



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

Curso: Ciência de Dados e Inteligência Artificial

Disciplina: PI: Programação Web e Banco de Dados Relacionais

Professor: Fernando Luiz de Almeida Silveira

Grupo: Venda.ai – Cauã Migotto 24014749, João Sydow 24780363, Lucas Chini Mendonça 24780215, Matheus Tasso 24007247, Vinicius Pinheiro Livinalli 24002125, Yuri Figueiredo 24004315

DOCUMENTAÇÃO PADRÃO PARA PROJETO DE APLICATIVO

Resumo

Este documento tem como objetivo apresentar as etapas para a documentação de um projeto de desenvolvimento de aplicativos. O documento do projeto que será entregue deve seguir as normas da ABNT, descrevendo as fases essenciais abaixo para garantir a organização, qualidade e rastreabilidade do desenvolvimento do software.

1. Introdução

O desenvolvimento de aplicativos requer um planejamento estruturado para garantir a qualidade e funcionalidade do sistema final. A documentação do projeto é fundamental para registrar decisões, requisitos, processos e soluções implementadas. Este documento apresenta as cinco etapas essenciais para a documentação do projeto.

2. Etapas da Documentação

2.1. Levantamento de Requisitos

2.1.1 Objetivos

Auxiliar vendedores durante chamadas e reuniões, sugerindo perguntas, abordagens e informações relevantes. Melhorar a performance comercial por meio de insights em tempo real. Centralizar informações estratégicas sobre clientes e negociações.

2.1.2 Funcionalidades

Análise automática de chamadas (áudio e/ou vídeo) com retorno imediato de sugestões. Chat integrado para consultas rápidas sobre clientes, produtos ou contexto.

Histórico de interações, permitindo aprendizado contínuo da IA. Painel administrativo para configuração e gestão de usuários.

2.1.3 Restrições Técnicas

Processamento de áudio deve ocorrer de forma segura e conforme LGPD. Necessidade de conexão estável com a internet. Integração com APIs de reconhecimento de voz e NLP (Processamento de Linguagem Natural).

2.1.4 Personas

Persona 1 – Carlos, o Vendedor Júnior

- Idade: 23 anos
- Cargo: Vendedor em início de carreira
- Desafios: insegurança em chamadas, dificuldade em saber o que perguntar.
- Objetivo: aprender a conduzir vendas com confiança.
- Como o Venda.Aí ajuda: sugere perguntas e caminhos durante a ligação.

Persona 2 – Fernanda, a Executiva de Contas

- Idade: 31 anos
- Cargo: Executiva de vendas B2B
- Desafios: muitos clientes, dificuldade em lembrar informações específicas.
- Objetivo: aumentar a taxa de conversão em negociações.
- Como o Venda.Aí ajuda: traz histórico e insights rápidos durante a reunião.

Persona 3 – Roberto, o Gestor Comercial

- Idade: 42 anos
- Cargo: Gerente de Vendas
- Desafios: acompanhar desempenho da equipe e padronizar processos.
- Objetivo: melhorar performance e fechar mais negócios.
- Como o Venda.Aí ajuda: fornece relatórios de chamadas e apoia no treinamento da equipe.

2.1.5 Requisitos Funcionais

RF01: Permitir login seguro de usuários.

RF02: Captar e processar áudio das chamadas.

RF03: Exibir sugestões de perguntas ou respostas em tempo real ao longo da reunião.

RF04: Manter um chat para consultas instantâneas.

RF05: Salvar histórico de interações.

2.1.6 Requisitos Não Funcionais

RNF01: Garantir resposta da IA em até 2 segundos.

RNF02: Interface intuitiva e responsiva.

RNF03: Cumprir requisitos da LGPD quanto à privacidade de dados.

RNF04: Suporte a alta disponibilidade (99,5% uptime).



2.2. Planejamento

No planejamento, definimos como o projeto será desenvolvido, incluindo metodologias e ferramentas a serem utilizadas. Deve conter:

- Definição do escopo

1- Objetivos Principais:

Auxiliar vendedores em chamadas e reuniões com sugestões de perguntas, abordagens e informações relevantes.

Fornecer insights em tempo real para melhorar a performance comercial.

Centralizar dados estratégicos sobre clientes, negociações e histórico de interações.

Garantir aprendizado contínuo da IA com base em interações passadas.

Oferecer um painel administrativo para configuração e gestão de usuários.

2- Limites do Projeto:

O sistema é voltado para assistência em chamadas de áudio/vídeo em tempo real, com integração a ferramentas de CRM e APIs de IA.

Não inclui funcionalidades offline (planejadas para médio prazo) ou suporte a telefonia tradicional (planejado para longo prazo).

O escopo inicial prioriza ambientes web e mobile responsivos, sem ênfase em realidade aumentada ou análise preditiva (futuras).

- Tecnologia e ferramentas escolhidas

1- Frontend (Interface do Usuário):

- Framework Principal: React.Js e Electron

2- Backend (Lógica de Negócios e APIs):

- Linguagem e Framework: Python com FastAPI

- APIs Externas: Axios ou Fetch para integrações com Zoom/Teams/Meet e CRMs

3- Banco de Dados

- Principal: PostgreSQL
- Cache e Tempo Real: Redis

4- IA e Processamento de Áudio

- Reconhecimento de Voz: Google Cloud Speech-to-Text ou AssemblyAI
- NLP e IA: OpenAI API (GPT models) ou Gemini
- Aprendizado Contínuo: MLflow ou TensorFlow para treinar modelos com feedback de usuários

5- Gerenciamento de Projeto e Versionamento

- Gerenciamento Ágil: Jira
- Controle de Versão: Git com GitHub

6- Implantação e DevOps

Para multi-ambientes (DEV, TEST, STAGING, PROD) com CI/CD:

- Containerização: Docker
- CI/CD: GitHub Actions

- Metodologia de desenvolvimento (Scrum, Kanban, etc.)

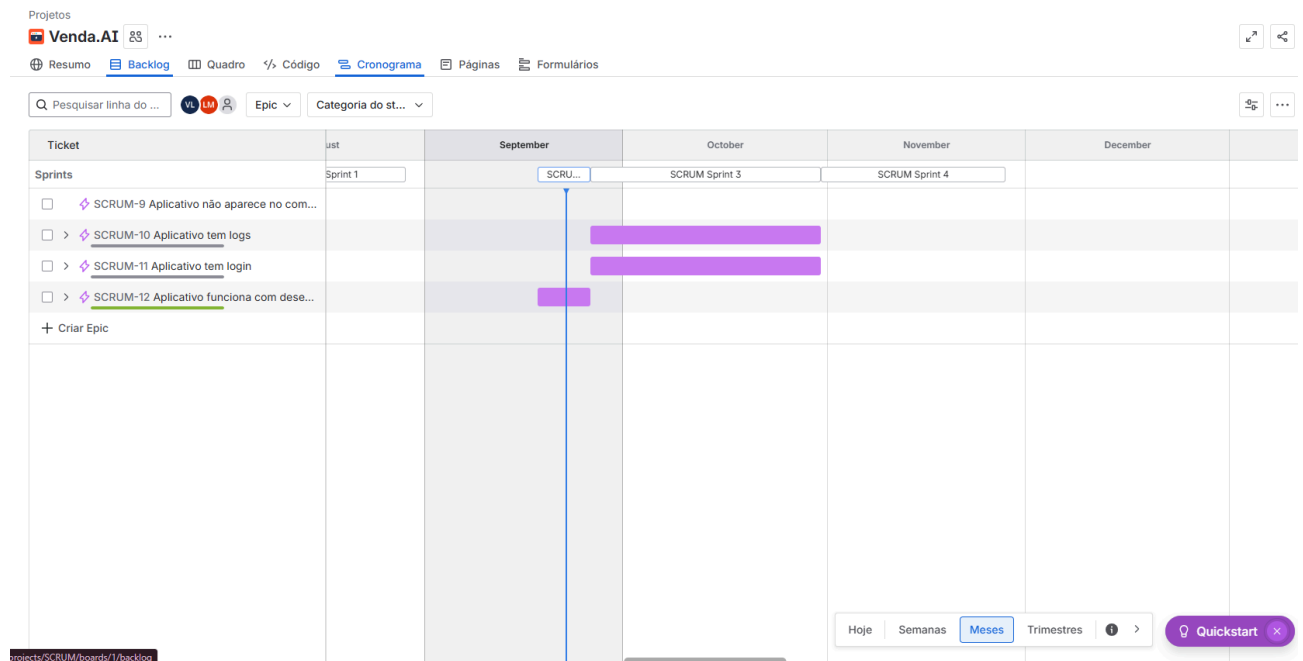
Estamos utilizando Scrum com sprints planejadas e definidas e Kanban para organizar as entregas de cada sprint.

- Cronograma e sprints de trabalho

The screenshot displays a Jira project board for 'Venda.AI'. The interface includes a top navigation bar with tabs for 'Resumo', 'Backlog', 'Quadro', 'Código', 'Cronograma', 'Páginas', and 'Formulários'. Below the navigation bar is a search bar and a filter dropdown set to 'Epic'. The main area shows three Scrum sprints:

- SCRUM Sprint 2** (18 set – 25 set, 6 tickets): This sprint is active. It contains six tasks:
 - SCRUM-4: Usuário precisa realizar perguntas rapidamente (Status: A FAZER)
 - SCRUM-5: Programa precisa captar audio (Status: A FAZER)
 - SCRUM-13: Requisitos funcionais (Status: APLICATIVO FUNCIO... CONCLUÍDO)
 - SCRUM-7: Site comercial para converter clientes (Status: EM ANÁLISE)
 - SCRUM-16: Planejamento e desing do banco de dados (Status: EM ANÁLISE)
 - SCRUM-17: Bug de build no electron está nos impedindo de continuar (Status: EM ANDAMENTO, 30 de set.)
- SCRUM Sprint 1** (4 ago – 28 ago, 0 tickets): This sprint is inactive. It contains a placeholder text: 'Para planejar um sprint, arraste tickets no sprint ou arraste o rodapé do sprint.'
- SCRUM Sprint 3** (26 set – 30 out, 3 tickets): This sprint is inactive. It contains two tasks:
 - SCRUM-6: Programa precisa salvar contexto (Status: A FAZER)
 - SCRUM-3: Usuário precisa ter acesso ao relatório da conversa (Status: APLICATIVO TEM LO... A FAZER)

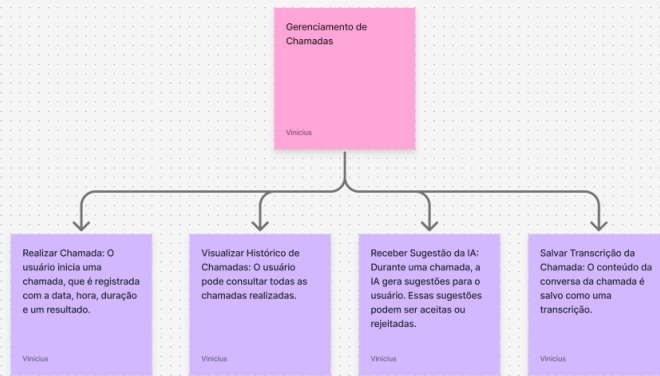
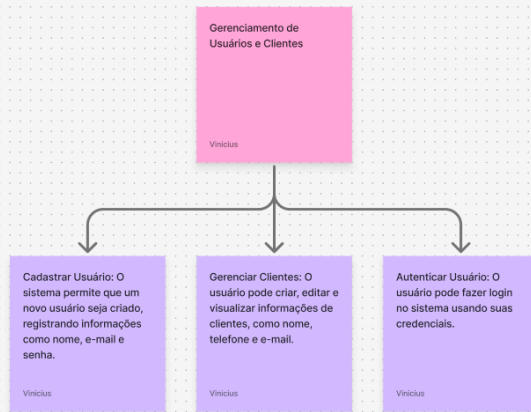
At the bottom right, there is a 'Quickstart' button and a '0 ticket | Estimativa: 0' indicator.



2.3. Desenvolvimento

A implementação do código deve ser documentada para garantir a compreensão do sistema. Inclui:

- Estrutura do código e padrões adotados
- Modelos de dados e diagramas (UML, MER/DER)



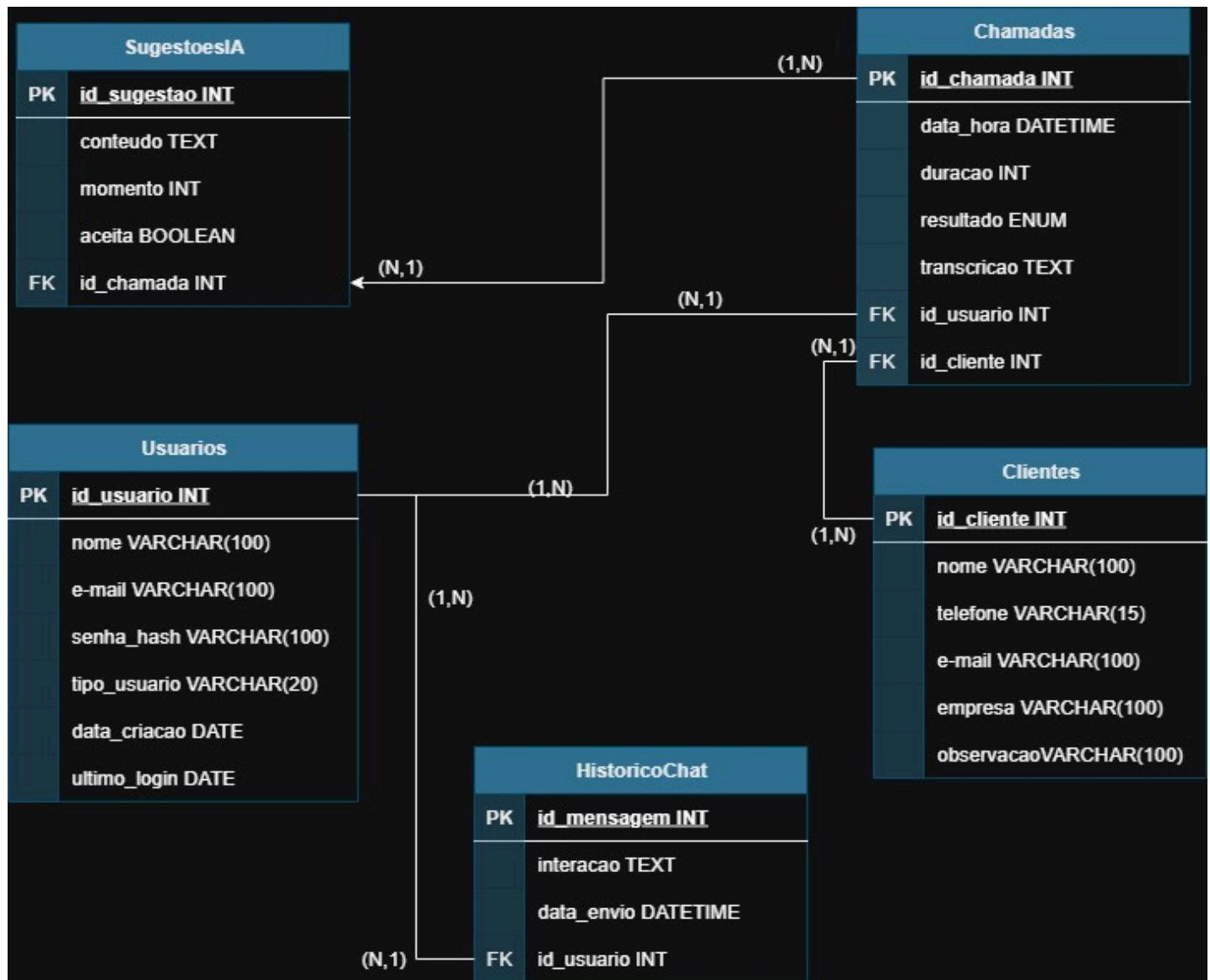
Gerenciamento de Chat

Vinicius



Registrar Interação de Chat: O sistema armazena o histórico de interações de chat entre o usuário e o cliente.

Vinicius



- Controle de versão (Git, GitHub, GitLab)

Versionamento Semântico no formato MAJOR.MINOR.PATCH. O objetivo principal é comunicar as mudanças em uma nova versão, permitindo que os desenvolvedores compreendam o impacto da atualização sem precisar analisar o código.

- MAJOR:** Indica mudanças incompatíveis com versões anteriores da API. Um incremento neste número exige que os usuários da biblioteca ou API atualizem seu código, pois a nova versão contém alterações que quebram a compatibilidade.
- MINOR:** Indica a adição de novas funcionalidades de forma retrocompatível. O incremento deste número significa que a nova versão inclui novas features, mas não quebra a compatibilidade com a versão anterior.
- PATCH:** Indica correções de bugs retrocompatíveis. O incremento deste número significa que a nova versão contém apenas correções de bugs ou pequenas melhorias que não alteram a funcionalidade principal ou a API.

2.4. Testes e Qualidade

Garantir que o aplicativo funcione corretamente em todas as situações previstas.

Considerando a natureza crítica da aplicação (tempo real + IA + vendas), trabalharemos com três tipos de testes essenciais:

Testes Funcionais

Testes de Integração de API

Integração com APIs de chamadas (Zoom, Teams, Meet)
Conexão com CRMs (Salesforce, HubSpot, Pipedrive)
APIs de processamento de linguagem natural
Sincronização de dados em tempo real

Testes de Processamento de Áudio

Captura de áudio em diferentes qualidades
Transcrição de fala em tempo real
Detecção de diferentes idiomas/sotaques
Filtragem de ruídos de fundo

Testes de IA e Sugestões

Precisão das sugestões contextuais
Tempo de resposta das recomendações
Relevância das sugestões por segmento de cliente
Aprendizado com feedback do usuário

Testes de Performance

Testes de Latência

Tempo entre fala do cliente e sugestão
Responsividade da interface durante chamadas
Sincronização áudio-texto-sugestões

Testes de Carga

Múltiplas chamadas simultâneas
Alto volume de processamento de dados
Consumo de recursos do sistema (CPU, RAM, rede)

Testes de Escalabilidade

Comportamento com equipes grandes
Performance com histórico extenso de chamadas

Testes de Usabilidade

Testes de Interface

Facilidade de uso durante chamadas ativas
Visibilidade das sugestões sem interferir na conversa
Acessibilidade para diferentes perfis de usuário

Testes de Workflow

Fluxo completo: pré-chamada → chamada → pós-chamada
Integração com processo de vendas existente
Exportação de relatórios e insights

Testes de Segurança e Privacidade

Testes de Proteção de Dados

Criptografia de áudio e transcrições
Conformidade com LGPD/GDPR
Armazenamento seguro de dados sensíveis

Testes de Autenticação

Login seguro e gestão de sessões
Controle de acesso por níveis de usuário
Integração com SSO corporativo

Testes Específicos do Contexto

Testes de Cenários de Venda

Diferentes tipos de objeções de clientes
Vários perfis de compradores (decisor, influenciador, etc.)
Produtos/serviços com complexidades distintas

Testes de Ambiente Real

Diferentes dispositivos de áudio
Conexões de internet variadas
Ambientes ruidosos ou silenciosos

Ferramentas Utilizadas

Automação de Testes

Selenium/Playwright para testes de interface
Jest/Cypress para testes unitários e e2e
Postman/Newman para APIs

Performance

JMeter ou Artillery para testes de carga
Lighthouse para performance web
Profilers específicos da linguagem usada

Monitoramento

Sentry ou Bugsnag para tracking de erros
New Relic ou DataDog para performance em produção
Logs estruturados para debugging

Relatório de Bugs e Melhorias Implementadas

Visão Geral

Esta seção documenta os principais bugs identificados durante o desenvolvimento e testes do aplicativo de assistência de vendas com IA, bem como as melhorias implementadas com base no feedback dos usuários e análises de performance.

Metodologia de Tracking

Ferramentas Utilizadas

Sistema de Tracking: GitHub Issues / Jira

Monitoramento em Produção: Sentry

Logs Estruturados: Winston/Logstash

Métricas de Performance: New Relic

Classificação de Severidade

Crítico: Impede o uso da aplicação ou causa perda de dados

Alto: Funcionalidade principal não funciona como esperado

Médio: Funcionalidade secundária com problemas ou UX prejudicada

Baixo: Problemas cosméticos ou melhorias de conveniência

Roadmap de Melhorias Futuras

Próxima Sprint (Q2 2025)

Suporte para videochamadas em grupo

Análise de emoções via reconhecimento facial

API pública para integrações personalizadas

Médio Prazo (Q2 - 2 2025)

Modelo de IA treinado especificamente para vendas B2B brasileiras

Suporte offline para situações de conectividade limitada

Longo Prazo (H2 2026)

Integração com telefonia (chamadas de voz)

Realidade aumentada para apresentações de vendas

Análise preditiva de intenção de compra

2.5. Implantação e Manutenção

Por fim, o sistema é colocado em produção e mantido para correções e atualizações. Deve contemplar:

Arquitetura de Deploy

O sistema utiliza uma arquitetura multi-ambiente com pipeline automatizada de CI/CD:

Ambientes

1. Desenvolvimento (DEV)

- Finalidade:** Desenvolvimento ativo e testes iniciais
- Infraestrutura:** Docker containers locais
- Dados:** Dataset sintético e anonimizado
- Deploy:** Automático a cada commit na branch `develop`

2. Teste (TEST)

- Finalidade:** Testes automatizados e validação de qualidade
- Dados:** Dados de teste controlados
- Deploy:** Automático após aprovação dos testes unitários

3. Homologação (STAGING)

- Finalidade:** Validação de usuário e testes de aceitação
- Infraestrutura:** Mirror da produção com 30% da capacidade
- Dados:** Cópia sanitizada da produção (último snapshot)
- Deploy:** Manual, após aprovação em TEST

4. Produção (PROD)

- Finalidade:** Ambiente live para usuários finais
- Infraestrutura:** auto-scaling
- Dados:** Dados reais com backup contínuo
- Deploy:** Manual com janela de manutenção programada

Estratégias de Atualização e Suporte

1. Versionamento e Release Strategy

Semantic Versioning

- MAJOR** (1.0.0 → 2.0.0): Breaking changes na API
- MINOR** (1.0.0 → 1.1.0): Novas funcionalidades compatíveis
- PATCH** (1.0.0 → 1.0.1): Bug fixes e melhorias menores

Release Schedule

Releases Maiores: Trimestralmente

Releases Menores: Mensalmente

Patches: Conforme necessário (24-48h para críticos)

Hotfixes: Imediatamente para vulnerabilidades de segurança

Suporte ao Cliente

Multi-canal: E-mail, chat, phone, Slack

SLA por Tier: Enterprise, Professional, Starter

Self-service: Status page, documentation, forum

Manutenção Preventiva: Automated tasks e health checks

3. Considerações Finais

O desenvolvimento do aplicativo de assistência de vendas com inteligência artificial representa um marco significativo na aplicação de tecnologias emergentes para resolver problemas reais do mundo dos negócios. Este projeto demonstra como a combinação estratégica de processamento de linguagem natural, análise em tempo real e interfaces intuitivas pode transformar fundamentalmente a experiência de vendas.