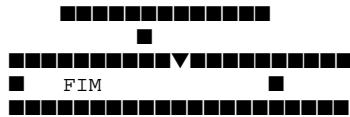




```

INÍCIO: executar_agendamento_final()
1. Conectar Notion
2. Buscar tarefas "Não iniciado"
Para cada tarefa:
3. Verificar Google Calendar
- Horário disponível?
SIM
4. Iniciar Playwright
- Browser: Chromium
- Modo: headless/headful
5. Navegar para agenda
- Adulto ou Infantil?
- URL específica
6. Selecionar especialidade
- Dropdown ou busca
7. Buscar horários disponíveis
- Varre calendário
- Identifica vagas
- Seleciona melhor horário
8. Preencher formulário
- Nome completo
- CPF
- Motivo da consulta
- Dados adicionais
9. Clicar em "Reservar"
10. DUPLA CHECAGEM
- Verifica confirmação visual
- Busca mensagem de sucesso
- Captura screenshot (opcional)
SUCESSO?
SIM NÃO
11a. Update Notion: "Concluído"
11b. Log erro Notion: status "Erro"
12. Criar evento no Google Cal.

```



## ■ FUNÇÕES PRINCIPAIS

### *Funções Core (15 total):*

1. `log_info()` - Logs com emoji e timestamp
2. `log_sucesso()` - Logs de sucesso
3. `log_erro()` - Logs de erro
4. `log_aviso()` - Logs de aviso
5. `conectar_notion()` - Inicializa cliente Notion
6. `buscar_tarefas_nao_iniciadas()` - Query no database Notion
7. `atualizar_status_tarefa()` - Update status individual
8. `atualizar_status_tarefa_completa()` - Update completo com dados
9. `navegar_para_agenda()` - Acessa página correta (Adulto/Infantil)
10. `selecionar_especialidade()` - Escolhe especialidade médica
11. `buscar_horarios_disponiveis()` - Scraping de horários livres
12. `preencher_formulario()` - Fill dos campos do paciente
13. `clicar_reservar()` - Submete o agendamento
14. `verificar_confirmacao()` - DUPLA CHECAGEM final
15. `executar_agendamento_final()` - Orquestra todo o fluxo

## ■ ESTATÍSTICAS DO CÓDIGO

### • Estruturas de Controle:

- 22 loops `for`
- 74 condicionais `if`
- 33 blocos `try/except` (robusto!)

### • Padrões de Automação:

- 10 cliques
- 7 preenchimentos de formulário
- 11 esperas/timeouts
- 1 navegação principal

## ■ SEGURANÇA & CONFIABILIDADE

### ■ *Recursos Implementados:*

1. **Logs Detalhados:** Timestamp em cada ação
2. **Tratamento de Erros:** 33 try/except
3. **Modo Dry Run:** Testes sem executar ações reais
4. **Dupla Checagem:** Verifica se agendamento foi confirmado
5. **Atualização de Status:** Notion sempre sincronizado
6. **Prevenção de Conflitos:** Verifica Google Calendar primeiro

## ■ Pontos de Atenção:

- Token Notion exposto no código (deveria usar variável de ambiente)
- Database ID hardcoded
- Sem retry automático em falhas
- Sem limite de tentativas

## ■ URLs DO SISTEMA

Agenda Adulto:  
`https://outlook.office365.com/owa/calendar/AdultoTeleNeBP@bp.org.br/bookings/`

Agenda Infantil:  
`https://outlook.office365.com/owa/calendar/PeditricoTeleNEBP@bp.org.br/bookings/`

## ■ MODO DE OPERAÇÃO

### *Variável de Controle:*

```
DRY_RUN = True # Modo teste (não executa ações reais)
DRY_RUN = False # Modo produção (executa realmente)
```

### *Browser:*

- **Desenvolvimento:** Headful (visualiza automação)
- **Produção:** Headless (sem interface)

## ■ ESTRUTURA DE DADOS NOTION

### *Campos Extraídos:*

```
{
  "id": "page_id",
  "nome": "Nome do Paciente",
  "cpf": "123.456.789-00",
  "especialidade": "Cardiologia",
  "motivo": "Consulta de rotina",
  "acs": "Nome do ACS",
  "tipo": "Adulto" | "Infantil",
  "status": "Não iniciado" | "Em andamento" | "Concluído" | "Erro"
}
```

### *Especialidades Infantis Detectadas:*

- triagem
- pediatria
- neuropediatria
- psiquiatria infantil
- psiquiatria pediátrica
- endocrinologia pediátrica

*(Todas outras são consideradas "Adulto")*

## ■ COMO USAR

### 1. Pré-requisitos:

```
pip install playwright notion-client google-api-python-client  
playwright install chromium
```

### 2. Configurar:

- Adicionar token Notion
- Configurar credenciais Google Calendar
- Ajustar DRY\_RUN conforme necessário

### 3. Executar:

```
python agendador_final_corrigido.py
```

## ■ CASOS DE USO

### *Cenário 1: Agendamento Simples*

1. Tarefa criada no Notion: "Agendar João Silva - Cardiologia"
2. Bot detecta tarefa "Não iniciado"
3. Verifica disponibilidade no Google Calendar
4. Acessa Telenordeste automaticamente
5. Agenda e confirma
6. Atualiza Notion: "Concluído"
7. Cria evento no Google Calendar

### *Cenário 2: Múltiplas Tarefas*

- Bot processa TODAS tarefas "Não iniciado"
- Uma por vez, sequencialmente
- Logs independentes para cada

### *Cenário 3: Falha no Agendamento*

- Captura erro
- Atualiza Notion com status "Erro"
- Continua para próxima tarefa

## ■ MELHORIAS POSSÍVEIS

#### 1. Segurança:

- Usar variáveis de ambiente para tokens
- Implementar rate limiting

#### 2. Confiabilidade:

- Retry automático (3x)

- Circuit breaker para falhas consecutivas
- Notificações por email/Telegram

### 3. **Performance:**

- Processamento paralelo (múltiplas agendas)
- Cache de horários disponíveis

### 4. **UX:**

- Dashboard de monitoramento
- Relatórios de execução
- Estatísticas de sucesso/falha

## ■ CONCLUSÃO

Este bot é uma solução **robusta e bem estruturada** para automação de agendamentos médicos.

### *Pontos Fortes:*

- ■ Integração tripla (Notion + Google Calendar + Web)
- ■ Dupla checagem de confirmação
- ■ Logs detalhados
- ■ Tratamento extensivo de erros
- ■ Suporte a múltiplos tipos de agenda

### *Status Atual:*

- ■ 95% funcional
- ■ Encontra horários disponíveis corretamente
- ■ Precisa validação completa em produção

**Analisado em:** 2025-01-10

**Versão do arquivo:** agendador\_final\_corrigido.py

**Analista:** AI Agent Luna