

## # Checklist de Implementações para a Integração do Agente PepeAi

### ## Backend

1. Criar arquivos do backend em `backend/src/services/AIServices/`
  - AiService.ts
  - IntraService.ts
  - AiControlService.ts
  - AiConfigService.ts
2. Configurar tabelas no banco de dados (PostgreSQL)
  - Adicionar coluna `ai\_agent\_enabled` na tabela `plans`
  - Criar tabela `ai\_configs` para configurações por empresa
  - Criar tabela `ai\_sessions` para sessões ativas
  - Adicionar coluna `ai\_assisted` na tabela `tickets`
  - Criar usuário `PepeAi@ai.system` na tabela `users`
3. Implementar integração com OpenRouter (DeepSeek)
  - Configurar modelo `deepseek/deepseek-chat-v3-0324:free`
  - Gerenciar limite de 20 requests/minuto
4. Integrar lógica no `wbotMessageListener.ts`
  - Verificar plano e habilitar PepeAi
  - Atribuir tickets ao usuário PepeAi
  - Processar respostas e gerenciar fallback
5. Configurar sistema de busca semântica
  - Habilitar extensão `vector` no PostgreSQL
  - Criar tabelas `knowledge\_embeddings` e `question\_embeddings`
  - Implementar serviço `SemanticSearchService.ts`
6. Criar middleware de controle de limites
  - Verificar `ai\_agent\_enabled` no plano
  - Controlar limites de requests mensais
  - Validar tamanho da base de conhecimento

### ## Frontend

7. Adicionar nova aba "Pepe AI" nos atendimentos
  - Atualizar `ticketTabs` com status `ai\_assisted`
8. Criar aba de configuração de IA nas Integrações
  - Checkbox para ativar/desativar PepeAi
  - Campo para treinamento de informações da empresa
  - Restringir acesso por plano
9. Implementar controles visuais no chat
  - Botão para pausar IA

- Botão para reativar IA
- Indicador visual de IA ativa

## ## Sistema

### 10. Configurar sistema de alertas

- Implementar tipos de alerta (`NEEDS\_HUMAN`, `TECHNICAL\_ERROR`, `CUSTOMER\_R
- Notificar operadores em caso de falhas

### 11. Configurar treinamento personalizado

- Criar campos para informações da empresa (nome, políticas, FAQs)
- Implementar extração automática de sites
- Configurar prompt personalizado

### 12. Implementar parâmetros por plano

- Configurar limites (requests, tokens, modelos) conforme tabela
- Habilitar busca semântica para planos Pro/Enterprise
- Configurar níveis de analytics

### 13. Configurar controles de segurança e qualidade

- Implementar prompt restritivo para evitar alucinações
- Validar respostas incertas
- Configurar monitoramento (taxa de resolução, tempo de resposta, etc.)

### 14. Testar integração completa

- Validar fluxo de atendimento
- Testar fallback para humanos
- Verificar desempenho da busca semântica