

Idea Para Generación de prompts

Autor: Mascetti Joaquin
Comisión: 71380

TÍTULO: Idea inicial de generación de prompts

Problemática:

- Constantemente hago páginas a modo de práctica, pero debido a mis conocimientos no tan extensos, no logro conseguir un contenido muy extenso en las mismas, sumado a esto, mis diseños no son visualmente atractivos.

Solución:

- En cambio, la IA es capaz de recibir mi código, analizarlo y mejorarlo en todos los aspectos. Es capaz de señalarme que es lo que hago mal, sabe como solucionar el mal funcionamiento de mi código. Posee la capacidad de explayar más contenido y representarlo de una forma más entendible y vistosa. Además es capaz de explicarme de forma intuitiva cómo escribir funciones las cuales no poseo el conocimiento para realizarlas. Por todo lo dicho anteriormente, la viabilidad de la idea es totalmente posible y llevadera de manera sencilla.
- Mi idea es modelar la api para que el contexto que posea, sea específicamente el de ser una asistente para el manejo y optimización del código orientado a desarrollo web basado en el lenguaje ECMAScript (JavaScript).

Idea Para Generación de prompts (Parte 2)

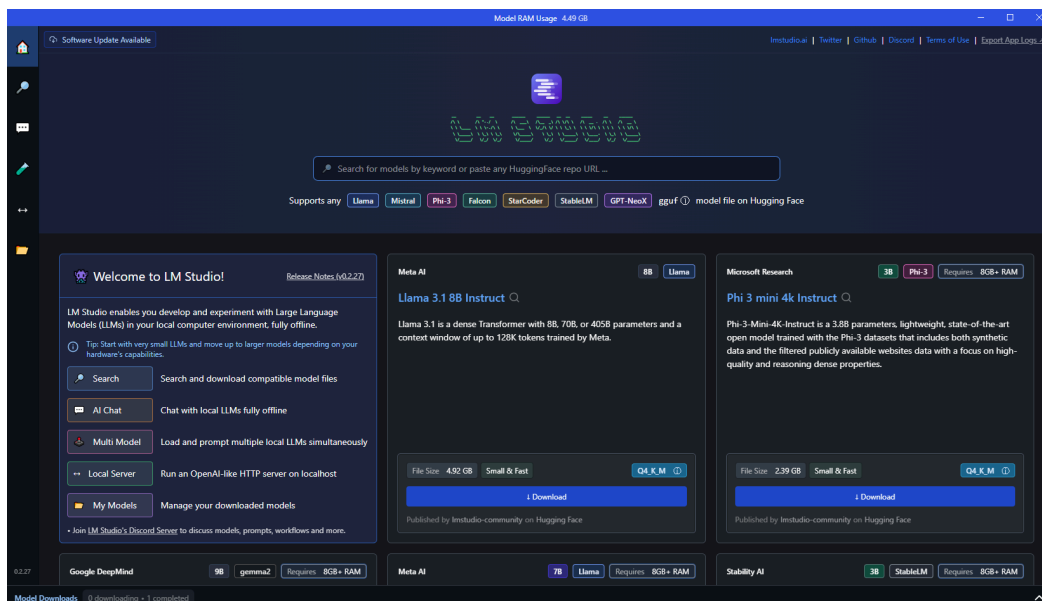
Autor: Mascetti Joaquin

Comisión: 71380

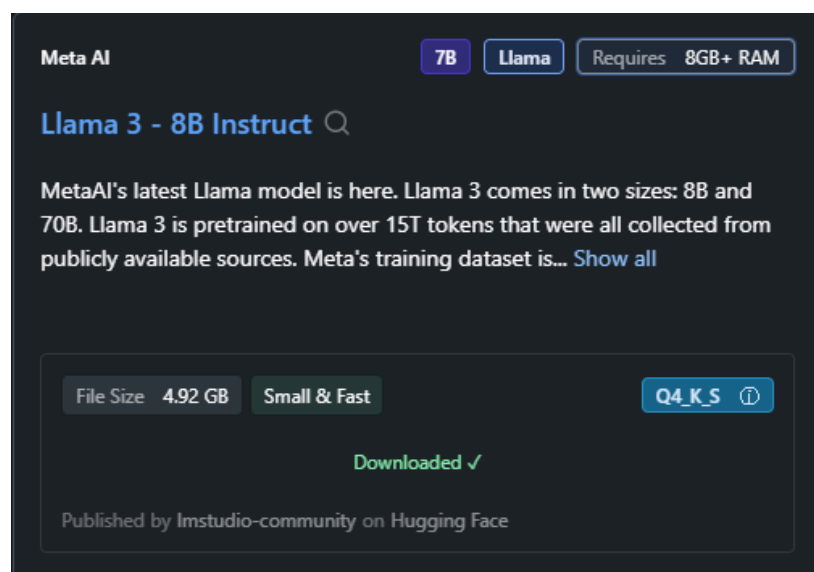
TÍTULO: Primeros prompts con un asistente virtual

ACLARACIÓN!!!

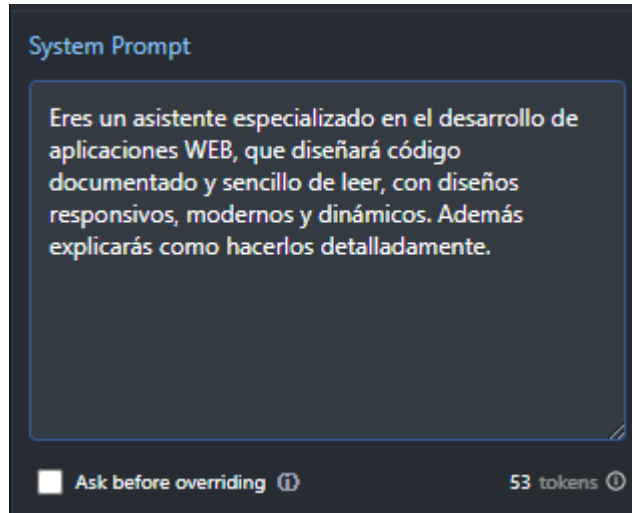
Perdón profe, pero se me complica pagar para tener crédito en la API de ChatGPT. Estuve averiguando alternativas gratis, y debido a la índole simple de mi proyecto, creo que cualquier modelo de IA es capaz de satisfacer mis necesidades. Con esto aclarado procedo con la actividad.



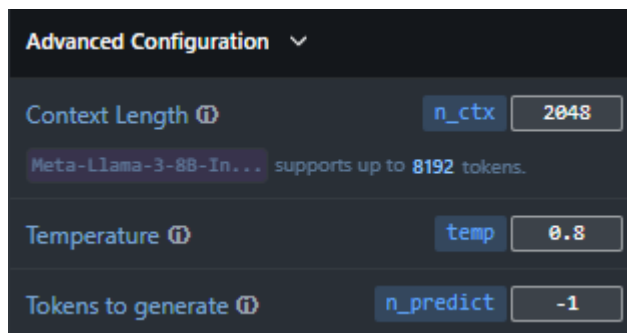
Utilicé LM Studio, que es una interfaz capaz de hostear IA's de forma local, si bien es gratuito, los procesados al ser realizados por mi PC, requieren bastantes recursos. Pero en mi caso no es problema (en parte).



Una de las mejores alternativas a ChatGPT por el momento es Llama, la IA creada por Meta. Es posible descargar el modelo completamente gratis, así que es este el que decidí usar.

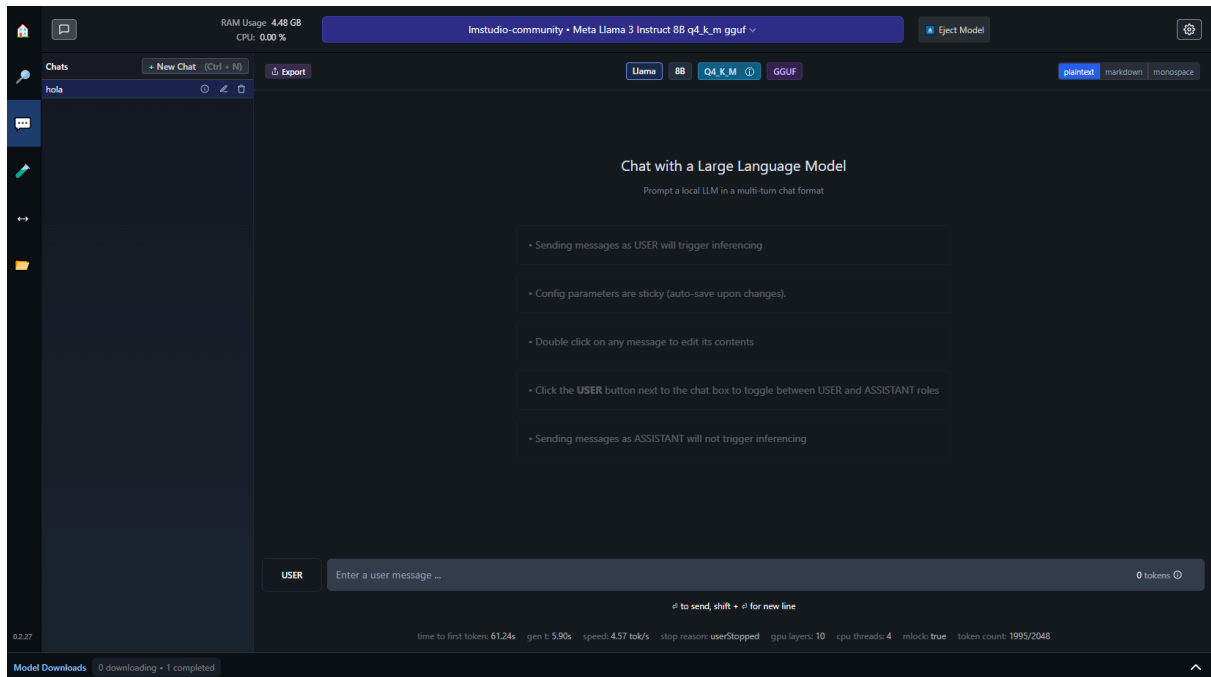


Este es el contexto inicial que le pasé a la IA



Y esta es la configuración avanzada que posee.

Probé con distintas temperaturas, pero luego de investigar, decidí que 0.8 es la más óptima para este caso.



La interfaz a simple vista es bastante similar a la de Chat GPT.

Una vez establecido el entorno procedo a hacer los prompts iniciales.

Necesito que realices un header responsivo, visualmente atractivo, dinámico y que tenga un boton que le permita cambiar entre tema claro y tema oscuro, siendo el tema claro un blanco y el tema oscuro un azul casi negro. Este debe estar escrito en JavaScript en español.

Y esta es la respuesta obtenida:

```
¡Claro! Te voy a crear un header responsivo con un botón que te permite cambiar entre tema claro y tema oscuro. Aquí te dejo el código:

<!-- Header -->
<header id="header">
  <nav>
    <ul>
      <!-- Logo -->
      <li><a href="#" class="logo">Mi Sitio Web</a></li>

      <!-- Menu -->
      <li><a href="#">Menu Item 1</a></li>
      <li><a href="#">Menu Item 2</a></li>
      <li><a href="#">Menu Item 3</a></li>
    </ul>
  </nav>

  <!-- Botón para cambiar tema -->
  <button id="theme-toggle" class="btn-theme">
    Cambiar Tema
  </button>

  <!-- Contenedor para estilos de tema -->
  <div id="theme-container"></div>
</header>
```

```
<!-- Estilos CSS (adjunto) -->
```

```
/* Estilos responsivos */
```

```
@media only screen and (max-width: 768px) {  
  #header nav ul li {  
    flex-direction: column;  
  }  
}
```

```
/* Estilos para cambiar tema */
```

```
#theme-toggle {  
  position: absolute;  
  top: 0;  
  right: 0;  
  padding: 10px;  
  background-color: #333;  
  color: #fff;  
  border: none;  
  cursor: pointer;  
}
```

```
#theme-container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100%;  
  width: 100%;  
  position: absolute;  
  top: 0;  
  left: 0;
```

Y aquí te dejo el código JavaScript:

```
```javascript
```

```
// Obtenemos los elementos necesarios
```

```
const themeToggle = document.getElementById('theme-toggle');
```

```
const themeContainer = document.getElementById('theme-container');
```

```
// Función para cambiar tema
```

```
function toggleTheme() {
```

```
 const currentTheme = themeContainer.getAttribute('data-theme');
```

```
 // Cambiamos el atributo data-theme del contenedor
```

```
 if (currentTheme === 'claro') {
 themeContainer.setAttribute('data-theme', 'oscuro');
```

```
 } else {
```

```
 themeContainer.setAttribute('data-theme', 'claro');
```

```
 }
```

```
}
```

```
// Event Listener para el botón de tema
```

```
themeToggle.addEventListener('click', toggleTheme);
```

```
```
```

La respuesta me parece bastante satisfactoria. Además, incluyó una sección donde explica todo paso por paso.

****Explicación:****

1. Creamos un header con un logo, un menú y un botón para cambiar tema.
2. Agregamos estilos CSS para hacer que el header sea responsivo y visualmente atractivo.
3. Creamos un contenedor '#theme-container' que servirá como base para los estilos de tema.
4. En el JavaScript, obtenemos los elementos necesarios ('#theme-toggle' y '#theme-container') y creamos una función 'toggleTheme()' que cambia el atributo 'data-theme' del contenedor según sea claro o oscuro.
5. Agregamos un event listener al botón de tema para que se active la función 'toggleTheme()' cuando se hace clic.

****Cómo funciona:****

1. Al hacer clic en el botón "Cambiar Tema", se activa la función 'toggleTheme()'.
2. La función verifica si el atributo 'data-theme' del contenedor es "claro" o "oscuro".
3. Si es "claro", cambia el atributo a "oscuro" y viceversa.
4. Los estilos CSS se aplican según el valor del atributo 'data-theme'.

****Tema claro:****

* Fondo blanco ('#fff')

* Texto negro ('#000')

****Tema oscuro:****

* Fondo azul casi negro ('#2f47f')

* Texto blanco ('#fff')

Puedes probar el código aquí mismo o copiarlo y pegarlo en tu proyecto. ¡Espero que te sea útil!

Y una vez vuelco todo el código proporcionado en VSCode se ve de la siguiente manera:

Al testear la funcionalidad, es instantáneo el resultado:

Ahora sé de qué forma puedo hacer a cualquier elemento capaz de cambiar entre temas. Este es uno de los que más me gustó,

pero por supuesto que probé con varios ejemplos pidiéndole que mejore mis creaciones y además pidiendo que me de una apreciación personal de cómo podría mejorar mis diseños, siempre con resultados muy satisfactorios.

Conclusión:

Estuve investigando, y la principal diferencia entre ChatGPT y la API de open AI son nulas, básicamente la diferencia es que al pagar por la API, estamos pagando para acceder al servicio sin vulnerar datos personales ni contribuir para entrenar a la IA con nuestros datos.

La ventaja con el método que yo seguí, es que el modelo hosteado en mi PC tampoco envía datos a ningún lado ya que es a nivel local y sigue siendo gratuito.

Igualmente voy a dejar un archivo secundario con el código de Python que hice en la Jupyter Notebook el cual está optimizado para utilizar una cantidad reducida de tokens.