# Astra Maturador instalação

Link: https://github.com/AstraOnlineWeb/astramaturador

git clone https://github.com/AstraOnlineWeb/astramaturador.git

**Ferramenta de Maturação de Números para WhatsApp com Chatbot Inteligente**

Sobre • Funcionalidades • Instalação • Configuração • Tecnologias • Licença

---

# Sobre

**Astra Maturador** é uma ferramenta desenvolvida para **maturar números de WhatsApp** de forma automatizada e inteligente. O objetivo principal é aquecer e maturar chips novos através de conversas automáticas, utilizando um sistema de chatbot avançado com integração de Inteligência Artificial.

## O que é Maturação de Números?

A maturação é o processo de "aquecer" um número novo de WhatsApp, simulando uso orgânico através de conversas e interações. Números maturados têm menor chance de serem banidos e funcionam melhor para uso comercial e marketing.

## Como o Astra Maturador Ajuda?

- **Chatbot Inteligente** - Responde mensagens automaticamente de forma natural
- **Integração com IA** - Usa OpenAI (GPT) ou Groq (Llama) para respostas humanizadas
- **Simulação de Uso Real** - Delays configuráveis, horários de funcionamento e anti-spam
- **Contexto de Conversa** - A IA mantém o histórico para conversas coerentes

# Funcionalidades

# Chatbot de Maturação

- **Respostas automáticas** - Responde mensagens 24/7 para maturar o chip
- **Regras personalizadas** - Crie gatilhos específicos com respostas customizadas
- **Tipos de correspondência** - Exato, Contém, Inicia com, ou Todas as mensagens
- **Delay configurável** - Simula tempo de digitação humano
- **Anti-spam** - Intervalo entre respostas para parecer natural

# Integração com IA

- **OpenAI (GPT-3.5/GPT-4)** - Respostas inteligentes usando ChatGPT
- **Groq (Llama 3.1)** - Alternativa rápida e gratuita
- **Histórico de conversa** - IA mantém contexto das mensagens anteriores
- **System Prompt personalizado** - Configure a personalidade do bot

# Configurações de Maturação

- **Agendamento** - Defina horários e dias para o bot funcionar
- **Filtro de contatos** - Lista de bloqueio para ignorar contatos específicos
- **Responder grupos** - Opção para ativar/desativar respostas em grupos
- **Resposta única** - Responder apenas uma vez por contato

# Interface

- **Material Design 3** - Interface moderna com Jetpack Compose
- **Tema escuro** - Suporte nativo a dark mode
- **Widget** - Ative/desative o bot direto da home screen
- **Histórico** - Visualize todas as mensagens respondidas

# Screenshots

Tela Principal Regras de Resposta Editar Regra

Histórico Configurações Configurações IA

| Tela | Descrição |
| --- | --- |

| | |
|---|---|
| **Tela Principal** | Toggle do bot, contador de mensagens e menu de navegação |
| **Regras de Resposta** | Lista de regras configuradas com opção de ativar/desativar |
| **Editar Regra** | Configuração de gatilho, tipo de correspondência e opção de IA |
| **Histórico** | Log de mensagens recebidas e respostas enviadas |
| **Configurações** | Apps suportados, delay, anti-spam e agendamento |
| **Configurações IA** | Provedor (OpenAI/Groq), API Key e prompt do sistema |

# Instalação

## Requisitos

- Android 7.0 (API 24) ou superior
- Permissão de acesso às notificações

## Download

Baixe a versão mais recente na seção Releases.

## Build do Código Fonte

1. Clone o repositório:

```
git clone https://github.com/AstraOnlineWeb/astramaturador.git
cd astramaturador
```

2. Abra o projeto no Android Studio
3. Sincronize o Gradle e compile:

```
./gradlew assembleDebug
```

4. O APK estará em `app/build/outputs/apk/debug/`

# Configuração

## Permissões Necessárias

1. **Acesso às Notificações** - Necessário para ler mensagens do WhatsApp
   - Vá em Configurações > Apps > Astra Maturador > Permissões
   - Ou acesse: Configurações > Notificações > Acesso a notificações
2. **Notificações** - Para exibir o status do serviço em segundo plano

## Configurando IA (Recomendado para Maturação)

Para maturação eficiente, recomendamos usar IA para respostas mais naturais:

### OpenAI

1. Crie uma conta em platform.openai.com
2. Gere uma API Key em API Keys
3. Cole a chave nas configurações do app

### Groq (Gratuito)

1. Crie uma conta em console.groq.com
2. Gere uma API Key
3. Cole a chave nas configurações do app

## Criando Regras de Resposta

1. Abra o app e vá em "Regras de Resposta"
2. Toque em "+" para criar uma nova regra
3. Configure:
   - **Gatilho**: Texto que ativa a resposta
   - **Resposta**: Mensagem a ser enviada

- **Tipo de correspondência**: Como o gatilho deve ser detectado
- **Delay**: Tempo de espera antes de responder (importante para maturação)

# Estrutura do Projeto

```
astramaturador/
├── app/src/main/
│   ├── java/com/meuapp/autoreply/
│   │   ├── data/
│   │   │   ├── database/      # Room Database
│   │   │   ├── model/         # Data classes
│   │   │   └── repository/    # Repositório de dados
│   │   ├── service/
│   │   │   ├── AIService.kt           # Integração com IA
│   │   │   ├── ForegroundService.kt   # Serviço em segundo plano
│   │   │   └── NotificationListener.kt # Listener de notificações
│   │   ├── ui/
│   │   │   ├── main/          # Tela principal
│   │   │   ├── rules/         # Gerenciamento de regras
│   │   │   ├── history/       # Histórico de mensagens
│   │   │   ├── settings/      # Configurações
│   │   │   ├── components/    # Componentes reutilizáveis
│   │   │   └── theme/         # Tema do app
│   │   ├── util/              # Utilitários
│   │   ├── widget/            # Widget da home screen
│   │   └── receiver/          # Boot receiver
│   └── res/                   # Recursos (layouts, strings, etc)
├── build.gradle.kts
├── settings.gradle.kts
└── README.md
```

# Tecnologias

| Tecnologia | Uso |
| --- | --- |

| | |
|---|---|
| **Kotlin** | Linguagem principal |
| **Jetpack Compose** | Interface declarativa |
| **Material Design 3** | Design system |
| **Room Database** | Persistência local |
| **DataStore** | Preferências |
| **Coroutines** | Programação assíncrona |
| **Glance** | Widget AppWidget |
| **OpenAI API** | Integração GPT |
| **Groq API** | Integração Llama |

# Contribuindo

Contribuições são bem-vindas! Sinta-se à vontade para:

1. Fazer fork do projeto
2. Criar uma branch para sua feature (`git checkout -b feature/NovaFeature`)
3. Commit suas mudanças (`git commit -m 'Adiciona NovaFeature'`)
4. Push para a branch (`git push origin feature/NovaFeature`)
5. Abrir um Pull Request

# □□ Suporte Profissional

Precisa de ajuda para melhorar, customizar ou implementar o projeto?

□□ **WhatsApp:** +55 61 9 9687-8959

□□ **Temos uma equipe especializada para:**

- □ Customizações e melhorias
- □ Implementação e deploy completo
- □ Configuração de arquitetura SaaS
- □ Integração com outras APIs
- □ Desenvolvimento de features específicas
- □ Suporte técnico dedicado
- □ Consultoria em automação WhatsApp
- □ Treinamento e documentação

---

# Aviso Legal

Este aplicativo é fornecido apenas para fins educacionais e de automação pessoal. O uso deste software deve estar em conformidade com os Termos de Serviço do WhatsApp. Os desenvolvedores não se responsabilizam pelo uso indevido do aplicativo ou por eventuais banimentos de conta.

**Use por sua conta e risco.**

# Licença

Este projeto está licenciado sob a **GNU Affero General Public License v3.0 (AGPL-3.0)**.

Isso significa que:

- Você pode usar, modificar e distribuir este software
- Se você modificar e disponibilizar o software (incluindo como serviço web), deve disponibilizar o código fonte
- Trabalhos derivados devem usar a mesma licença

Veja o arquivo LICENSE para mais detalhes.

---

Revision #3
Created 2026-11-08 14:08:02 UTC by Administrador
Updated 2026-11-08 20:19:14 UTC by Administrador