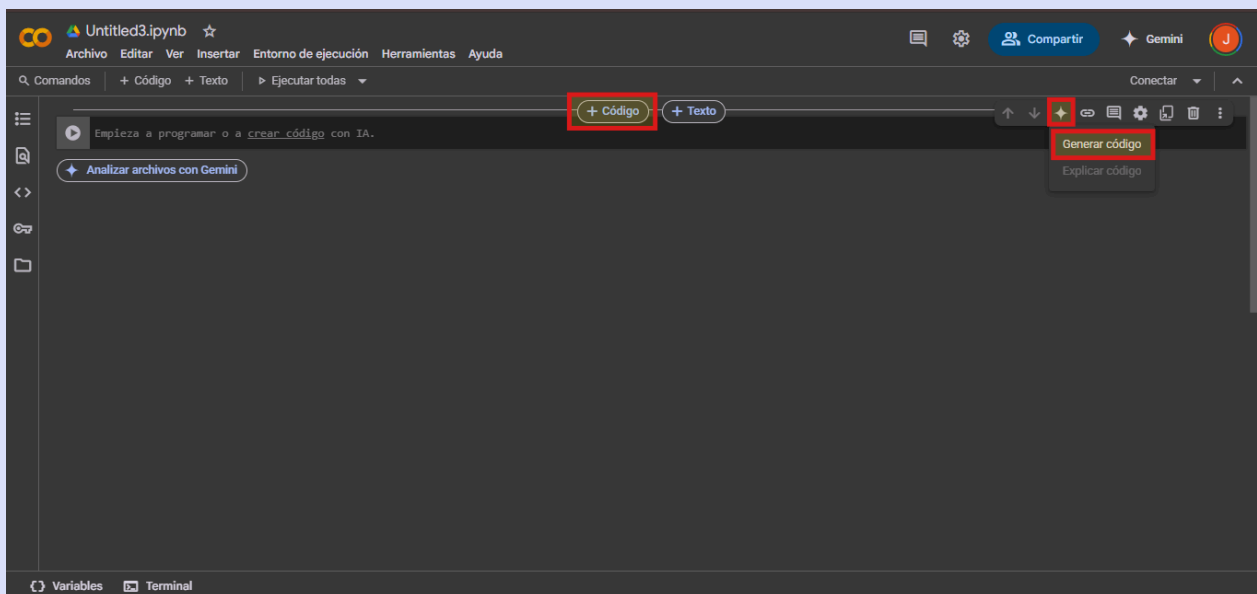
El DSA a nivel de celda es ideal cuando:

- Quieres **control específico** sobre cada paso del análisis
- Necesitas **modificar código** generado antes de continuar
- Trabajas con **múltiples datasets** en el mismo notebook
- Quieres **combinar código manual** con código generado por IA

## Paso a paso para DSA a nivel de celda

### Paso 1: Crear una celda nueva >

1. Haz clic en **+ Código** para crear una nueva celda
2. En lugar de escribir código, haz clic en el **ícono del DSA** () que aparece al lado de la celda



### Paso 2: Escribir prompt específico Escribe un prompt enfocado en una tarea específica: >

Carga el archivo CSV que subí y muestra un resumen estadístico básico de las columnas numéricas

O para análisis más específicos:

Crea un gráfico de dispersión entre las variables 'edad' y 'salario' del dataset, añadiendo una línea de tendencia

### Paso 3: Revisar y ejecutar >

3. El DSA generará el código en la celda

4. **Revisa el código** antes de ejecutarlo
5. **Modifica si es necesario** (puedes editar manualmente)
6. Ejecuta la celda con `Shift + Enter`

#### Paso 4: Iterar y refinar >

- Crea nuevas celdas para **análisis adicionales**
- **Combina celdas** generadas por DSA con código manual

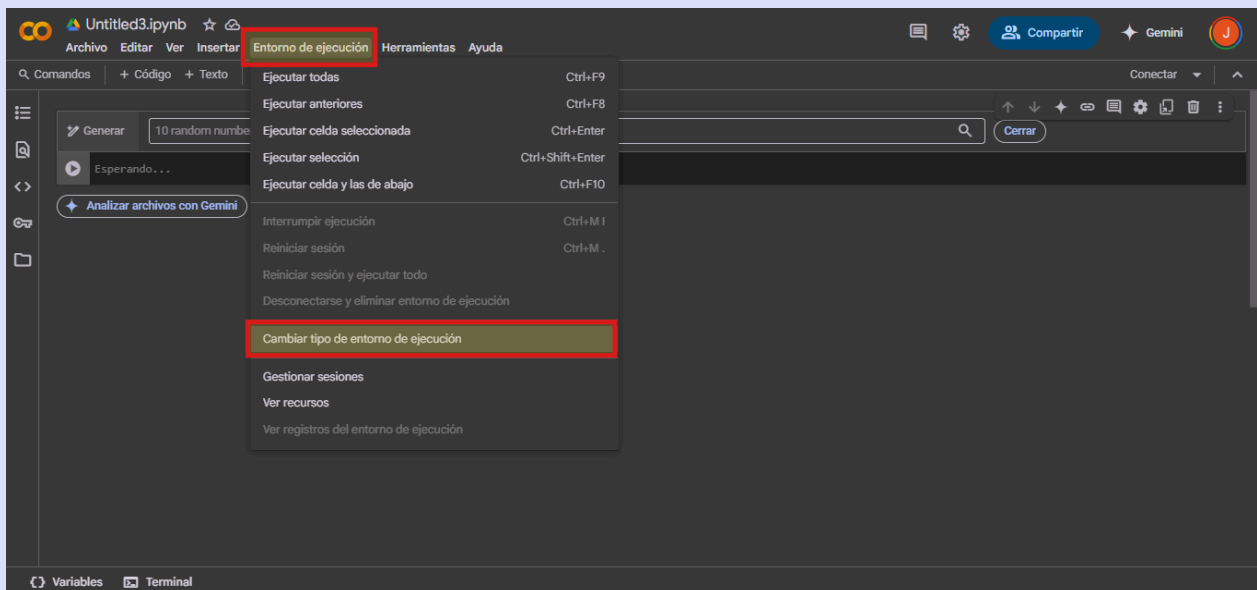
## Tutorial: Cambiar el entorno de ejecución a R

### Configurando R en Google Colab >

#### Paso 1: Cambiar el tipo de entorno de ejecución

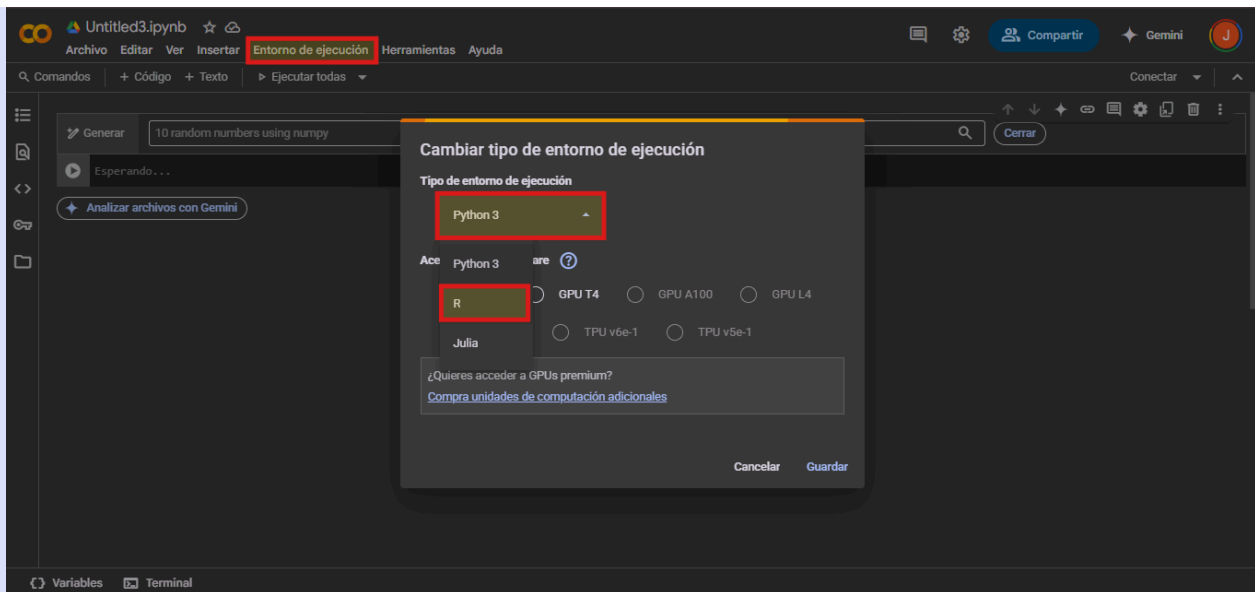
##### Acceder a la configuración >

1. Ve al menú `Entorno de ejecución` en la barra superior
2. Selecciona `Cambiar tipo de entorno de ejecución`



##### Configurar R >

3. En la ventana que se abre, cambia `Tipo de entorno de ejecución` de **Python 3** a **R**
4. Puedes mantener las demás configuraciones (GPU, TPU, etc.)
5. Haz clic en `Guardar`



⚠ **¡Importante! Al cambiar a R, el notebook se reiniciará y perderás las variables en memoria. Asegúrate de guardar tu trabajo antes de cambiar.**

## Tutorial: DSA con R - Limitaciones y uso recomendado

### ⚠ Limitaciones importantes del DSA con R ▾

#### ⚠ DSA de notebook completo con R

##### ⚠ ¿Por qué no funciona bien? >

El DSA de notebook completo **NO** funciona eficientemente con R debido a:

- **Incompatibilidades** en la interpretación de sintaxis R
- **Limitaciones** en la generación automática de código R complejo
- **Errores frecuentes** en la secuencia de análisis completos
- **Falta de optimización** del modelo para flujos de trabajo en R

#### ✅ DSA a nivel de celda con R

##### ✅ ¿Cuándo **SÍ** funciona bien? El **DSA a nivel de celda** funciona correctamente con R ▾ para:

- **Tareas específicas** y bien definidas
- **Análisis puntuales** (estadísticas, gráficos simples)
- **Transformaciones de datos** básicas

- **Visualizaciones** individuales

⚠ **Para usar el DSA a nivel de celda con R, debes especificar explícitamente en el prompt que use R**

## 🔧 Tutorial paso a paso: DSA con R a nivel de celda

☰ [Guía completa para usar DSA con R](#) ▾

### 🚀 Configuración inicial

#### 🔗 Paso 1: Preparar el entorno R >

1. **Cambia a entorno R** (siguiendo el tutorial anterior)
2. **Carga tus datos** usando el método tradicional de Colab
3. **Verifica** que los datos se cargaron correctamente:

```
# Verificar archivos disponibles
list.files()
```

#### 🔗 Paso 2: Cargar librerías básicas (manual) Es recomendable cargar las librerías manualmente antes de usar DSA: >

```
# Librerías esenciales para análisis de datos
library(readr)      # Para leer archivos
library(dplyr)      # Para manipulación de datos
library(ggplot2)    # Para visualizaciones
library(summary)    # Para estadísticas descriptivas
```

### 📊 Usando DSA para análisis específicos

#### 🔗 Prompts efectivos para R >

**Para carga de datos:**

```
Lee el archivo CSV llamado 'datos.csv' y muéstralo como un dataframe en R
```

**Para estadísticas descriptivas:**



Calcula estadísticas descriptivas (media, mediana, desviación estándar) para todas las columnas numéricas del dataframe

#### Para visualizaciones:

Crea un histograma de la variable 'edad' usando ggplot2, con título y etiquetas en español

#### Para análisis de correlación:

Genera una matriz de correlación entre las variables numéricas y visualízala como un mapa de calor

#### ⚠ Mejores prácticas con R y DSA >

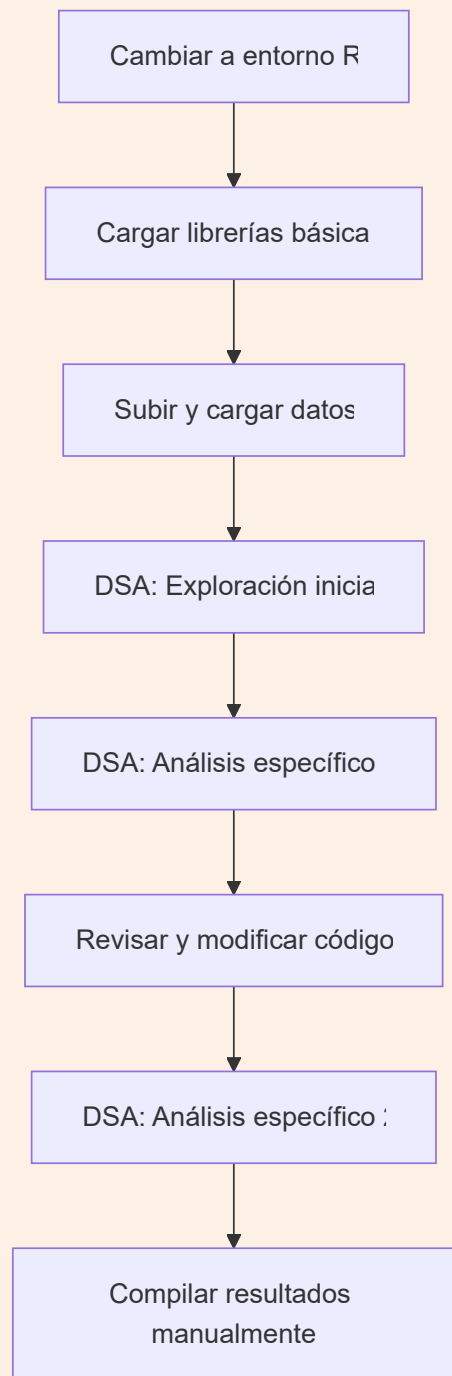
##### ✅ Haz esto:

- Usa **prompts específicos** y claros
- **Revisa el código** antes de ejecutar
- **Carga librerías manualmente** cuando sea necesario
- **Trabaja celda por celda** para mayor control

##### ❌ Evita esto:

- Prompts muy **complejos o ambiguos**
- **Múltiples tareas** en un solo prompt
- Confiar en DSA para **flujos completos** en R
- **Análisis muy específicos** sin contexto previo

#### 🤔 ¿Flujo de trabajo recomendado con R? ▾



## Ventajas de usar Colab frente a un chatbot impulsado por LLMs

### ✓ ¿Por qué elegir Colab? ✓

1. **Transparencia:** Ves el código que se ejecuta, lo puedes revisar y aprender de él.
2. **Modificabilidad:** Puedes editar el código a tu gusto, ajustar parámetros, depurar errores, etc.

🔥 ¡Juega y explora! Prueba diferentes prompts, modifica el código, cambia los gráficos. ¡La mejor forma de aprender es explorando!

---

## Referencias y recursos adicionales

- [Google I/O 2025: An AI-first reimagining of Google Colab](#)
- [Data Analysis and Visualization with the Google Colab Data Science Agent](#)
- [New productivity-boosting capabilities in Colab Enterprise](#)
- [Google I/O 2025: Keynote highlights from Colab, AI, and more](#)
- [Google Colab's AI advancements: A game-changer for data science](#)
- [Reddit discussion: Google Gemini in Colab fail pretty bad](#)
- [Reddit discussion: What is the best AI tool for coding](#)
- [Databricks Assistant: Conversational AI for Data and ML](#)
- [Boost your BI with the AI/BI Genie in Databricks](#)
- [AWS Documentation: Amazon Q Developer in SageMaker Canvas](#)
- [Introducing Copilot for Data Science and Data Engineering in Microsoft Fabric](#)