

# apibot-Bitrix

---

## 1. Introducción del proyecto

Integración del motor conversacional Ánima Bot con Bitrix24, utilizando una arquitectura backend puro en Laravel. Permite recibir mensajes desde Bitrix (Telegram, WhatsApp, etc.) y procesarlos mediante árboles conversacionales definidos en Ánima Bot, en el archivo .env example se encuentra un ejemplo de como se debe configurar el archivo .env para producción, importante llamar las variables como se muestra en el archivo de ejemplo, la url del portal que se usa para diferentes configuraciones es la url donde se creo la cuenta de bitrix, mas adelante se muestra el ejemplo.

## 2. Requisitos previos

- Laravel 10 o superior
- PHP 8.2+
- Composer
- MySQL/MariaDB
- ngrok para exponer localmente (ojo: únicamente si el proyecto se ejecuta de manera local).
- Cuenta de administrador en Bitrix24.

## 3. Instalación de dependencias

- Clonar el repositorio
- Ejecutar ``composer install``
- Crear y configurar el archivo ``env``
- Ejecutar ``php artisan migrate``

## 4. Registro de la App en Bitrix

A continuación se muestra la configuración usada para registrar la Aplicación Local y el Webhook de salida, las imágenes que se muestran después de estas instrucciones del paso 4 son tomadas de la configuración con las que se estuvo realizando las pruebas en el local y cuenta de pruebas de bitrix.

Configuración de Aplicación Local Bitrix, en todos los casos donde se ve la url configurada de ngrok se debe poner la url del proyecto nuevo, muy importante configurar los permisos adecuados en la aplicación local y configurar los eventos indicados en el Webhook de salida, para que la aplicación funcione correctamente.

## Configuración de la aplicación local

Aplicación local

☒ Servidor ☐ Estático

Ruta de su controlador\*

https://2e60-189-129-90-194.ngrok-free.app/api/v1.0.0/bitrix/oauth/callback

Ruta de instalación inicial

https://2e60-189-129-90-194.ngrok-free.app/api/v1.0.0/bitrix/oauth/authorize

ID de la aplicación (client\_id)

local.683cdf210db240.90675736

Clave de la aplicación (client\_secret)

ifWuDhKJ5z1boui30T4XZFLhKW56X9kJkg2qUhju4n4McDYzLa

☒ Solo script (sin interfaz de usuario)

REINSTALAR

Asignar permisos

Chat y Notificaciones (im) x

Creación y administración de los chatbots (imbot) x


Canales Abiertos (imopenlines) x

Usuarios (básico) (user\_basic) x + seleccionar

GUARDAR

CERRAR

## Configuración del Webhook de salida

Webhook de salida 

Cree un webhook de salida para obtener información sobre los eventos que ocurren en su Bitrix24.

Webhook de salida

URL de su controlador\*

https://2e60-189-129-90-194.ngrok-free.app/api/v1.0.0/webhook/bitrix/message

Token de la aplicación

siju3mrhnlw014v42a69z671g6h5hawo

Eventos

New chat message (ONIMBOTMESSAGEADD) x

New message from Open Channel (ONIMCONNECTORMESSAGEADD) x

+ seleccionar

GUARDAR

CERRAR

## 5. Insertar credenciales desde Laravel ejecutando endpoint.

POST /api/v1.0.0/bitrix/instance/setup

Campos: portal, client\_id, client\_secret, auth\_token

Bearer Token: <auth\_token>

Body

```
{
  "portal": "b24-4bkru2.bitrix24.mx",
  "client_id": "local.68535e07c46c85.62081250",
  "client_secret": "pMEptEHsONHkLXOW4liAPWqw7j02yKhHAHjWUrPzdkW23W2Blo",
  "auth_token": "g3q77o79cslj9tczp4z6gryfmied462e"
}
```

## 6. Insertar hash del chat para la instancia desde Laravel ejecutando endpoint.

POST /api/v1.0.0/bitrix/update-hash

Campos: portal, hash

Bearer Token: <auth\_token>

Body

```
{
  "portal": "b24-4bkru2.bitrix24.mx",
  "hash": "eyJpdil6Im....."
}
```

## 7. Endpoint de instalación OAuth (se ejecutan cuando se instala o repíntala la aplicación local de Bitrix)

- Ruta de instalación inicial: /api/v1.0.0/bitrix/oauth/authorize
- Ruta del controlador: /api/v1.0.0/bitrix/oauth/callback

## 8. Flujo conversacional

Bitrix recibe mensaje → Webhook hacia Laravel → Laravel detecta portal y sesión → Llama al árbol de Ánima → Procesa respuesta → Responde a Bitrix (Telegram, WhatsApp, etc.)

## 9. Seguridad del Webhook

El middleware `verify.bitrix` protege el webhook validando:

- Bearer Token (auth\_token) enviado en la cabecera.
- application\_token en el payload, validado contra la BD.