

# Documentación Aplicativo Web CHATBOT Universidad de Caldas

El chatbot se desarrollaría para mejorar la experiencia de los estudiantes, profesores y personal administrativo de la Universidad de Caldas, brindando un sistema de asistencia y consulta que pueda responder preguntas frecuentes, ofrecer información relevante sobre la institución, servicios, eventos, programas académicos, horarios, entre otros.

## Funcionalidades Principales:

- Interacción Natural: Capacidad para entender y responder preguntas escritas de manera natural.
- Consulta de Información: Proporcionar datos sobre programas académicos, horarios de clases, eventos, ubicaciones, etc.
- Asistencia: Responder preguntas frecuentes de estudiantes y profesores.
- Direcciones y Mapas: Proporcionar direcciones y mapas de ubicaciones dentro del campus universitario.
- Actualizaciones y Noticias: Mantener informados a los usuarios sobre noticias, eventos importantes y cambios en políticas o normativas.
- Feedback y Mejora Continua: Recopilar feedback de los usuarios para mejorar las respuestas y la experiencia general del chatbot.

## Características Principales:

- Integración de la API de ChatGPT: Utilizar la API de ChatGPT para permitir respuestas coherentes y contextualmente relevantes.
- Diseño Responsivo: Interfaz adaptable para su uso en diferentes dispositivos (computadoras, tablets, móviles).
- Seguridad: Garantizar la seguridad de la información sensible y la privacidad de los usuarios.
- Escalabilidad: Capacidad para manejar múltiples usuarios simultáneamente sin comprometer su rendimiento.
- Beneficios Potenciales:
- Acceso Rápido a la Información: Reducción del tiempo dedicado a buscar información en múltiples fuentes.
- Disponibilidad 24/7: Acceso a respuestas instantáneas en cualquier momento, mejorando la experiencia del usuario.

## Requisitos Funcionales:

1. Comprensión de Lenguaje Natural: Capacidad para entender y responder preguntas en lenguaje natural.
2. Identificación de Usuarios: Crear un sistema de autenticación para identificar a los usuarios.
3. Historial del Chat: Almacenar conversaciones de usuarios y respuestas.
4. Acceso a Funciones Premium: Interfaz para acceder a funciones especiales para usuarios Premium.
5. Resolución Flexible de Consultas: Responder preguntas en cualquier orden y tolerar errores ortográficos.

6. **Búsqueda de Información Pública:** Acceso a información pública de la Universidad de Caldas (fechas, ubicaciones, documentos no sensibles).
7. **Ubicación y Direcciones en la Universidad:** Capacidad para ubicar a los estudiantes en el campus y ofrecer indicaciones para llegar a lugares específicos.
8. **Búsqueda de Documentos Académicos:** Ayuda en la búsqueda de documentos como informes de trabajo de grado.
9. **Acceso a Documentos Oficiales para Usuarios Premium:** Proporcionar documentos oficiales como reglamento estudiantil a usuarios Premium.
10. **Búsqueda de Aulas de Clase para Usuarios Premium:** Capacidad para localizar aulas para usuarios Premium.
11. **Manejo de Respuestas no Encontradas:** Proporcionar respuestas claras cuando no se tenga una respuesta o buscar una adecuada.
12. **Métodos de Pago Implementados:** Permitir transacciones según el lenguaje utilizado.
13. **Generación de Reportes de Uso:** Crear informes sobre estadísticas de uso de la herramienta.

### **Requisitos No Funcionales:**

1. **Seguridad:** Garantizar la protección de datos de usuarios.
2. **Rendimiento:** Mantener un rendimiento óptimo incluso con múltiples usuarios simultáneos.
3. **Escalabilidad:** Capacidad para crecer y manejar un mayor volumen de usuarios.
4. **Usabilidad:** Interfaz fácil de usar para usuarios de todos los niveles.
5. **Disponibilidad:** Disponibilidad del servicio 24/7 para usuarios.
6. **Precisión:** Proporcionar respuestas precisas y relevantes.
7. **Adaptabilidad Lingüística:** Ser capaz de manejar múltiples idiomas para responder a usuarios internacionales.

### **Propósito:**

**Guía de Desarrollo:** Detallar los pasos necesarios para la creación del chatbot, desde la comprensión de requisitos hasta la implementación técnica.

**Referencia para Implementación:** Servir como referencia para desarrolladores, equipos técnicos y de diseño.

**Manual de Usuario:** Proporcionar instrucciones claras sobre cómo utilizar el chatbot y acceder a funciones especiales.

**Mantenimiento y Actualización:** Ofrecer pautas para el mantenimiento continuo, actualización y mejora del sistema.

### **Alcance:**

**Desarrollo Técnico:** Incluye detalles sobre la integración con la API de ChatGPT, la configuración de autenticación de usuarios, el almacenamiento de historial de chats y la implementación de funciones Premium.

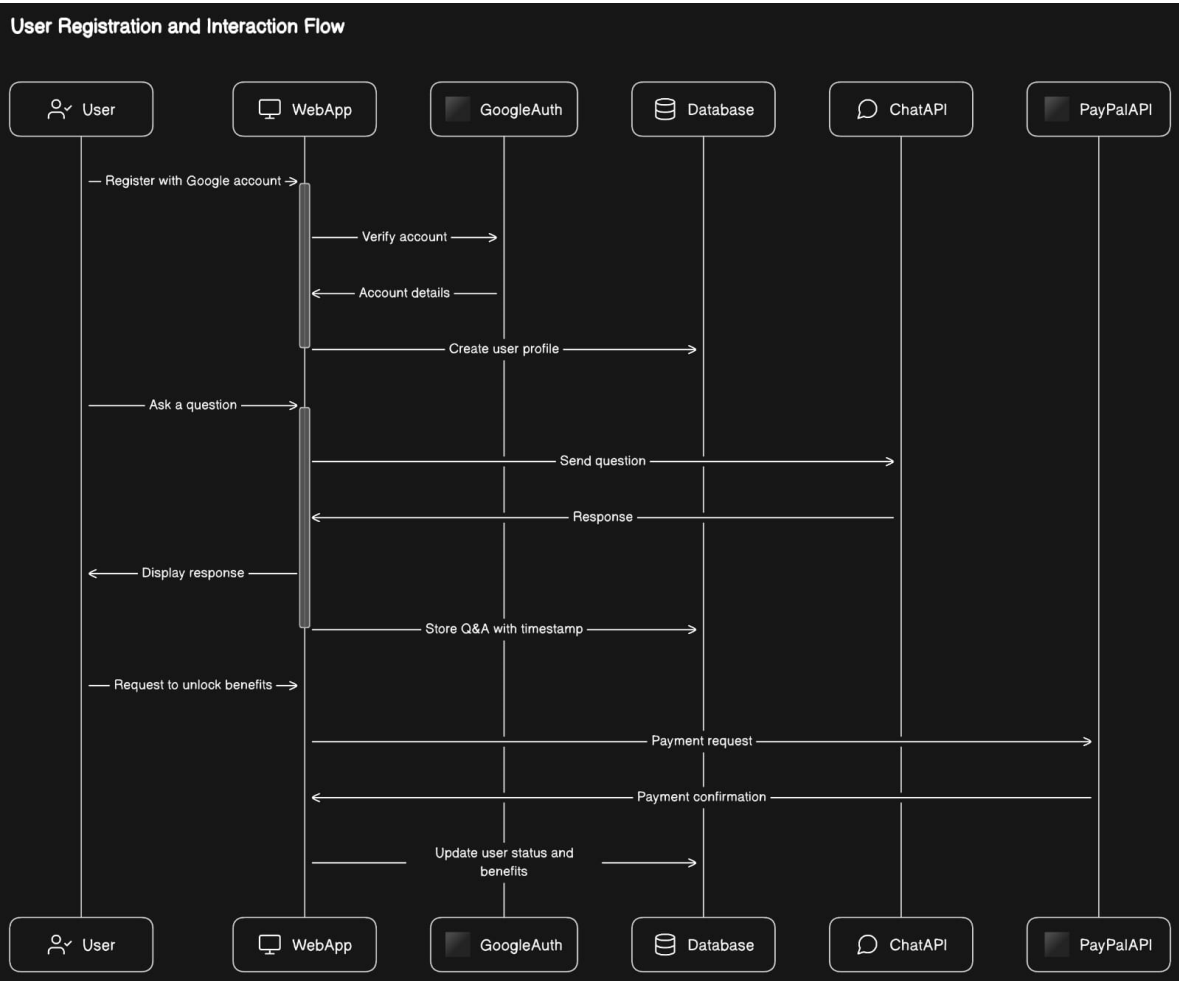
**Interfaz de Usuario:** Describe la interfaz web necesaria para la autenticación, acceso a funciones Premium, visualización de historial de chats y búsqueda de información.

**Límites de la Información:** Se centra en acceder a información pública disponible en la página de la Universidad de Caldas, excluyendo datos sensibles.

**Funciones de Ayuda al Usuario:** Detalla cómo el chatbot ayudará a los usuarios a encontrar ubicaciones en el campus, documentos académicos, fechas relevantes y proporcionará respuestas claras incluso cuando no tenga información específica.

**Integración de Métodos de Pago:** Si bien no se detalla en profundidad en la descripción, la documentación podría abordar cómo integrar métodos de pago en función de los requisitos del sistema.

Flujo de Actividades / Diagrama de Secuencia



## API y Datos

### 1. Acceso a la API:

**Registro y Autenticación:** Primero, se necesita obtener acceso a la API. Esto podría requerir un registro en la plataforma que provea la API de ChatGPT y la obtención de credenciales de autenticación (tokens, claves API). Teniendo una cuenta Premium se podrá usar libremente la API sin problemas.

### 2. Integración en el Aplicativo Web:

**Llamadas a la API:** Una vez se tiene acceso, se hacen los llamados a la API desde el aplicativo web. Estas llamadas se hacen, enviando solicitudes HTTP a los endpoints proporcionados por la API.

### 3. Procesamiento de las Solicitudes y Respuestas:

**Envío de Consultas:** El aplicativo web envía las consultas de los usuarios al servidor de la API de ChatGPT. Estas consultas son preguntas en lenguaje natural.

**Procesamiento en la API:** La API procesará estas consultas utilizando el modelo de lenguaje entrenado y devolverá respuestas generadas de manera contextual y coherente.

### 4. Manejo de Respuestas:

**Visualización de Respuestas:** Las respuestas generadas por la API se recibirán en el aplicativo web y se mostrarán al usuario a través de la interfaz de chat.

**Gestión de Contexto:** En caso de mantener conversaciones continuas, el aplicativo debe manejar el contexto de la conversación para mantener la coherencia en las respuestas.

### 5. Implementación Adicional (Funciones Premium, Búsqueda de Documentos):

**Lógica Adicional:** Para funciones Premium o búsquedas específicas, el aplicativo cuenta con una función para poder usar el chat donde debe ser Premium esta tiene su propia lógica de negocio y manejo de datos, interactuando con la API de ChatGPT y el método de pago utilizado en este caso PAYPAL.

Base de Datos

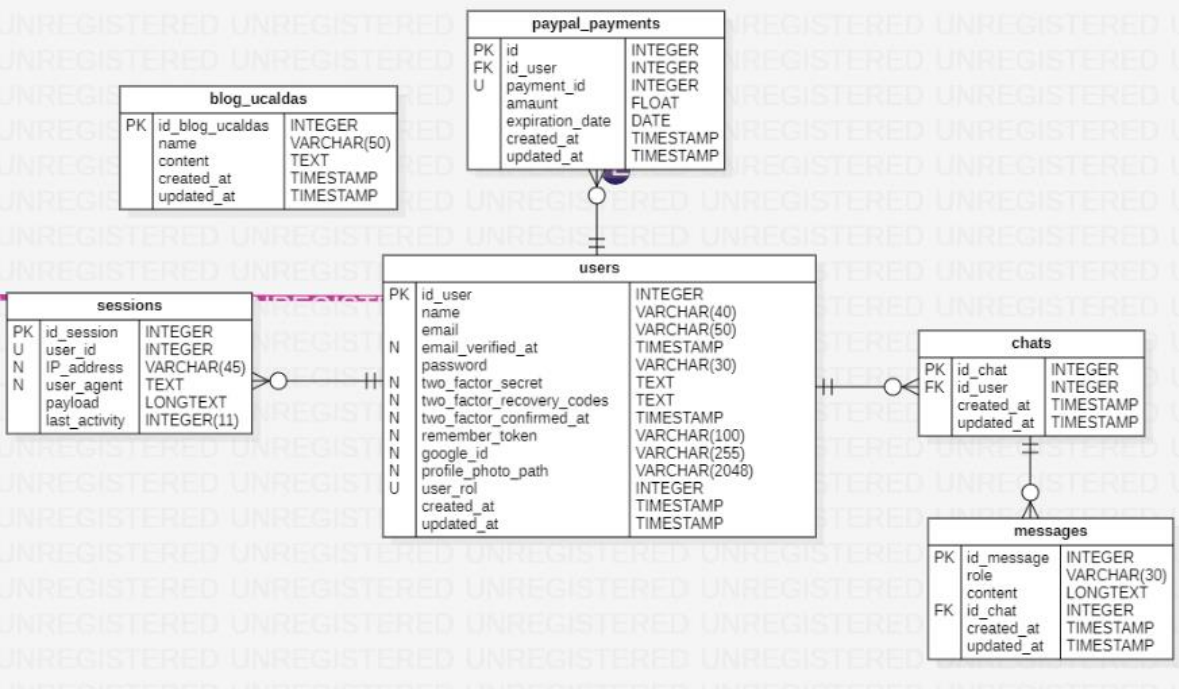
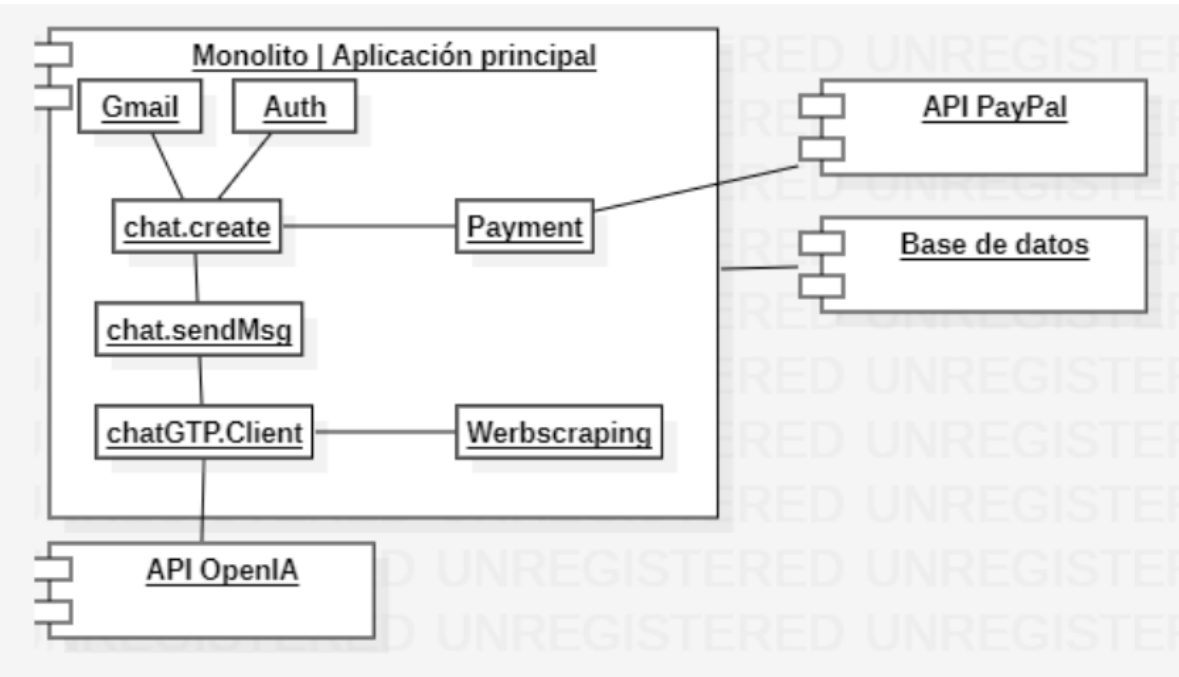
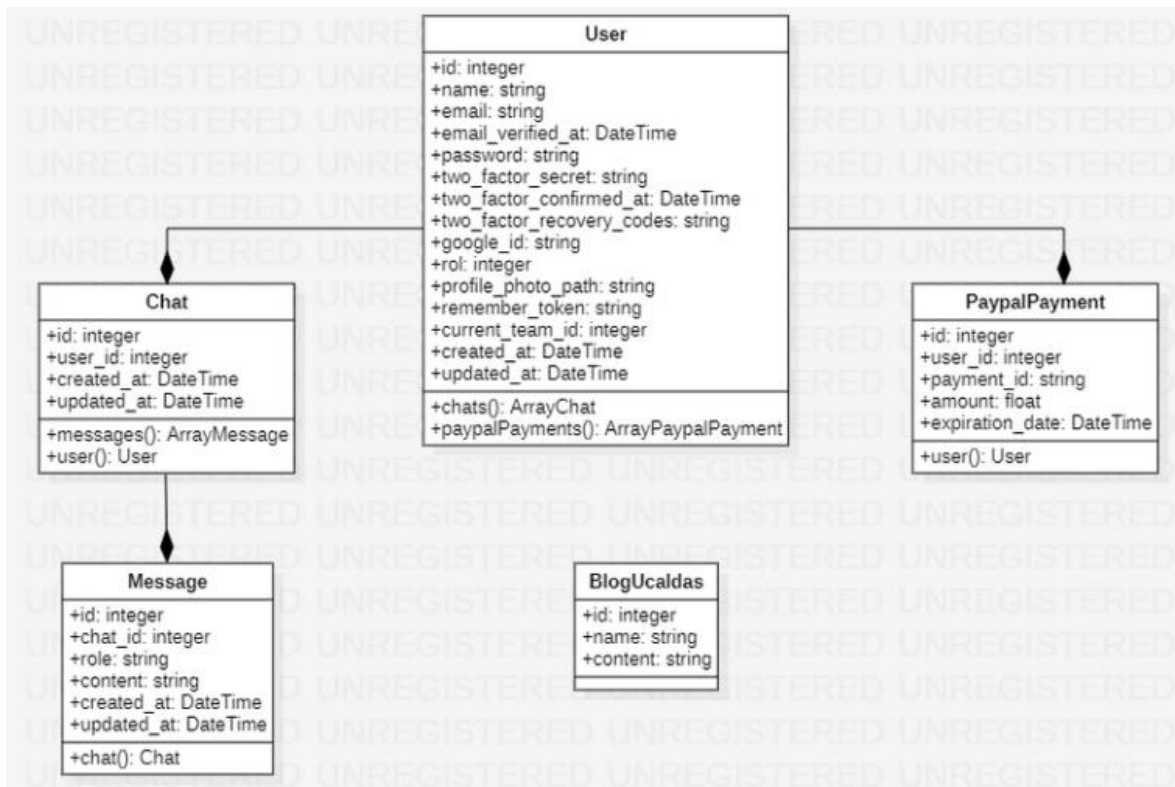


Diagrama de Componentes

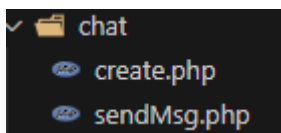


## Diagrama de Clases



## Lógica del Aplicativo

### Controladores



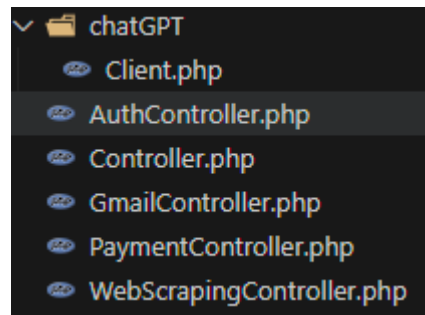
En primera instancia se tiene la carpeta chat con 2 controladores en donde chat contiene la lógica del CRUD del chat y en msg donde se envía.

```

12 class create extends Controller
13 {
14
15     public function create(Request $request) {
16
17         if (Auth::check()) {
18             $chat = new Chat();
19             $chat->user_id = Auth::id();
20             $chat->save();
21
22             return response()->json([
23                 "success" => true,
24                 "chat" => $chat,
25             ]);
26         } else {
27             return response()->json([
28                 "success" => false,
29                 "message" => "User is not authenticated.",
30             ], 401); // Código de estado 401 indica no autorizado
31         }
32     }
33
34     public function listar(){ "listar": Unknown word.
35         //$chats = chat::all();
36         //dd($chats);
37         $chats = Auth::user()->chats;
38         return response()->json([
39             "success" => true,
40             "chats" => $chats,
41         ]);

```

Luego se tiene otra carpeta donde se tiene ChatGPT que es donde se consume el servicio y están listado descriptivamente los controladores con que es lo que hace cada uno.



En cada uno de ellos tenemos con comentarios que identifican que hace cada tipo de función y método designado.

```
if ($result->getState() === 'approved' ) {
    $status = 'Gracias! El pago a través de PayPal se ha realizado correctamente.';    "Gracias": Unknown word.

    // Obtener información relevante del pago    "Obtener": Unknown word.
    $paymentId = $result->getId();
    $payerId = $result->getPayer()->getPayerInfo()->getPayerId();
    $amount = $result->getTransactions()[0]->getAmount()->getTotal();

    // Obtener al usuario que realizó el pago    "Obtener": Unknown word.
    $user = auth()->user();
    $user->rol = 2;
    $user->save();

    // Obtener la fecha de vencimiento (por ejemplo, 10 días después de hoy)    "Obtener": Unknown word.
    $expirationDate = now()->addDays(5);

    // Crear un registro en la tabla de pagos    "Crear": Unknown word.
    $paymentRecord = new PaypalPayment([
        'user_id' => auth()->user()->id, // o la lógica que uses para obtener el ID del usuario    "Obtener": Unknown word.
        'payment_id' => $paymentId,
        'amount' => $amount,
        'expiration_date' => $expirationDate,
    ]);
    $paymentRecord->save();

    return redirect('/dashboard')->with(compact('status'));
}

$status = 'Lo sentimos! El pago a través de PayPal no se pudo realizar...';    "sentimos": Unknown word.
return redirect('/dashboard')->with(compact('status'));
}

// Validar Plan de cada usuario    "Validar": Unknown word.
public function pagos(Request $request) {    "pagos": Unknown word.

    if (Auth::check()) {
        $user = auth()->user();
        $pagos = $user->paypalPayments()->get();    "pagos": Unknown word.
        return response()->json([
            "success" => true,
            "pagos" => $pagos,    "pagos": Unknown word.
        ]);
    } else {
```