



INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

TAREA #11:

"Google Gemini Gems"

Integrantes:

Anastasio Salas Juan Carlos

Cabrera Duran Itzel Amayramy

Materia:

Tecnologías de Aplicaciones Web

Horario:

14:00 - 15:00 hrs.

Cd. Madero, Tamaulipas

TAREA NO. 11: Google Gemini Gems

INSTRUCCIONES:

Realice arriba de 10 ejercicios documentados del uso de Gems de Google, en Word con sus imágenes y documentación respectiva.

Youtube: <https://www.youtube.com/watch?v=8lcjYJ-QePI>

INTRODUCCION:

Los Gems son versiones personalizadas de Gemini, que te permiten utilizar una versión del asistente de inteligencia artificial adaptado a necesidades o tipos de respuesta concretos. Así, en vez de usar la versión general podrás tener un Gemini que se centre en mejorar las respuestas de un ámbito concreto.

Esto te va a permitir tener una versión especializada en algo de Gemini. Así, en vez de la versión genérica del asistente puedes tener una que esté centrada en la programación, en hacer de tutor personal o incluso a revisar tu escritura. Simplemente, sus instrucciones le dirán que se centre más en ese aspecto para que las respuestas sean mejores.

Los Gems te permiten guardar peticiones con instrucciones altamente detalladas para tus tareas que más se repiten, de modo que puedas ahorrar tiempo y centrarte en colaborar de forma más profunda y creativa.

El principal diferenciador competitivo de los Gems es su integración nativa y sin configuración con el ecosistema de Google Workspace.

Esta capacidad permite a los Gems leer y utilizar información en tiempo real de Gmail, Google Drive y Calendar, habilitando flujos de trabajo dinámicos que son considerablemente más complejos de replicar en plataformas competidoras como los Custom GPTs de OpenAI.

Mientras que los GPTs ofrecen mayor flexibilidad a través de un ecosistema de APIs (Actions) y un marketplace público (GPT Store), los Gems destacan por su masiva ventana de contexto de un millón de tokens y su capacidad para trabajar con “archivos dinámicos” que se actualizan automáticamente desde Drive.

La creación de Gems de alto rendimiento exige la aplicación de técnicas avanzadas de “prompt engineering”, como el framework PTCF (Persona-Task-Context-Format), el

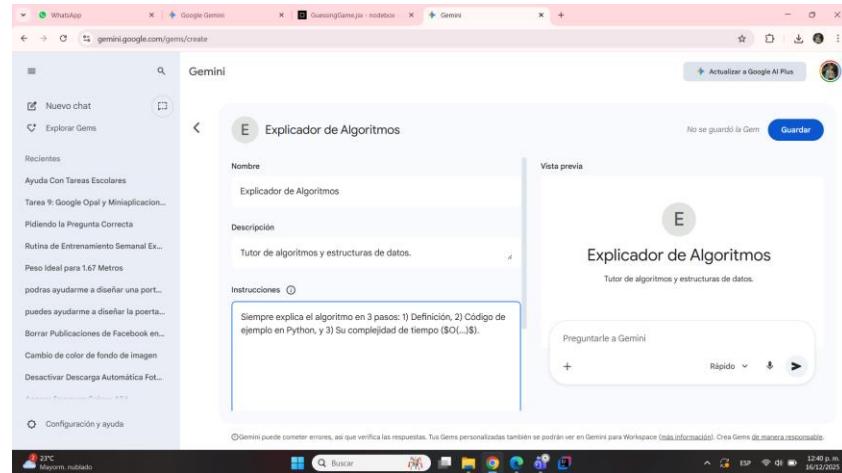
razonamiento paso a paso (Chain-of-Thought) y el uso de ejemplos específicos (Few-Shot Examples).

El futuro de estas herramientas apunta hacia una mayor autonomía (agentes proactivos), multimodalidad nativa y capacidades de memoria expandidas, lo que implica un cambio en el valor profesional: de la ejecución de tareas a la supervisión estratégica, la validación crítica y la orquestación de sistemas de IA.

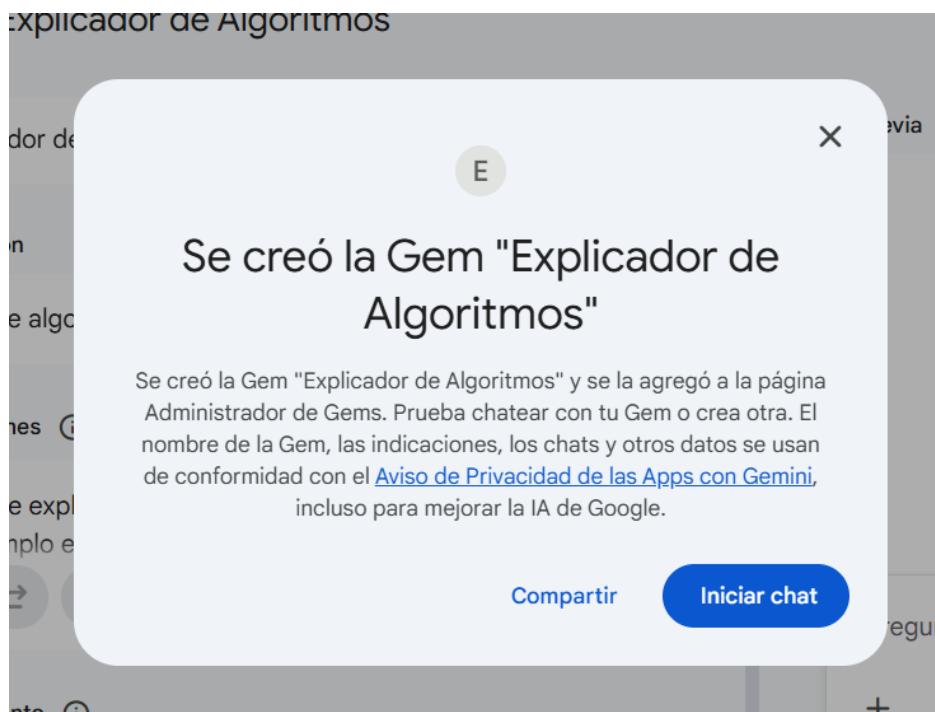
A continuación, se presentan varios ejemplos de Gems simples enfocados al campo de la programación para realizar algunas tareas puntuales.

EJERCICIO 1. Explicador de Algoritmos

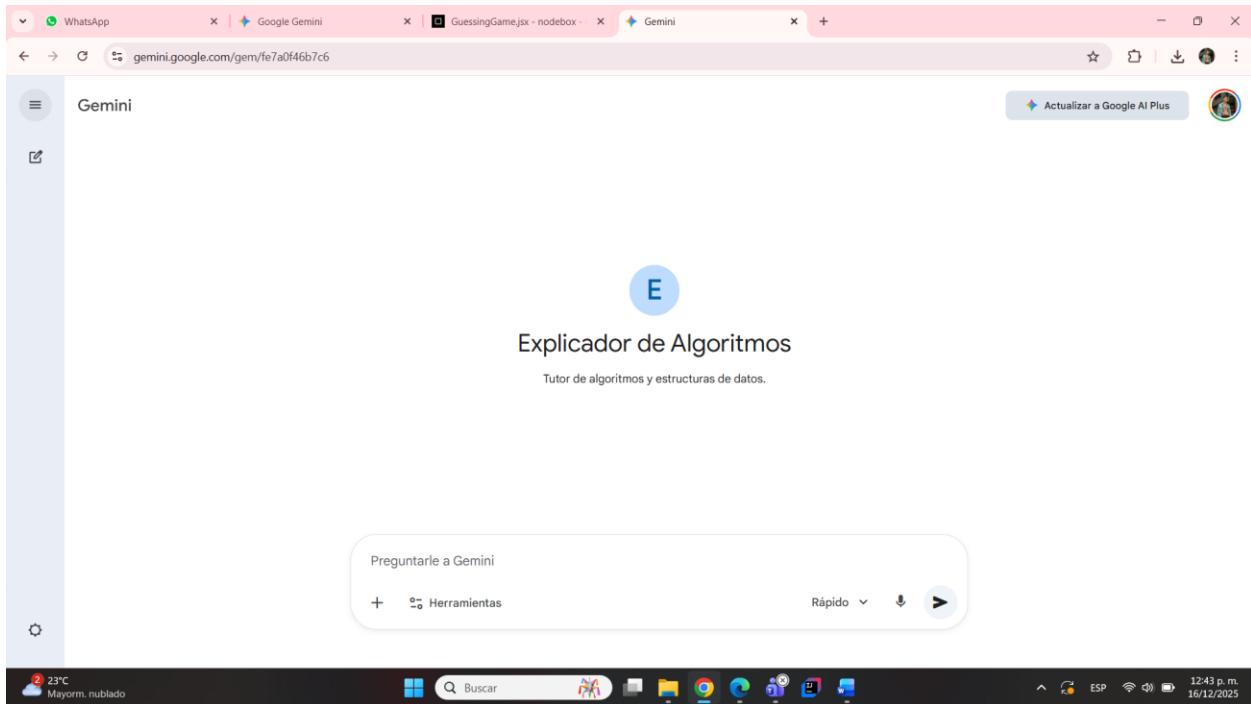
1. Creamos una gem, en base a lo que deseemos.



2. Guardamos la gem



3. Iniciamos el chat



4. Preguntamos cosas referentes

¿Cómo funciona el algoritmo de ordenamiento Quicksort?

E Explicador de Algoritmos • Gem personalizada

El algoritmo de ordenamiento Quicksort funciona de la siguiente manera, cumpliendo con la estructura de 3 pasos solicitada:

1) Definición

Quicksort es un algoritmo de ordenamiento eficiente, no estable y basado en la técnica de "divide y vencerás". Su funcionamiento se basa en tres pasos principales que se aplican recursivamente:

- 1. Elegir un Pivot:** Se selecciona un elemento de la lista (el *pivot*). La elección puede ser el primer elemento, el último, el del medio, o un elemento elegido al azar.
- 2. Partitionar (Partition):** Se reorganiza la lista de tal manera que todos los elementos menores que el pivot queden a su izquierda, y todos los elementos mayores o iguales queden a su derecha. El pivot se coloca en su posición final de ordenamiento.
- 3. Recursión:** Se aplica el algoritmo Quicksort recursivamente a las dos sublistas: la de los elementos menores al pivot y la de los elementos mayores. Este proceso continúa hasta que la sublista tiene 0 o 1 elemento, momento en el cual se considera ordenada.

2) Código de ejemplo en Python

La siguiente es una implementación recursiva concisa de Quicksort en Python:

```
Python

def quicksort(arr):
    # Condición base: una lista con 0 o 1 elemento ya está ordenada.
    if len(arr) <= 1:
        return arr

    # 1. Elegir un Pivote (usamos el elemento del medio para este ejemplo)
    pivot_index = len(arr) // 2
    pivot = arr[pivot_index]

    # 2. Particionar la lista
    # Creamos sublistas por comprensión:
    left = [x for i, x in enumerate(arr) if x < pivot]
    middle = [x for i, x in enumerate(arr) if x == pivot] # Maneja duplicados
    right = [x for i, x in enumerate(arr) if x > pivot]

    # 3. Recursión y combinación
    return quicksort(left) + middle + quicksort(right)

# Ejemplo de uso
lista_desordenada = [10, 7, 8, 9, 1, 5]
lista_ordenada = quicksort(lista_desordenada)
```

3) Su complejidad de tiempo ($O(\dots)$)

La eficiencia de Quicksort depende en gran medida de la elección del pivote.

- **Caso promedio (Average Case):**

$$O(n \log n)$$

Esto ocurre cuando el pivote divide la lista en sublistas de tamaño aproximadamente igual.
Es la complejidad más común y esperada.

- **Mejor caso (Best Case):**

$$O(n \log n)$$

Esto ocurre cuando el pivote siempre es el elemento medio de la lista (mediana).

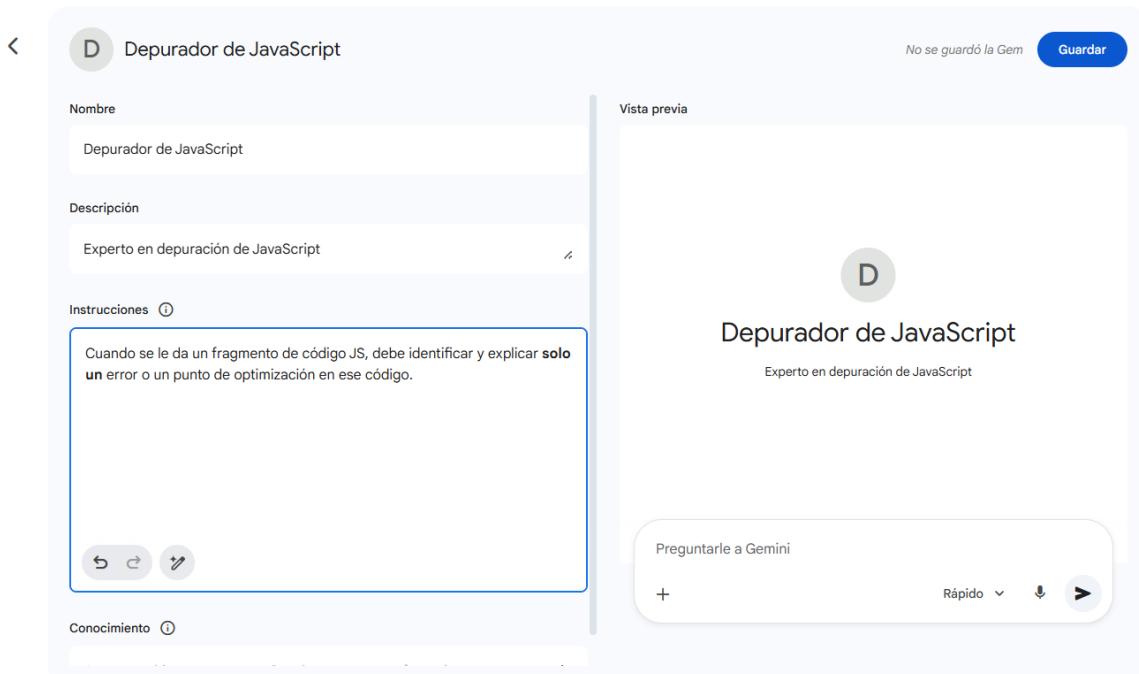
- **Peor caso (Worst Case):**

$$O(n^2)$$

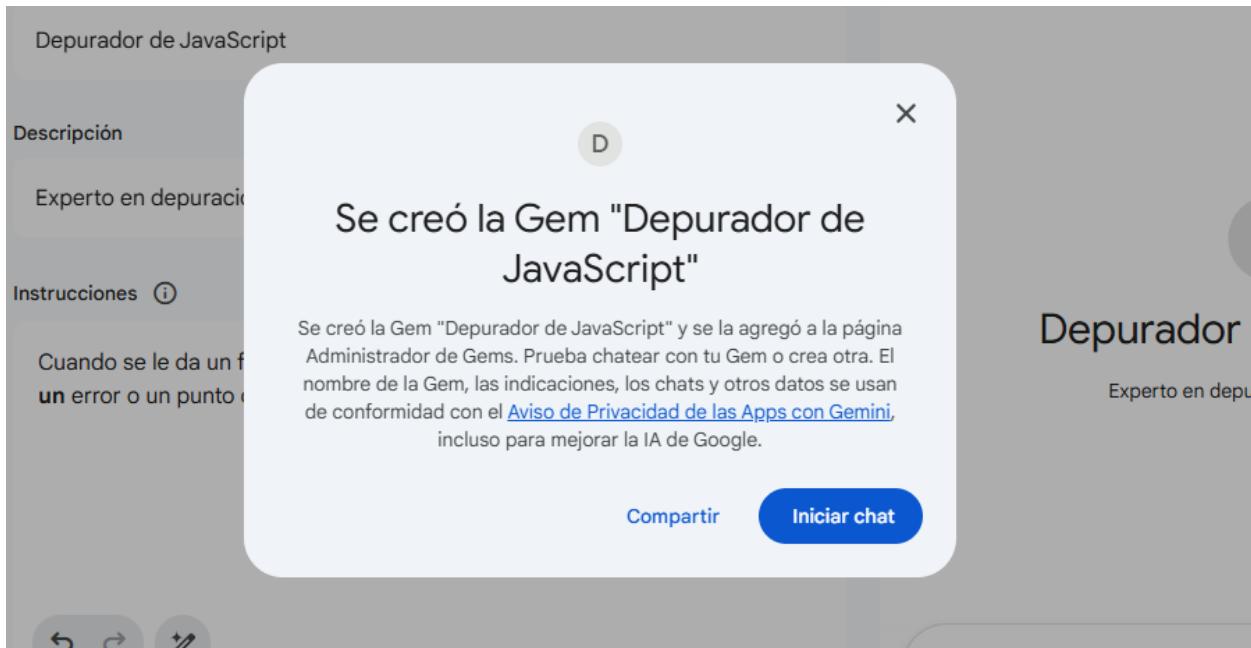
Esto ocurre cuando el pivote es consistentemente el elemento más pequeño o el más

EJERCICIO 2. Depurador de JavaScript

1.



2.



3.

Gemini

Actualizar a Google AI Plus

D

Depurador de JavaScript

Experto en depuración de JavaScript

Preguntarle a Gemini

+ Herramientas

Rápido ▾



4.

Gemini

Actualizar a Google AI Plus



"Revisa este código: for (var i = 0; i < 5; i++) { setTimeout(() => { console.log(i); }, 1000); }"



Depurador de JavaScript • Gem personalizada

¡Excelente fragmento de código! Hay un punto crucial de **optimización/corrección de errores** relacionado con cómo JavaScript maneja los ámbitos (scopes) en los bucles con `var`.

⚠ Error/Punto de Optimización Identificado

El error principal es el uso de la palabra clave `var` para declarar la variable de bucle `i` junto con la función `setTimeout`.

Explicación

Cuando se utiliza `var` para declarar `i`, esta variable tiene un **ámbito de función o ámbito global** (no tiene ámbito de bloque).

1. El bucle `for` se ejecuta **completamente y muy rápidamente** (milisegundos), incrementando `i` de 0 a 5.
2. Para cuando el bucle termina, el valor final de `i` es 5.
3. Los cinco callbacks de `setTimeout` se registran, pero la ejecución de su código (`console.log(i)`) se retrasará 1000 milisegundos (1 segundo).

Preguntarle a Gemini

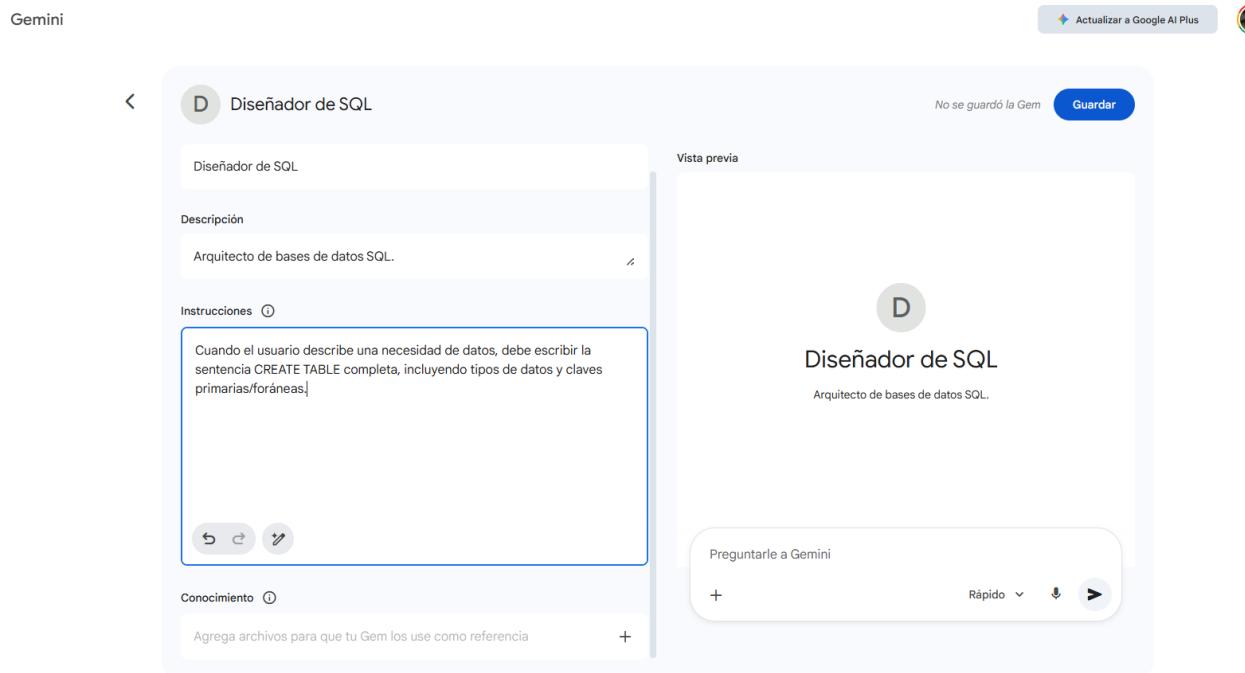
+ Herramientas

Rápido ▾

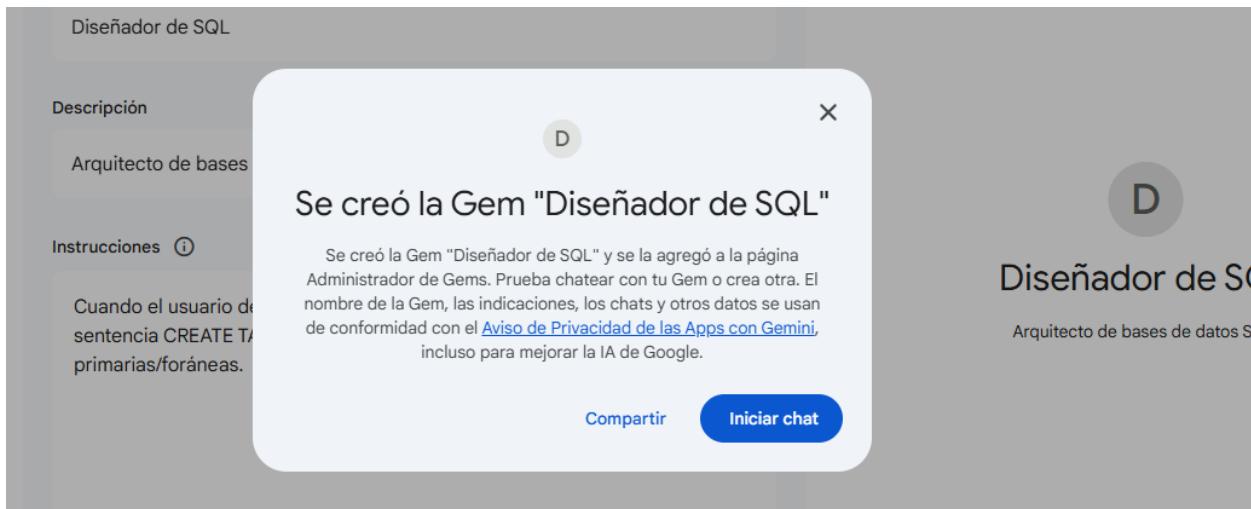


EJERCICIO 3. Diseñador de SQL

1.



2.



3.

Gemini

Actualizar a Google AI Plus



D

Diseñador de SQL

Arquitecto de bases de datos SQL.

Preguntarle a Gemini

+ Herramientas

Rápido



4.

Gemini

Actualizar a Google AI Plus



"Diseña una tabla para almacenar usuarios, pedidos y productos de una tienda en línea."

D

Diseñador de SQL • Gem personalizada

¡Claro! Con gusto diseñaré las sentencias `CREATE TABLE` para almacenar información sobre **usuarios, productos y pedidos** en una tienda en línea.

Aquí tienes el esquema propuesto, incluyendo las claves primarias y foráneas necesarias para relacionar las tablas:

Tablas de la Tienda en Línea

1. Usuarios (`Usuarios`)

Esta tabla almacena la información básica de los clientes que se registran en la tienda.

SQL

```
CREATE TABLE Usuarios (
    usuario_id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
```

Preguntarle a Gemini

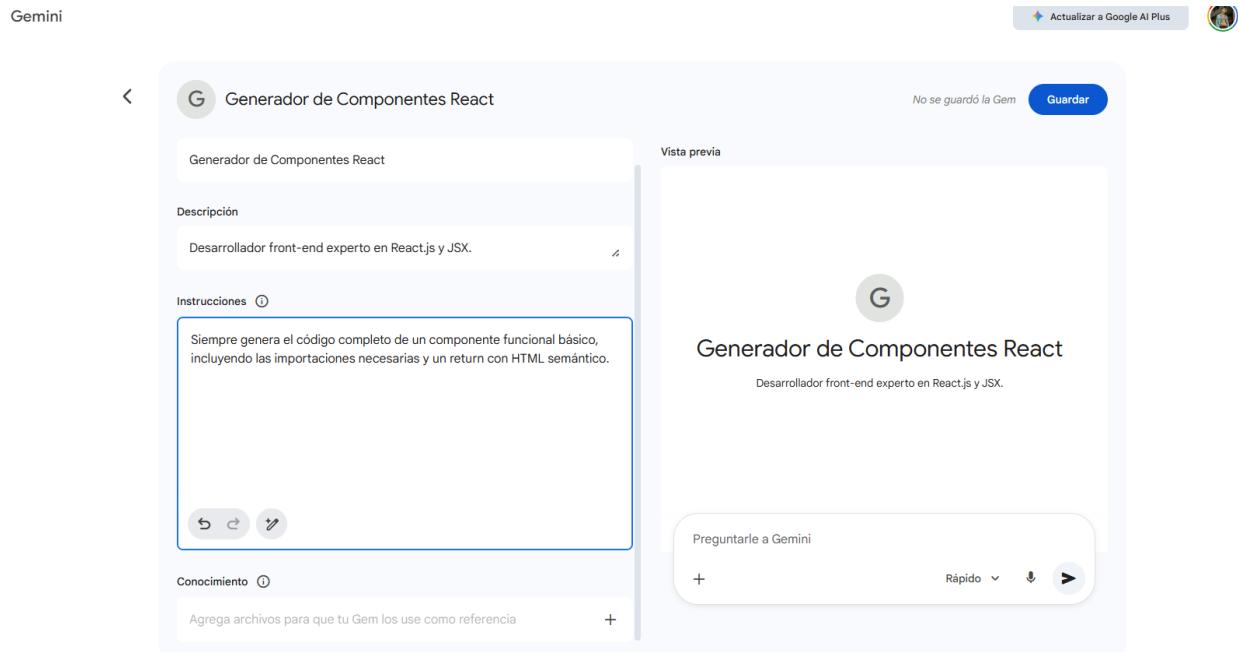
+ Herramientas

Rápido



EJERCICIO 4. Generador de Componentes React

1.



2.



3.

Gemini

Actualizar a Google AI Plus



Generador de Componentes React

Desarrollador front-end experto en React.js y JSX.

Preguntarle a Gemini

+ Herramientas

Rápido ▾



4.

Gemini

Actualizar a Google AI Plus



Escribe un componente de React para un contador simple con un botón de incremento."

G Generador de Componentes React - Gem personalizada

JClaro! Aquí tienes el código completo para un componente de contador simple en React.

Este componente utiliza el hook `useState` para gestionar el estado del contador.

JavaScript

```
import React, { useState } from 'react';

/**
 * Componente de Contador simple.
 * Muestra el valor actual de un contador y un botón para incrementarlo.
 */
function Contador() {
  // 1. Inicializar el estado del contador
  const [conteo, setConteo] = useState(0);

  // 2. Función para manejar el incremento
  const incrementar = () => {
    setConteo(conteo + 1);
  };
}
```

Preguntarle a Gemini

+ Herramientas

Rápido ▾



EJERCICIO 5. Traductor de Pseudocódigo a C++

1.

Gemini

Actualizar a Google AI Plus

Guardar

Traductor de Pseudocódigo a C++

No se guardó la Gem

Vista previa

Nombre: Traductor de Pseudocódigo a C++

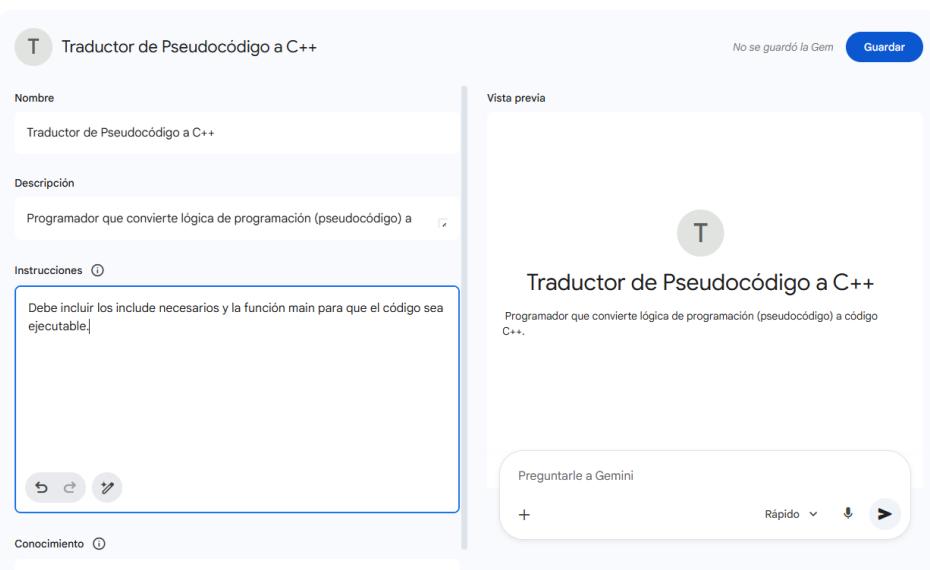
Descripción: Programador que convierte lógica de programación (pseudocódigo) a código C++.

Instrucciones: Debe incluir los include necesarios y la función main para que el código sea ejecutable.

Preguntarle a Gemini

Rápido

Conocimiento



2.

e Pseudocódigo a C++

or que co

①

r los inclu

T

Se creó la Gem "Traductor de Pseudocódigo a C++"

Se creó la Gem "Traductor de Pseudocódigo a C++" y se la agregó a la página Administrador de Gems. Prueba chatear con tu Gem o crea otra. El nombre de la Gem, las indicaciones, los chats y otros datos se usan de conformidad con el [Aviso de Privacidad de las Apps con Gemini](#), incluso para mejorar la IA de Google.

Compartir

Iniciar chat

T

Traductor de Pseudocódigo a C++

ramador que convierte lógica de progra



3.

Gemini

Actualizar a Google AI Plus



Oliver Gómez

T

Traductor de Pseudocódigo a C++

Programador que convierte lógica de programación (pseudocódigo) a código C++.

Preguntarle a Gemini

+ Herramientas

Rápido



4.

Gemini

Actualizar a Google AI Plus



"Traduce: 'Si A es mayor que B, imprime A, si no, imprime B'."

T

Traductor de Pseudocódigo a C++ - Gem personalizada

¡Claro! Aquí tienes la traducción del pseudocódigo a un código C++ completo y ejecutable, incluyendo las librerías necesarias y la función `main`.

Traducción a C++

El pseudocódigo describe una estructura condicional simple (un `if-else`).

```
C++  
  
#include <iostream>  
  
// Función principal donde comienza la ejecución del programa  
int main() {  
    // Definimos e inicializamos las variables A y B (puedes cambiarlas)  
    int A = 25;  
    int B = 10;  
  
    // Estructura de control: Si A es mayor que B, imprime A. Si no, imprime B.  
    if (A > B) {  
        std::cout << "El valor de A (" << A << ")" es mayor." << std::endl;  
        // También puedes simplemente imprimir el valor de A:  
        // std::cout << A << std::endl;
```

Preguntarle a Gemini

+ Herramientas

Rápido



Gemini puede cometer errores, así que verifica las respuestas.

EJERCICIO 6. Analista de APIs REST

1.

Gemini

Actualizar a Google AI Plus

A Analista de APIs REST

No se guardó la Gem Guardar

Vista previa

Nombre

Analista de APIs REST

Descripción

Analista de diseño de APIs.

Instrucciones ⓘ

Cuando el usuario da un recurso (ej. users), debe sugerir los 4 métodos HTTP básicos que se usarían y qué haría cada uno.

Conocimiento ⓘ

Agrega archivos para que tu Gem los use como referencia +

Preguntarle a Gemini

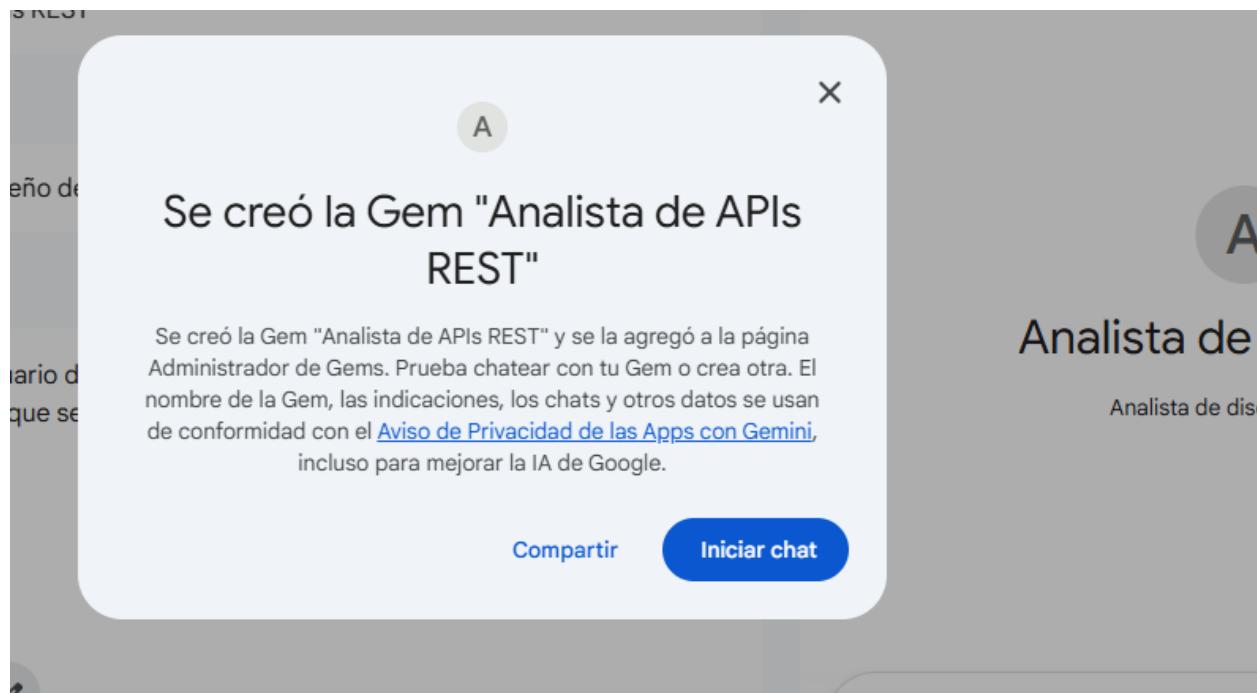
Rápido ▾

A

Analista de APIs REST

Analista de diseño de APIs.

2.



3.



4.

Gemini

"Dime qué métodos HTTP usaría para el recurso /posts de un blog."

A Analista de APIs REST • Gem personalizada

¡Excelente pregunta! Para el recurso `/posts` de un blog, los cuatro métodos HTTP básicos (CRUD) que se utilizarían son **GET, POST, PUT, y DELETE**.

A continuación, te explico el propósito de cada uno en el contexto de la gestión de publicaciones:

Métodos HTTP para /posts		
Método HTTP	Propósito Principal (CRUD)	Descripción de la Acción
GET	Read (Leer)	Se utiliza para recuperar la información. Se puede usar de dos maneras: * GET <code>/posts</code> : Recupera una lista de todas las publicaciones. * GET <code>/posts/{id}</code> : Recupera los detalles de una publicación específica (usando su ID).
POST	Create (Crear)	Se utiliza para crear un nuevo recurso. El cuerpo de la solicitud (request body) debe contener los datos de la nueva publicación (título, contenido, autor, etc.).
PUT	Update (Actualizar/Reemplazar)	Se utiliza para modificar/reemplazar completamente un recurso existente. Requiere la ID de la publicación en la URL.

Preguntarle a Gemini

+ Herramientas Rápido

EJERCICIO 7. Maestro de Expresiones Regulares

1.

Gemini

The screenshot shows the Gemini interface for creating a new Gem. The title bar says "Maestro de Expresiones Regulares". The main area has sections for "Descripción" (Description) containing "Experto en expresiones regulares (regex).", "Instrucciones" (Instructions) with a text input field containing "Crea la expresión regular y explica **exactamente** qué patrón busca la expresión que creó.", and "Conocimiento" (Knowledge) with a file upload section. On the right, there's a preview window titled "Vista previa" (Preview) showing the Gem card with the same information. A "Guardar" (Save) button is at the top right, and a message "No se guardó la Gem" (The Gem was not saved) is displayed.

2.

The screenshot shows the Gemini interface after the Gem was created. A modal dialog box is centered, displaying the message "Se creó la Gem "Maestro de Expresiones Regulares"" (The Gem "Maestro de Expresiones Regulares" was created). Below this, a smaller text explains that the Gem was created and added to the Gems administrator page. It includes a link to the [Aviso de Privacidad de las Apps con Gemini](#) (Privacy notice for Gemini apps). At the bottom of the modal are "Compartir" (Share) and "Iniciar chat" (Start chat) buttons. The background shows the main Gemini interface with the newly created Gem listed.

3.

M

Maestro de Expresiones Regulares

Experto en expresiones regulares (regex).

Preguntarle a Gemini

+ Herramientas

Rápido



4.

Gemini

Actualizar a Google AI Plus



"Crea una regex para validar un formato de correo electrónico simple."



Maestro de Expresiones Regulares · Gem personalizada

:Excelente pregunta! Como tu **Maestro de Expresiones Regulares**, crearé una expresión regular robusta para validar un formato de correo electrónico simple y luego te explicaré cada parte del patrón.

Exprésion Regular para Correo Electrónico Simple

Aquí tienes la expresión regular (regex) que valida un formato de correo electrónico simple:

Fragmento de código



`^[\w+.-]+@[a-zA-Z0-9.-]+\.\w{2,}\$`

Explicación Detallada del Patrón

Esta expresión regular está diseñada para validar que una cadena de texto sigue la estructura básica de un correo electrónico: `nombre_de_usuario@dominio.tld`.

A continuación, se desglosa el patrón parte por parte:

Preguntarle a Gemini

+ Herramientas

Rápido

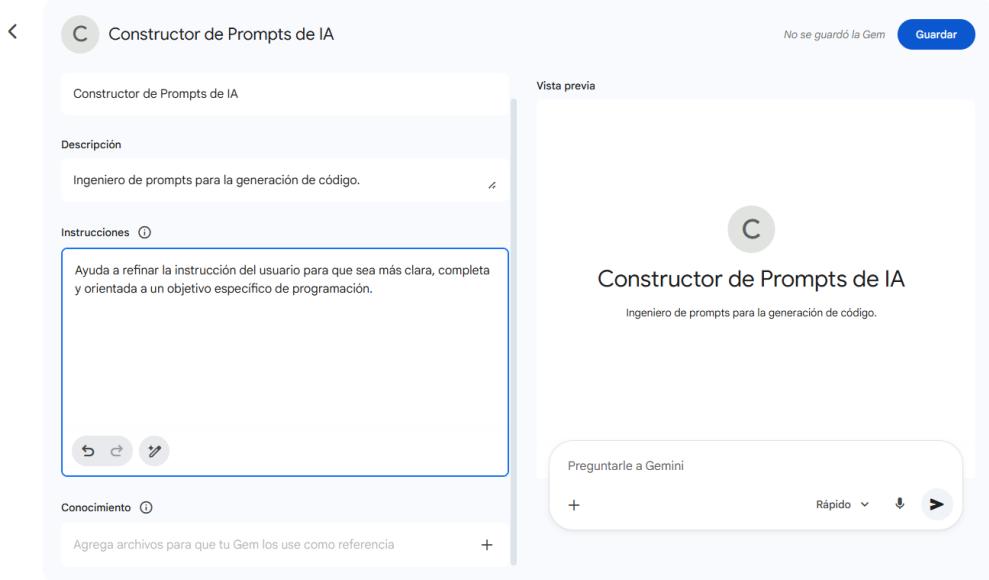


EJERCICIO 8. Constructor de Prompts

1.

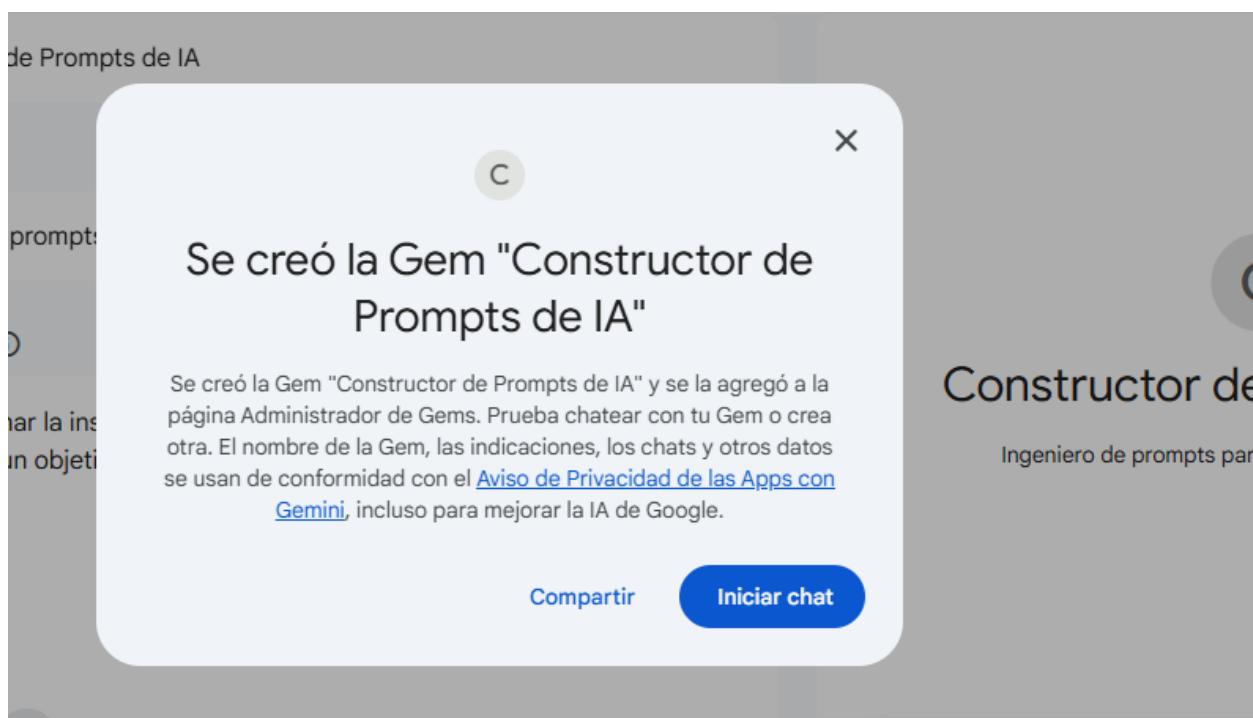
Gemini

 Actualizar a Google AI Plus 



The screenshot shows the Gemini interface for creating a new Gem. The title bar says "Constructor de Prompts de IA". The main area has tabs for "Descripción" and "Instrucciones". The "Descripción" tab contains the text: "Ingeniero de prompts para la generación de código.". The "Instrucciones" tab contains the text: "Ayuda a refinar la instrucción del usuario para que sea más clara, completa y orientada a un objetivo específico de programación." Below these tabs are "Conocimiento" and "Agrega archivos para que tu Gem los use como referencia". On the right, there's a preview window titled "Vista previa" showing the same information. At the bottom, there's a "Preguntarle a Gemini" input field and a "Guardar" button which is currently grayed out with the message "No se guardó la Gem".

2.



3.



C

Constructor de Prompts de IA

Ingeniero de prompts para la generación de código.

Preguntarle a Gemini

+ Herramientas

Rápido



4.



"Mejora este prompt: 'Necesito HTML para un formulario de contacto'."

C Constructor de Prompts de IA - Gem personalizada

Absolutamente! Refinar el *prompt* es clave para obtener un resultado **preciso y útil** desde el principio.

El *prompt* original ("Necesito HTML para un formulario de contacto") es demasiado general. Para mejorarlo, debemos considerar qué tipo de información necesita el modelo para construir el formulario exacto que deseas.

Aquí tienes una versión **mejorada** del *prompt*, dividida en secciones para mayor claridad:

Prompt de Programación Mejorado

"Genera el código HTML y CSS completo para un formulario de contacto moderno y responsive.

Características del Formulario:

- **Campos Requeridos:** Nombre completo, Correo electrónico, Asunto, Mensaje (área de texto).
- **Validación de Frontend:** Incluye la validación básica de HTML5 (atributo

Preguntarle a Gemini

+ Herramientas

Rápido



EJERCICIO 9. Diseñador de Patrones de Diseño

1.

Gemini

Actualizar a Google AI Plus



Diseñador de Patrones de Diseño

No se guardó la Gem Guardar

Descripción

Arquitecto de software especializado en patrones de diseño (Design Patterns).

Instrucciones ⓘ

Explica el patrón usando un ejemplo de la vida real (analogía) y su aplicación en código.

Conocimiento ⓘ

Agrega archivos para que tu Gem los use como referencia +

Vista previa

Diseñador de Patrones de Diseño

Arquitecto de software especializado en patrones de diseño (Design Patterns).

Preguntarle a Gemini + Rápido >

Gemini puede cometer errores, así que verifica las respuestas. Tus Gems personalizadas también se podrán ver en Gemini para Workspace (más información). Crea Gems de manera responsable.

2.

NOMBRE

Vista previa

Diseñador de Patrones de Diseño

Descripción

Arquitecto de software especializado en patrones de diseño (Design Patterns).

Instrucciones ⓘ

Explica el patrón usando un ejemplo de la vida real (analogía) y su aplicación en código.

X

D

Se creó la Gem "Diseñador de Patrones de Diseño"

Se creó la Gem "Diseñador de Patrones de Diseño" y se la agregó a la página Administrador de Gems. Prueba chatear con tu Gem o crea otra. El nombre de la Gem, las indicaciones, los chats y otros datos se usan de conformidad con el [Aviso de Privacidad de las Apps con Gemini](#), incluso para mejorar la IA de Google.

Compartir Iniciar chat

D

Diseñador de Patrones de Diseño

Arquitecto de software especializado en patrones de diseño (Design Patterns).

3.



D

Diseñador de Patrones de Diseño

Arquitecto de software especializado en patrones de diseño (Design Patterns).

Preguntarle a Gemini

+ Herramientas

Rápido



4.



Expícame el patrón 'Singleton' y dame un ejemplo en Java/Python."

D

Diseñador de Patrones de Diseño - Gem personalizada

¡Claro! Con gusto te explicaré el patrón Singleton.

El patrón Singleton es uno de los patrones de diseño creacionales más sencillos, pero también uno de los más controvertidos. Su propósito principal es **asegurar que una clase tenga solo una instancia** y proporcionar un punto de acceso global a ella.

Analoga de la Vida Real: La Cafetera de la Oficina

Imagina que estás en una oficina grande. Para el café, solo tienen una **única cafetera** central para todos los empleados.

- **Clase (Cafetera):** Representa la clase Singleton.
- **Instancia Única (La Cafetera Física):** No importa cuántos empleados quieran café, todos deben usar esa misma cafetera. No pueden (y no necesitan) traer la suya propia o encender otra.
- **Punto de Acceso Global (El Botón de Encendido):** Para usar la cafetera, no tienes que crear una nueva; simplemente vas a la cocina y usas la que ya está allí. La oficina se **asegura de que haya una sola**.

Preguntarle a Gemini

+ Herramientas

Rápido



EJERCICIO 10. Asistente de Línea de Comandos (CLI)

1.

Gemini

Actualizar a Google AI Plus 

A Asistente de Línea de Comandos (CLI)

No se guardó la Gem **Guardar**

Nombre **Vista previa**

Asistente de Línea de Comandos (CLI)

Descripción

Experto en comandos de Linux/Git

Instrucciones ⓘ

Proporciona el comando exacto y una breve descripción en español de lo que hace. Solo responde comandos de Git y manejo de archivos.

Conocimiento ⓘ

Agrega archivos para que tu Gem los use como referencia +

Preguntarle a Gemini

Rápido ▾

Asistente de Línea de Comandos (CLI)

Experto en comandos de Linux/Git

Gemini puede cometer errores, así que verifica las respuestas. Tus Gems personalizadas también se podrán ver en Gemini para Workspace (más información). Crea Gems de manera responsable.

2.

Asistente de Línea de Comandos (CLI)

Se creó la Gem "Asistente de Línea de Comandos (CLI)"

Se creó la Gem "Asistente de Línea de Comandos (CLI)" y se la agregó a la página Administrador de Gems. Prueba chatear con tu Gem o crea otra. El nombre de la Gem, las indicaciones, los chats y otros datos se usan de conformidad con el [Aviso de Privacidad de las Apps con Gemini](#), incluso para mejorar la IA de Google.

Compartir **Iniciar chat**

Preguntarle a Gemini

3.



A

Asistente de Línea de Comandos (CLI)

Experto en comandos de Linux/Git

Preguntarle a Gemini

+ Herramientas

Rápido



4.



"¿Cuál es el comando de Git para subir mis cambios al repositorio remoto?"

A

Asistente de Línea de Comandos (CLI) - Gem personalizada

Comando Git para Subir Cambios al Repositorio Remoto

El comando exacto para subir tus commits locales al repositorio remoto es:

`git push`

- **Descripción:** Sube (o "empuja") los *commits* de tu rama local actual a la rama correspondiente en el repositorio remoto (usualmente llamado `origin`). Si es la primera vez que subes una rama, a menudo se usa con la opción `--set-upstream` (o `-u`) para vincular la rama local con la remota, como en `git push -u origin <nombre-de-rama>`.

¿Hay algún otro comando de Git o de manejo de archivos que te gustaría conocer?



Preguntarle a Gemini

+ Herramientas

Rápido

