NOVA COR ESTAMPARIA

Documentação Técnica — v2.0

WPPConnect Server

Guia de Integração

Como conectar, configurar e usar o bot de WhatsApp para envio de mensagens automáticas — do zero ao avançado.

Python 3	WPPConnect 2.9	Ubuntu 24.04	VPS 72.60.11.219

Elaborado por Codinome Studio · Março 2026

Sumário

1. O que é o WPPConnect?

O WPPConnect é um servidor open-source que transforma o WhatsApp Web em uma API REST. Ele roda no seu VPS, cria uma sessão autenticada via QR Code e expõe endpoints HTTP para enviar mensagens, arquivos, imagens e muito mais — tudo de forma programática, sem custo por mensagem.

Na Nova Cor Estamparia, ele está rodando na porta 21465 do servidor 72.60.11.219, gerenciado pelo PM2 (processo chamado orc-wpp).

> ■ *O WPPConnect NÃO é a API Oficial do WhatsApp Business. Ele funciona via automação do WhatsApp Web. Para volumes altos (acima de 500 msg/dia), considere a API Oficial da Meta.*

2. Arquitetura do Sistema

Componente	Localização	Função
WPPConnect Server	VPS :21465	API REST do WhatsApp
PM2 (orc-wpp)	VPS /opt/wppconnect	Mantém o processo vivo
WordPress Plugin	novacorestamparia.com	Envia notificações de OS
intermax_service.py ■	VPS :5000	Webhook + Gemini IA
PM2 (intermax) ■	VPS /opt/intermax	Processa imagens do grupo
raizes_cobranca.py	PC local (Windows)	Disparo em massa de cobranças

3. Passo a Passo: Primeira Conexão

1 — Conectar no VPS via SSH

```
ssh root@72.60.11.219
```

2 — Verificar se o processo está rodando

```
pm2 list

# Deve mostrar orc-wpp com status 'online'
```

3 — Acessar o Manager Web

```
# No navegador:

http://72.60.11.219:21465/manager
```

4 — Gerar o Token de Autenticação

```
curl -s -X POST \

  http://72.60.11.219:21465/api/novacor/THISISMYSECURETOKEN/generate-token
```

5 — Criar a Sessão e Obter o QR Code

```
TOKEN="SEU_TOKEN_AQUI"

curl -s -X POST http://72.60.11.219:21465/api/novacor/start-session \

  -H "Authorization: Bearer $TOKEN" \

  -H "Content-Type: application/json" \

  -d '{"webhook":"http://localhost:5000/webhook","waitForLogin":false}'
```

6 — Escanear o QR Code

```
# Configurações → Dispositivos Conectados → Conectar Dispositivo
```

7 — Verificar a Conexão

```
curl -s http://72.60.11.219:21465/api/novacor/check-connection-session \

  -H "Authorization: Bearer $TOKEN"
```

```
# Deve retornar: {"status": true}
```

■ *O token gerado tem validade. Se aparecer erro 401, gere um novo token.*

■ *IMPORTANTE: sempre inicie a sessão com webhook apontando para http://localhost:5000/webhook para que o Intermax receba as imagens do grupo.*

4. Reconectar após Queda

O WhatsApp pode desconectar por vários motivos: reinício do celular, atualização do app, ou inatividade prolongada. O processo no VPS continua rodando — só a sessão precisa ser reautenticada.

Opção A — Pelo Manager Web (mais fácil)

1. Acesse http://72.60.11.219:21465/manager
2. Clique no ícone de engrenagem na instância novacor
3. Clique em Conectar e escaneie o QR Code com o celular

Opção B — Via curl

```
# 1. Gera token novo

TOKEN=$(curl -s -X POST \

  http://72.60.11.219:21465/api/novacor/THISISMYSECURETOKEN/generate-token \

  | python3 -c "import sys,json; print(json.load(sys.stdin)['token'])")


# 2. Reinicia a sessão COM webhook

curl -s -X POST http://72.60.11.219:21465/api/novacor/start-session \

  -H "Authorization: Bearer $TOKEN" \

  -H "Content-Type: application/json" \

  -d '{"webhook":"http://localhost:5000/webhook","waitForLogin":false}'
```

■ *Se o processo PM2 cair (raro): pm2 restart orc-wpp*

5. Endpoints Essenciais da API

A URL base é sempre http://72.60.11.219:21465. Todos os endpoints autenticados precisam do header Authorization: Bearer SEU_TOKEN.

Endpoint	Método	Descrição
/api/{session}/{secret}/generate-token	POST	Gera token de autenticação
/api/{session}/check-connection-session	GET	Verifica se está conectado
/api/{session}/start-session	POST	Inicia sessão / gera QR Code
/api/{session}/qrcode-session	GET	Retorna o QR Code atual
/api/{session}/send-message	POST	Envia mensagem de texto
/api/{session}/send-file-base64	POST	Envia arquivo em base64 (PDF, img)
/api/{session}/send-image	POST	Envia imagem com legenda
/api/{session}/status-session	GET	Status detalhado da sessão
/api/{session}/logout-session	POST	Desconecta a sessão
/api/{session}/all-chats	GET	Lista todos os chats
/api/{session}/check-number-status/{phone}	GET	Verifica se número existe no WPP

■ *Substitua {session} por 'novacor' e {secret} por 'THISISMYSECURETOKEN'.*

6. Enviando Mensagens via Python

```
import requests


WPP_URL = "http://72.60.11.219:21465"

SESSION = "novacor"

SECRET  = "THISISMYSECURETOKEN"


# 1. Gera o token

r = requests.post(f"{WPP_URL}/api/{SESSION}/{SECRET}/generate-token")

TOKEN = r.json()["token"]


# 2. Envia a mensagem

# Formato: 55 + DDD + número + @c.us

numero = "5577988887777@c.us"

resp = requests.post(

    f"{WPP_URL}/api/{SESSION}/send-message",

    headers={"Authorization": f"Bearer {TOKEN}", "Content-Type": "application/json"},

    json={"phone": numero, "message": "Olá! *Negrito* e _itálico_ funcionam!"}

)

print(resp.json()["status"])  # "success"
```

Formatação de texto (igual WhatsApp)

Efeito	Sintaxe	Exemplo
Negrito	*texto*	*Olá mundo*
Itálico	_texto_	_Obrigado!_
Tachado	~texto~	~Cancelado~
Monoespaçado	```texto```	```código```

Enviando para Grupos ■

■ **Para enviar mensagens em grupos, sempre inclua isGroup: True no payload.**

```
resp = requests.post(
    f"{WPP_URL}/api/{SESSION}/send-message",
    headers={"Authorization": f"Bearer {TOKEN}", "Content-Type": "application/json"},
    json={
        "phone":   "120363406942821334@g.us",
        "message": "Olá grupo!",
        "isGroup": True
    }
)
```

7. Enviando Arquivos e PDFs

O WPPConnect aceita arquivos em base64. O processo é: ler o arquivo → converter para base64 → enviar via API.

Enviar PDF local

```
import requests, base64

with open('orcamento.pdf', 'rb') as f:
    b64 = base64.b64encode(f.read()).decode('utf-8')

resp = requests.post(
    f"{WPP_URL}/api/{SESSION}/send-file-base64",
    headers={"Authorization": f"Bearer {TOKEN}", "Content-Type": "application/json"},
    json={
        "phone":    "5577988887777@c.us",
        "base64":   b64,
        "filename": "orcamento.pdf",
        "caption":  "Segue seu orçamento!"
    }
)
```

■ *Envie o base64 LIMPO, sem o prefixo 'data:application/pdf;base64,'. Com o prefixo, o arquivo chega corrompido.*

Enviar imagem em grupo ■

■ **Para grupos, inclua isGroup: True. O WPPConnect retorna status 201 (não 200) no sucesso.**

```
resp = requests.post(
    f"{WPP_URL}/api/{SESSION}/send-file-base64",
    headers={"Authorization": f"Bearer {TOKEN}", "Content-Type": "application/json"},
    json={
        "phone":    "120363406942821334@g.us",
```

```
        "base64":   b64_imagem,
        "filename": "arte.png",
        "caption":  "Arte processada!",
        "isGroup":  True
    }
)
# Sucesso: status_code in (200, 201)
```

8. Webhook — Recebendo Mensagens ■

■ Esta seção é nova na v2.0 do guia.

O WPPConnect pode enviar um POST para uma URL sua sempre que chegar uma mensagem. Isso permite reagir a mensagens em tempo real — como processar imagens de um grupo automaticamente.

Configurar o webhook na sessão

O webhook é configurado no momento do start-session. Passe a URL no campo webhook:

```
curl -s -X POST http://72.60.11.219:21465/api/novacor/start-session \

  -H "Authorization: Bearer $TOKEN" \

  -H "Content-Type: application/json" \

  -d '{"webhook":"http://localhost:5000/webhook","waitForLogin":false}'
```

Configurar o webhook no arquivo de configuração

Para o webhook persistir entre reinicializações, edite o arquivo compilado do WPPConnect:

```
# Editar o arquivo compilado (não o .ts)

sed -i "s|url: null,|url: 'http://localhost:5000/webhook',|" \

  /opt/wppconnect/dist/config.js


# Confirmar a mudança

grep 'url:' /opt/wppconnect/dist/config.js | head -1


# Reiniciar

pm2 restart orc-wpp
```

■ *Sempre edite o /dist/config.js (compilado), não o /src/config.ts (fonte). Mudanças no .ts não têm efeito sem recompilar.*

Estrutura do payload recebido

```
{

  "event": "onmessage",

  "message": {

    "from":       "120363406942821334@g.us",

    "isGroupMsg": true,
```

```
      "type":       "image",
      "author":     "5577999999999@c.us",
      "notifyName": "Nome do Remetente",
      "body":       "<base64 da imagem>",
      "mimetype":   "image/jpeg"
    }
  }
```

9. Intermax — Processamento de Imagens com IA ■

■ Esta seção é nova na v2.0 do guia.

O Intermax é um serviço Python headless que roda no VPS e processa imagens enviadas em um grupo do WhatsApp usando o Gemini (Google AI). Quando um membro envia uma foto de camiseta, o Intermax extrai a arte, processa com IA e devolve a imagem tratada no grupo.

Arquitetura

```
Grupo WPP → WPPConnect (webhook) → intermax_service.py → Gemini API
                                          ↓
                                   WPPConnect (send-file-base64)
                                          ↓
                                   Grupo WPP (imagem tratada)
```

Instalação

```
mkdir -p /opt/intermax

pip install flask requests google-genai --break-system-packages


# Copiar o serviço

# scp intermax_service.py root@72.60.11.219:/opt/intermax/


# Subir com PM2

pm2 start /opt/intermax/intermax_service.py --interpreter python3 --name intermax

pm2 save
```

Painel Web

O serviço expõe um painel web em http://72.60.11.219:5000 para gerenciar prompt, modelo Gemini, chave de API e monitorar logs em tempo real.

Modelos Gemini disponíveis (Março 2026)

Modelo	Nome	Uso recomendado
gemini-3.1-flash-image-preview	Nano Banana 2	Alta eficiência, ideal para o grupo
gemini-3-pro-image-preview	Nano Banana Pro	Alta qualidade, mais lento
gemini-2.5-flash-image	Nano Banana	Versão anterior, estável

Comandos úteis

```
pm2 logs intermax --lines 30   # Ver logs em tempo real

pm2 restart intermax           # Reiniciar o serviço

pm2 stop intermax              # Parar o serviço

cat /opt/intermax/intermax_config.json  # Ver configuração atual
```

10. Ideias de Automações

■ Notificação de OS pronta

Quando uma Ordem de Serviço mudar de etapa no Leônidas, enviar WhatsApp automático para o cliente com o status atualizado. Já implementado via ld-wpp-conect.php.

■ Processamento de arte no grupo ■

Já implementado via Intermax. Membro envia foto de camiseta → IA extrai a arte → devolve imagem tratada pronta para serigrafia.

■ Cobrança agendada

Todo domingo às 9h, varrer os pedidos pendentes e disparar lembretes via cron job no VPS.

■ Envio de arte para aprovação

Ao fazer upload de uma arte no sistema, enviar a imagem diretamente para o cliente pedir aprovação — tudo via API.

■ Relatório semanal

Todo domingo, gerar um PDF com o resumo de vendas da semana e enviar para o grupo da equipe.

■ Alerta de novo pedido

Assim que um pedido entrar no sistema, notificar o responsável de produção via WhatsApp.

11. Solução de Problemas

■ Erro 401 — Unauthorized

```
O token expirou. Gere um novo:

curl -s -X POST http://72.60.11.219:21465/api/novacor/THISISMYSECURETOKEN/generate-toke
n
```

■ 'status': false no check-connection

A sessão do WhatsApp caiu. Siga a seção 4 (Reconectar após Queda).

■ Processo orc-wpp offline no PM2

```
pm2 restart orc-wpp

pm2 logs orc-wpp --lines 20  # Ver erros
```

■ PDF chegando corrompido

```
# ERRADO:

"base64": "data:application/pdf;base64,JVBERi0..."

# CERTO:

"base64": "JVBERi0..."
```

■ Grupo não recebe mensagens ■

```
# Sempre inclua isGroup: True no payload para grupos

"isGroup": True
```

■ WPPConnect retorna 400 'número não existe' para grupo ■

```
# Falso positivo para grupos. Adicione isGroup: True.

# O WPPConnect retorna 201 (não 200) para envios em grupo com sucesso.
```

■ Webhook não recebe eventos ■

```
# Verifique se o config.js foi editado (não o config.ts):

grep 'url:' /opt/wppconnect/dist/config.js | head -1
```

```
# Deve mostrar: url: 'http://localhost:5000/webhook'

# Após editar, reinicie: pm2 restart orc-wpp
```

■ Intermax não processa imagens ■

```
pm2 logs intermax --lines 30  # Ver logs do serviço

# Verifique se gemini_api_key está configurada no painel :5000
```

Referência Rápida

Dado	Valor
URL Base da API	http://72.60.11.219:21465
Manager Web	http://72.60.11.219:21465/manager
Painel Intermax ■	http://72.60.11.219:5000
Webhook URL ■	http://localhost:5000/webhook
Session Name	novacor
Secret Key	THISISMYSECURETOKEN
Processo PM2 WPP	orc-wpp
Processo PM2 Intermax ■	intermax
Pasta WPPConnect	/opt/wppconnect
Pasta Intermax ■	/opt/intermax
Porta WPPConnect	21465
Porta Intermax ■	5000
Formato número	55 + DDD + número + @c.us
Formato grupo ■	XXXXXXXXXXXXXXXXX@g.us
Exemplo número	5577988887777@c.us
Grupo Nova Cor ■	120363406942821334@g.us