Produto a ser desenvolvido deve ter

**Site:** chat ao vivo

**Vendas:** CRM, vendas, faturamento, restaurante, serviços

**Operações:** Financeiro, central de atendimento, compra

**Para consulta utilize o modelo do site https://www.odoo.com/pt\_BR/trial**

Documento básico para iniciar os passos de construção do sistema Envision ERP integrado com CRM.

ERP (Enterprise Resource Planning) foca na gestão interna da empresa, otimizando processos como:

1. Finanças: controle de fluxo de caixa, contas a pagar e receber, lançamentos contábeis, geração de relatórios financeiros e muito mais.
2. Estoque: controle de entrada e saída de produtos, níveis de estoque, gestão de armazéns, pedidos de compra e muito mais.
3. Produção: planejamento de produção, controle de qualidade, gestão de materiais, roteiros de produção e muito mais.
4. Recursos Humanos: folha de pagamento, controle de ponto, recrutamento e seleção, gestão de talentos e muito mais.

CRM (Customer Relationship Management) foca no relacionamento com o cliente, otimizando processos como:

1. Vendas: gestão de leads e oportunidades, acompanhamento do funil de vendas, cotações, propostas e fechamentos de vendas.
2. Marketing: campanhas de marketing, nutrição de leads, segmentação de clientes, personalização de mensagens e muito mais.
3. Atendimento ao cliente: registro de chamados, resolução de problemas, gestão de SLA, pesquisas de satisfação e muito mais.

Ao final temos que obter:

1. decisões mais inteligentes: com base em dados precisos e atualizados de todas as áreas da empresa.
2. Aumente a eficiência: automatizando tarefas repetitivas e manuais, liberando tempo para que os funcionários se concentrem em atividades mais estratégicas.
3. Melhore a produtividade: otimizando os processos internos e externos da empresa.
4. Reduza custos: eliminando retrabalho e desperdícios.
5. Aumente as vendas: fidelizando clientes e conquistando novos.
6. Tenha uma visão 360 graus do cliente: entendendo suas necessidades e preferências para oferecer um atendimento personalizado.

**Desenvolvendo seu software de ERP e CRM para empresas de médio porte:**

**1. Defina o escopo do projeto:**

* **Funcionalidades:** Quais funcionalidades o seu software terá? Quais processos ele irá automatizar? Quais dados ele irá coletar e armazenar?
* **Público-alvo:** Para quais empresas de médio porte o seu software será direcionado? Quais são as suas necessidades específicas?
* **Orçamento:** Quanto você tem disponível para investir no desenvolvimento do software?
* **Prazo:** Qual o prazo para o desenvolvimento do software?

**2. Escolha a tecnologia:**

* **Linguagem de programação:** Qual linguagem de programação você irá usar para desenvolver o software?
* **Banco de dados:** Qual banco de dados você irá usar para armazenar os dados do software?
* **Plataforma:** Em qual plataforma o software será executado (web, desktop, mobile)?

**3. Desenvolva o software:**

* **Comece com um protótipo:** Desenvolva um protótipo do software para testar as funcionalidades e a interface do usuário.
* **Faça testes rigorosos:** Teste o software completamente para garantir que ele esteja funcionando corretamente e sem bugs.
* **Obtenha feedback:** Obtenha feedback de usuários potenciais para melhorar o software.

**4. Implemente o software:**

* **Treine os usuários:** Treine os usuários do software para que eles possam usá-lo de forma eficaz.
* **Ofereça suporte:** Ofereça suporte técnico aos usuários do software.

**5. Mantenha o software:**

* **Corrija bugs:** Corrija quaisquer bugs que sejam encontrados no software.
* **Adicione novas funcionalidades:** Adicione novas funcionalidades ao software de acordo com as necessidades dos usuários.
* **Atualize o software:** Atualize o software regularmente para garantir que ele esteja compatível com as últimas tecnologias.

**Recursos adicionais:**

* **Modelos de software de ERP e CRM:** Você pode encontrar modelos de software de ERP e CRM online que podem te ajudar a começar o desenvolvimento do seu software.
* **Empresas de desenvolvimento de software:** Você pode contratar uma empresa de desenvolvimento de software para te ajudar a desenvolver o seu software.
* **Comunidades online:** Existem diversas comunidades online onde você pode encontrar ajuda e conselhos de outros desenvolvedores de software.

**Lembre-se:**

* **O desenvolvimento de software é um processo complexo que exige tempo, esforço e recursos.**
* **É importante planejar cuidadosamente o seu projeto antes de começar a desenvolver o software.**
* **Existem diversas ferramentas e recursos disponíveis para te ajudar a desenvolver o seu software.**
* **Não hesite em pedir ajuda se precisar.**

**Com dedicação e trabalho duro, você pode desenvolver um software de ERP e CRM de sucesso para empresas de médio porte.**

**ERP vs CRM: Funcionalidades Separadas para o Seu Protótipo**

**ERP (Enterprise Resource Planning - Sistema de Gestão Empresarial): Foco na Gestão Interna**

**Funcionalidades Essenciais:**

* **Módulo Financeiro:**
  + **Contas a Pagar e Receber:** Gerencie entradas e saídas de dinheiro, datas de vencimento, valores e históricos.
  + **Fluxo de Caixa:** Monitore o fluxo de entrada e saída de dinheiro em tempo real, com gráficos, indicadores e previsões.
  + **Lançamentos Contábeis:** Automatize lançamentos para todas as transações financeiras, integre-se com softwares de contabilidade e permita lançamentos manuais.
  + **Relatórios Financeiros:** Gere relatórios completos (balanço patrimonial, DRE, DFC), personalize-os e exporte-os em diferentes formatos.
* **Módulo Estoque:**
  + **Controle de Entrada e Saída:** Registre entradas e saídas com datas, quantidades, valores, documentos fiscais, gerencie lotes e validades e integre-se com PDVs.
  + **Níveis de Estoque:** Monitore os níveis em tempo real, defina alertas para níveis mínimos, gere relatórios de giro de estoque e identifique produtos com baixa rotatividade.
  + **Gestão de Armazéns:** Controle a localização dos produtos, otimize a organização do armazém e integre-se com leitores de código de barras.
  + **Pedidos de Compra:** Automatize a geração de pedidos com base nos níveis de estoque e na demanda, integre-se com fornecedores e permita pedidos manuais.
* **Módulo Vendas:**
  + **Fechamentos de Vendas:** Registre fechamentos com datas, valores, formas de pagamento e informações dos clientes, gere relatórios de vendas e analise os dados para identificar tendências e oportunidades.

**Funcionalidades Adicionais:**

* **Módulo Produção:** Controle a produção, gerencie materiais, roteiros de produção e expedição de produtos.
* **Módulo RH:** Folha de pagamento, controle de ponto, recrutamento e seleção, gestão de talentos.
* **Módulo Compras:** Gestão de fornecedores, negociação de compras, controle de pedidos de compra e gerenciamento de estoques.

**CRM (Customer Relationship Management - Gestão do Relacionamento com o Cliente): Foco no Cliente**

**Funcionalidades Essenciais:**

* **Módulo Clientes:**
  + **Cadastro de Clientes:** Cadastre clientes com informações de contato, endereço, histórico de compras e preferências, segmente-os e integre-se com ferramentas de CRM.
  + **Histórico de Compras:** Mantenha um histórico completo de compras de cada cliente, ofereça a opção de visualização online e utilize-o para personalizar ofertas e promoções.
  + **Segmentação:** Segmente clientes por localização, histórico de compras, perfil de consumo e outros critérios para criar campanhas de marketing direcionadas.
  + **Comunicação com Clientes:** Envie e-mails, SMS, notificações push e outros tipos de comunicação para manter os clientes informados e engajados.
* **Módulo Marketing:**
  + **Campanhas de Marketing:** Crie e gerencie campanhas de marketing em diferentes canais (e-mail, mídias sociais, etc.), personalize-as para cada segmento de clientes e monitore os resultados.
  + **Nutrição de Leads:** Nutra leads com conteúdo relevante para convertê-los em clientes, utilize automação de marketing e acompanhe o progresso dos leads no funil de vendas.
  + **Personalização de Mensagens:** Personalize mensagens de marketing com base no perfil e histórico de compras de cada cliente para aumentar a efetividade das campanhas.
  + **Análise de Resultados de Marketing:** Analise os resultados das campanhas de marketing para identificar o que funciona e o que precisa ser melhorado, otimizando as estratégias.
* **Módulo Atendimento ao Cliente:**
  + **Registro de Chamados:** Registre chamados de clientes com informações sobre o problema, prioridade e canal de atendimento, defina prazos de resolução e acompanhe o andamento dos chamados.
  + **Resolução de Problemas:** Resolva os problemas dos clientes de forma rápida e eficiente, utilize a base de conhecimento da empresa e ofereça diferentes canais de atendimento.
  + **Gestão de SLA:** Defina e gerencie SLAs (Service Level Agreements) para garantir a qualidade do atendimento ao cliente e monitore o cumprimento dos prazos acordados.
  + **Pesquisas de Satisfação:** Realize pesquisas de satisfação para avaliar a qualidade do atendimento ao cliente, identificar áreas de melhoria e fidelizar os clientes.

\*\*Funcional

**Integração de ERP e CRM com Bancos de Dados Relacionais e NoSQL: Uma Abordagem Estratégica**

A integração de um sistema ERP (Enterprise Resource Planning) e CRM (Customer Relationship Management) com bancos de dados relacionais e NoSQL exige um planejamento estratégico e uma implementação cuidadosa para garantir a eficiência, a confiabilidade e a escalabilidade da solução.

**Abordagens de Integração:**

**1. Integração Direta:**

* **Conexão via APIs:** Estabeleça conexões diretas entre os sistemas e os bancos de dados através de APIs (Interfaces de Programação de Aplicações).
  + **Vantagens:** Simplicidade, controle e flexibilidade.
  + **Desvantagens:** Complexidade de gerenciamento, responsabilidade pela integração e potencial gargalo de desempenho.

**2. Camada de Integração:**

* **Implementação de um middleware:** Utilize um middleware para centralizar a comunicação entre os sistemas e os bancos de dados, abstraindo as complexidades da integração.
  + **Vantagens:** Modularidade, reuso de código, facilidade de gerenciamento e escalabilidade.
  + **Desvantagens:** Custo adicional, aumento da complexidade e potencial dependência do middleware.

**3. Arquitetura de Microsserviços:**

* **Desenvolvimento de microsserviços autônomos:** Crie microsserviços independentes para cada funcionalidade, com integração através de mensagens assíncronas.
  + **Vantagens:** Alta escalabilidade, flexibilidade, resiliência e facilidade de desenvolvimento e manutenção.
  + **Desvantagens:** Complexidade de implementação, necessidade de orquestração de mensagens e potencial aumento da latência.

**Escolha do Banco de Dados:**

**1. Banco de Dados Relacional:**

* Ideal para armazenar dados estruturados e transacionais, como:
  + **ERP:** Contas a pagar e receber, estoque, pedidos de compra, etc.
  + **CRM:** Dados de clientes, histórico de compras, segmentação, etc.
  + **Vantagens:** Maturidade, confiabilidade, ACID (Atomicidade, Consistência, Isolamento e Durabilidade) e suporte robusto de ferramentas e profissionais.
  + **Desvantagens:** Menos escalável para grandes volumes de dados não estruturados, menos flexível para mudanças nos requisitos de negócio e desempenho inferior em algumas operações com dados não estruturados.

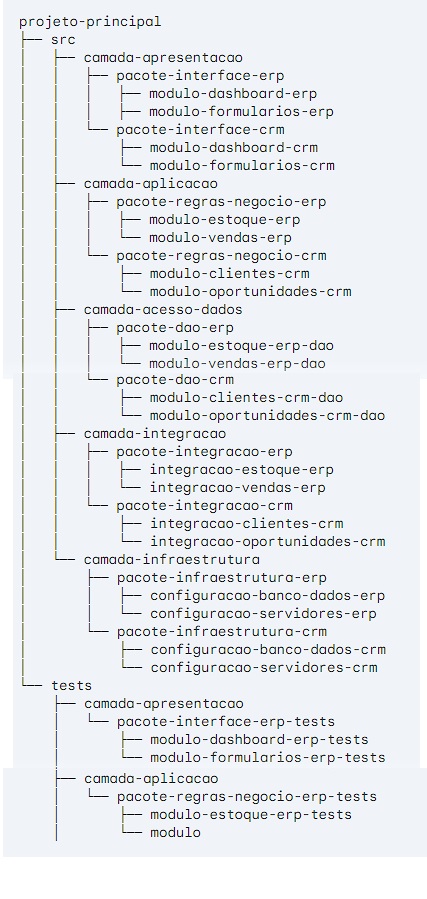
**2. Banco de Dados NoSQL:**

* Ideal para armazenar dados não estruturados e análises complexas, como:
  + **CRM:** Interações com clientes, perfis comportamentais, dados de mídias sociais, etc.
  + **Vantagens:** Alta escalabilidade, flexibilidade para dados não estruturados, alto desempenho em consultas complexas e grandes volumes de dados.
  + **Desvantagens:** Menos maduro que os bancos relacionais, com menos suporte de ferramentas e profissionais, ACID não garantido em todos os bancos NoSQL e consultas complexas podem ser mais desafiadoras.

**Considerações Importantes:**

* **Defina os requisitos de integração:** Volume e tipo de dados, padrões de acesso, escalabilidade, desempenho e custos.
* **Avalie as tecnologias disponíveis:** Bancos de dados relacionais, NoSQL, APIs, middleware e microsserviços.
* **Projete a arquitetura de integração:** Escolha a abordagem mais adequada (integração direta, camada de integração ou microsserviços) e defina os componentes da arquitetura.
* **Implemente a integração:** Siga as melhores práticas de desenvolvimento e utilize ferramentas adequadas.
* **Teste e monitore a integração:** Garanta a confiabilidade, o desempenho e a segurança da integração.

**Exemplo de Estrutura:**



**Desenvolvendo um ERP como Engenheiro de Software: Guia Detalhado**

**Introdução:**

Assumir o papel de engenheiro de software para desenvolver um ERP (Enterprise Resource Planning) exige um planejamento meticuloso, metodologias ágeis, expertise técnica e foco na entrega de valor ao negócio. Este guia detalhado irá te auxiliar nas etapas essenciais para o desenvolvimento de um ERP robusto e eficaz.

**1. Fase de Planejamento e Análise:**

* **Compreensão dos Requisitos:**
  + Realize workshops e entrevistas com stakeholders para entender profundamente as necessidades do negócio, os processos existentes, os pontos de dor e os objetivos estratégicos.
  + Documente detalhadamente os requisitos funcionais e não funcionais, incluindo casos de uso, fluxos de trabalho e indicadores-chave de desempenho (KPIs).
* **Análise da Situação Atual:**
  + Avalie os sistemas legados em uso, identificando pontos fortes, fracos e oportunidades de melhoria.
  + Mapeie os processos de negócio existentes, seus fluxos de dados e as interfaces com sistemas externos.
  + Defina os módulos essenciais do ERP, como gestão de estoque, compras, vendas, finanças, produção e RH.
* **Definição de Escopo e Prioridades:**
  + Estabeleça o escopo inicial do projeto, priorizando as funcionalidades mais críticas para o negócio.
  + Defina um cronograma realista e etapas de desenvolvimento com marcos bem definidos.
  + Escolha a metodologia de desenvolvimento mais adequada, como Scrum ou Kanban, para gerenciar o projeto de forma ágil e flexível.

**2. Fase de Design e Arquitetura:**

* **Arquitetura de Software:**
  + Projete uma arquitetura robusta e escalável, considerando fatores como volume de dados, usuários simultâneos, performance e segurança.
  + Utilize padrões de arquitetura adequados, como SOA (Service-Oriented Architecture) ou microserviços, para promover modularidade, flexibilidade e reuso de código.
  + Defina a infraestrutura tecnológica necessária, incluindo servidores, bancos de dados, redes e ferramentas de desenvolvimento.
* **Modelagem de Dados:**
  + Crie um modelo de dados conceitual que represente as entidades do negócio, seus atributos e relacionamentos.
  + Utilize diagramas UML (Unified Modeling Language) para visualizar e documentar o modelo de dados com clareza.
  + Normalize o modelo de dados para garantir consistência, integridade e eficiência no acesso aos dados.
* **Interface do Usuário (UI):**
  + Projete uma interface amigável e intuitiva, considerando a usabilidade e a experiência do usuário.
  + Utilize princípios de design de interfaces como simplicidade, consistência, clareza e responsividade.
  + Adapte a interface para diferentes dispositivos e tamanhos de tela, garantindo acessibilidade e usabilidade em desktops, tablets e smartphones.

**3. Fase de Desenvolvimento e Implementação:**

* **Implementação de Módulos:**
  + Siga as melhores práticas de desenvolvimento de software, como codificação modular, testes unitários, integração contínua e entrega contínua.
  + Utilize linguagens de programação adequadas, frameworks e bibliotecas para garantir a qualidade, performance e segurança do código.
  + Implemente os módulos do ERP de forma incremental, priorizando as funcionalidades mais críticas e valiosas para o negócio.
* **Testes e Validação:**
  + Realize testes unitários rigorosos para garantir o funcionamento correto de cada componente do sistema.
  + Execute testes de integração para verificar a interoperabilidade entre os módulos e a comunicação com sistemas externos.
  + Valide o sistema com usuários finais e stakeholders para identificar e corrigir falhas, bugs e usabilidade inadequada.
* **Documentação:**
  + Documente detalhadamente o código, a arquitetura, os casos de uso, os manuais do usuário e as instruções de instalação.
  + Crie documentação técnica completa e acessível para facilitar a manutenção, o suporte e futuras atualizações do sistema.

**4. Fase de Implantação e Suporte:**

* **Implantação e Treinamento:**
  + Implemente o ERP de forma gradual, migrando dados dos sistemas legados e treinando os usuários em suas novas funcionalidades.
  + Ofereça suporte técnico e treinamento aos usuários para garantir a familiarização com o sistema e a resolução de dúvidas.
  + Crie materiais de treinamento abrangentes, como tutoriais, vídeos e guias do usuário, para facilitar o aprendizado e a adoção do ERP.
* **Monitoramento e Manutenção:**
  + Monitore o desempenho do sistema, identificando gargalos e oportunidades de otimização.
  + Realize manutenções preventivas e

### Componentes de um SaaS CRM e ERP

#### 1. **Módulo de CRM**

**a. Gerenciamento de Contatos e Clientes**

* Cadastro e edição de informações de clientes e contatos.
* Registro de interações e histórico de comunicação.
* Segmentação de clientes por diferentes critérios.

**b. Gerenciamento de Vendas**

* Pipeline de vendas para acompanhar oportunidades e negócios.
* Funil de vendas visual para monitorar o progresso das vendas.
* Automação de tarefas de vendas, como envio de e-mails e lembretes.

**c. Suporte ao Cliente**

* Sistema de tickets para gerenciamento de solicitações e problemas dos clientes.
* Base de conhecimento e FAQs.
* Chat ao vivo e integração com sistemas de suporte.

**d. Marketing**

* Campanhas de e-mail marketing.
* Rastreamento de leads e conversões.
* Análises de campanhas de marketing.

#### 2. **Módulo de ERP**

**a. Gerenciamento Financeiro**

* Contabilidade geral.
* Gerenciamento de contas a pagar e a receber.
* Relatórios financeiros e dashboards.

**b. Gerenciamento de Estoque**

* Controle de inventário e rastreamento de estoque.
* Alertas de reabastecimento e gerenciamento de ordens de compra.
* Relatórios de estoque e análise de rotatividade.

**c. Gerenciamento de Compras**

* Criação e acompanhamento de ordens de compra.
* Integração com fornecedores.
* Análise de custos e negociações.

**d. Gerenciamento de Produção**

* Planejamento e controle de produção.
* Rastreamento de ordens de produção e recursos.
* Relatórios de eficiência de produção e qualidade.

**e. Recursos Humanos**

* Cadastro e gerenciamento de funcionários.
* Controle de folhas de pagamento e benefícios.
* Rastreamento de desempenho e avaliação.

### Funcionalidades Comuns

#### 1. **Integração**

* Integração com outras ferramentas de software, como e-mail, calendários e sistemas de e-commerce.
* API para desenvolvimento de integrações personalizadas.

#### 2. **Segurança e Acesso**

* Controle de acesso baseado em funções.
* Auditoria e logs de atividades.
* Segurança de dados e conformidade com regulamentações, como GDPR.

#### 3. **Usabilidade**

* Interface de usuário intuitiva e responsiva.
* Painéis e dashboards personalizáveis.
* Suporte a múltiplos idiomas e moedas.

#### 4. **Relatórios e Análises**

* Relatórios detalhados e customizáveis para todas as áreas do CRM e ERP.
* Análises em tempo real e geração de insights acionáveis.
* Exportação de dados para formatos como Excel e PDF.

### Exemplos de Como Poderia Ser Estruturado

#### **Dashboard Principal**

* Visão geral de métricas chave (KPI) de CRM e ERP.
* Acesso rápido a relatórios financeiros, status de vendas, e desempenho de marketing.

#### **Tela de Gerenciamento de Clientes**

* Lista de clientes com opções de filtragem e busca.
* Detalhes de cada cliente, incluindo histórico de interações, compras e suporte.

#### **Tela de Pipeline de Vendas**

* Visualização em Kanban do pipeline de vendas.
* Possibilidade de arrastar e soltar para mover oportunidades entre estágios.

#### **Tela de Controle de Estoque**

* Visualização do inventário atual.
* Alertas de reabastecimento e status de pedidos de compra.

### Implementação Técnica

**Tecnologias de Backend:**

* Linguagens como Python, Java ou Node.js.
* Banco de dados relacionais (PostgreSQL, MySQL) e/ou NoSQL (MongoDB).

**Tecnologias de Frontend:**

* Frameworks como React, Angular ou Vue.js.

**Infraestrutura:**

* Hospedagem em nuvem (AWS, Azure, Google Cloud).
* Containers e orquestração (Docker, Kubernetes).

### Conclusão

Um SaaS CRM e ERP bem-sucedido deve ser robusto, escalável e fácil de usar, atendendo a uma ampla gama de necessidades empresariais de maneira integrada. A interface deve ser intuitiva e a funcionalidade deve abranger todas as áreas críticas de CRM e ERP, garantindo que as empresas possam gerenciar suas operações de forma eficiente e eficaz.

**Importante**

a estrudtura do meu projetoé com .evision.useraccess, com .evision.crm, com .evision.erp, partindo do useraccess como principal e eu quero tornar meu software escalavel , e tndendo as diferenças que virão no futuro com exigencias de cada modelo de negocio, como eu me preparo para isso?

ChatGPT

Para tornar seu software escalável e adaptável às futuras exigências de diferentes modelos de negócio, é essencial adotar uma abordagem modular e baseada em boas práticas de arquitetura de software. Aqui estão algumas recomendações para preparar seu projeto para escalabilidade e flexibilidade:

### 1. Arquitetura Modular

#### a. Microservices

Divida seu sistema em serviços menores e independentes, cada um responsável por uma funcionalidade específica. Isso facilita a manutenção, a escalabilidade e o desenvolvimento paralelo por diferentes equipes.

* **User Access Service (.evision.useraccess):** Gerenciamento de usuários, autenticação e autorização.
* **CRM Service (.evision.crm):** Gerenciamento de relacionamento com clientes.
* **ERP Service (.evision.erp):** Gestão de recursos empresariais.

#### b. Domain-Driven Design (DDD)

Adote o DDD para organizar seu código em domínios de negócios claramente definidos, cada um com seus próprios modelos e lógica de negócios.

### 2. API Gateway e Comunicação Inter-Serviços

#### a. API Gateway

Use um API Gateway para centralizar as chamadas de API e facilitar a gestão de rotas, autenticação e autorização.

#### b. Comunicação Assíncrona

Utilize filas de mensagens (como RabbitMQ ou Kafka) para comunicação assíncrona entre serviços. Isso melhora a resiliência e permite escalar serviços independentemente.

### 3. Padrões de Projeto

#### a. CQRS (Command Query Responsibility Segregation)

Separe operações de leitura e escrita para melhorar o desempenho e a escalabilidade.

#### b. Event Sourcing

Registre todas as alterações de estado como uma sequência de eventos, o que facilita a recuperação e a reconstituição de estados.

### 4. Infraestrutura

#### a. Containers

Use contêineres (Docker) para isolar os serviços e facilitar o desenvolvimento, testes e implantação.

#### b. Orquestração de Contêineres

Empregue ferramentas de orquestração como Kubernetes para gerenciar a implantação, escalabilidade e monitoramento dos contêineres.

### 5. Banco de Dados

#### a. Banco de Dados Distribuído

Considere bancos de dados distribuídos (como Cassandra ou CockroachDB) que oferecem alta disponibilidade e escalabilidade horizontal.

#### b. Multi-Tenant

Implemente uma arquitetura multi-tenant para suportar múltiplos clientes com isolamento adequado de dados.

### 6. Automação e DevOps

#### a. Integração Contínua/Entrega Contínua (CI/CD)

Automatize o processo de construção, testes e implantação usando ferramentas como Jenkins, GitLab CI ou GitHub Actions.

#### b. Infraestrutura como Código (IaC)

Gerencie sua infraestrutura com código usando ferramentas como Terraform ou AWS CloudFormation.

### 7. Monitoramento e Observabilidade

#### a. Logging Centralizado

Use ferramentas como ELK Stack (Elasticsearch, Logstash, Kibana) para centralizar e analisar logs.

#### b. Monitoramento

Implemente soluções de monitoramento como Prometheus e Grafana para métricas e alertas em tempo real.

### 8. Segurança

#### a. Autenticação e Autorização

Use padrões seguros como OAuth2 e OpenID Connect para autenticação e autorização.

#### b. Segurança de API

Proteja suas APIs com rate limiting, throttling e inspeção de tráfego.