



Instituto Federal da Paraíba – Campus Esperança

Curso Análise e Desenvolvimento de Sistemas

Data: 08/11/2024

Projeto Integrador

Professores: Artur Luiz Torres de Oliveira, Jaíndson Valentim Santana

Alunos: Alan Mateus, Lucas Rickelme, Pedro Henrique Silva da Costa

Gregório

DOCUMENTO DE VISÃO

1. Título

NBA Prediction DataBase

2. Objetivo

O propósito deste documento é expor as necessidades e funcionalidades gerais do sistema, definindo os requisitos de alto nível do em termos de necessidades dos usuários finais.

3. Escopo

Esse sistema permitirá que o usuário tenha total controle da disposição dos dados que ele colocar no sistema, com a finalidade de possuir uma base de dados para um outro software utilizar esses dados, através de uma API, disponibilizando todos os recursos necessários para tal.

4. Gestor

Representante: Artur Luiz Torres de Oliveira

Qualificação: Professor das disciplinas de Análise e Projeto de Sistemas e Sistemas Operacionais do IFPB - Campus Esperança.

Responsabilidades: Responsável por ensinar e orientar o projeto integrador.

Critérios de sucesso: API bem documentada, aplicando as metodologias de análise e projeto de sistema.

5. Posicionamento

5.1. Oportunidade de Negócios

Atualmente, o gestor responsável por realizar previsões de apostas utiliza informações coletadas na internet de forma fragmentada e ineficiente, devido à multiplicidade de fontes disponíveis. Esse cenário resulta em dificuldades no controle tanto da coleta quanto do gerenciamento de dados ao longo do tempo, levando a processos manuais que comprometem a produtividade e a precisão das análises.

Com o objetivo de resolver esses desafios, o NBA Prediction Database surge como uma solução para a centralização e organização dos dados coletados, oferecendo uma modelagem de dados robusta. A aplicação permitirá não apenas a inserção manual de informações, mas também a automação da extração de dados diretamente da internet, facilitando a atualização e a manutenção do banco de dados de forma a se integrar com um software de previsão de apostas. Dessa forma, o sistema proporcionará maior controle e administração das informações, melhorando a qualidade das previsões e otimizando o processo de tomada de decisão.

5.2. Descrição do Problema

O problema:	A falta de centralização e organização das informações afeta diretamente a eficiência na realização de previsões de apostas baseadas na NBA.
Afeta:	Gestores e analistas responsáveis por realizar previsões, bem como desenvolvedores que necessitam de dados confiáveis para alimentar softwares de previsão.
O seu impacto é:	Perda de tempo na coleta manual de dados, baixa produtividade, dificuldade em manter os dados atualizados, análises imprecisas e falhas na tomada de decisões

	devido à fragmentação das informações em múltiplas fontes.
Uma solução ideal seria:	A implementação do NBA Prediction Database, uma API que centraliza os dados em um banco organizado, oferece automação na extração de informações por meio de ETL e suporte para integrações futuras com softwares de predição de apostas, garantindo eficiência e precisão nas análises.

5.3. Descrições dos Envolvidos e Usuários

O público-alvo do NBA Prediction Database é composto por analistas, apostadores e entusiastas da NBA que buscam uma solução eficiente para centralizar, organizar e gerenciar dados da liga. Atualmente, esses usuários enfrentam desafios com a coleta fragmentada e a administração manual de informações provenientes de múltiplas fontes, o que compromete a precisão e a produtividade em suas análises.

Nome	Representa	Papel
Alan Mateus	Aluno do IFPB do curso superior de Análise e Desenvolvimento de Sistemas.	Idealiza, desenvolve e concretiza o sistema.
Lucas Rickelme	Aluno do IFPB do curso superior de Análise e Desenvolvimento de Sistemas.	Idealiza, desenvolve e concretiza o sistema.
Pedro Henrique Silva da Costa Gregório	Aluno do IFPB do curso superior de Análise e Desenvolvimento de	Idealiza, desenvolve e concretiza o sistema.

	Sistemas.	
Artur Luiz Torres de Oliveira	Analista e consumidor final do produto.	Realizar demandas e pedidos sobre o que ele pretende que seja implantado no produto final. Além de realizar o papel de homologar para saber se os requisitos foram devidamente cumpridos.
Artur Luiz Torres de Oliveira	Professor e orientador do projeto integrador.	Ensinar, analisar e ministrar as disciplinas de análise e projeto de sistemas.

5.4. Ambiente do Usuário

O NBA Prediction Database foi projetado para atender clientes como analistas, apostadores e entusiastas da NBA que utilizam softwares integrados para consultas e análises. No ambiente do cliente, o sistema será acessado indiretamente por meio de aplicações externas que se conectam à API, como plataformas de análise preditiva ou sistemas personalizados de gestão de dados.

Esses softwares integrados permitem que os usuários, de qualquer lugar com acesso à internet, consultem informações detalhadas sobre temporadas, equipes, jogadores e estatísticas. A experiência é facilitada pela interface dos sistemas que utilizam a API, garantindo rapidez e precisão nos dados fornecidos.

O cliente poderá, por exemplo, realizar análises preditivas e acompanhar tendências com base em informações atualizadas automaticamente pelo sistema de ETL integrado ao banco. Isso proporciona maior confiabilidade e conveniência para que os usuários finais foquem em suas estratégias e objetivos, sem preocupações com a coleta ou organização dos dados.

6. Alternativas e Concorrências

Alternativas Disponíveis:

Existem soluções comerciais amplamente utilizadas que podem ser consideradas como alternativas, como NBA Prediction DataBase, que oferece uma plataforma robusta. Outra alternativa é NBA ANALYTICS, que tem um modelo de licenciamento mais flexível, mas com recursos mais limitados no que se refere à integração com sistemas externos.

Concorrência:

Entre os principais concorrentes, a NBA ANALYTICS se destaca por oferecer um sistema completo com funcionalidades adicionais que podem ser atraentes para empresas de grande porte. No entanto, a solução deles apresenta um custo elevado e exige um tempo de adaptação maior.

7. Perspectiva do Produto

O NBA Prediction Database tem como principal objetivo proporcionar um gerenciamento eficiente e centralizado de dados relacionados à NBA, facilitando o acesso a informações de temporadas, equipes, jogadores, jogos e estatísticas detalhadas.

Ao oferecer uma API robusta e confiável, o produto permite que softwares integrados utilizem os dados para análises, previsões ou visualizações, reduzindo significativamente o tempo e o esforço necessários para a coleta e organização manual de informações dispersas.

Dessa forma, o NBA Prediction Database contribui para aumentar a produtividade dos usuários finais, garantindo que os dados estejam sempre atualizados, organizados e prontos para serem utilizados, otimizando suas atividades e resultados.

8. Necessidades e Funcionalidades do Produto

Nesta seção, descrevem-se as principais necessidades e funcionalidades requeridas pelas partes interessadas, relacionando a cada ator.

O sistema visa proporcionar telas específicas para os recursos disponíveis, desenvolvidas de modo didático e bastante visual, com o intuito de facilitar a navegação do usuário, além de proporcionar automatização do trabalho e economia de tempo. Veja uma breve descrição de alguns recursos:

Necessidade 1	Benefício
Controle de dados da temporada	Crítico

Id Func.	Descrição das funcionalidades, autores envolvidos e discriminação de dados que serão precisos.
F1.1	Listagem de dados da temporada
	<p>Usuário – Poderá consultar os dados da temporada através de requisições API via uma interface. A API retornará os dados da tabela "temporada" estruturados em formato JSON.</p> <p>Exemplo de resposta: {data:[{ id: 1, year: 2023}]}</p>
F1.2	Criação dos dados da temporada
	<p>Usuário – Poderá criar novos registros da temporada por meio de requisições API via uma interface. O usuário enviará um JSON como entrada, contendo os dados necessários para que a API insira essas informações na tabela "temporada".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador da Temporada • Ano da Temporada
F1.3	Atualização dos dados da temporada
	<p>Usuário – Poderá atualizar os dados da temporada utilizando requisições API via uma interface. O usuário enviará um JSON especificando o identificador do registro e as informações a serem modificadas para que a API realize a atualização na tabela "temporada".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador da Temporada • Ano da Temporada
F1.4	Deleção dos dados da temporada
	<p>Usuário – Poderá remover dados da temporada por meio de requisições API via uma interface. O usuário enviará um JSON especificando apenas o identificador do registro da temporada que</p>

	deseja excluir, permitindo que a API remova os dados correspondentes da tabela "temporada".
	Dados necessários: <ul style="list-style-type: none"> Identificador da Temporada

Necessidade 2		Benefício
Controle de dados dos jogadores		Crítico
Id Func.	Descrição das Funcionalidades/atores envolvidos	
F2.1	Listagem de dados dos jogadores	
	<p>Usuário – Poderá consultar os dados dos jogadores através de requisições API via uma interface. A API retornará os dados da tabela "jogadores" estruturados em formato JSON.</p> <p>Exemplo de resposta: {data:[{ id: 1, nome: "João", apelido: "Joca", numeroCamisa: 10, posicao: "Atacante", dataNascimento: "1990-05-12", peso: 75, altura: 1.80}]} }</p>	
F2.2	Criação dos dados dos jogadores	
	<p>Usuário – Poderá criar novos registros de jogadores por meio de requisições API via uma interface. O usuário enviará um JSON como entrada, contendo os dados necessários para que a API insira essas informações na tabela "jogadores".</p>	
	Dados necessários: <ul style="list-style-type: none"> Identificador Nome Apelido Data de Nascimento Número da Camisa Posição 	

	<ul style="list-style-type: none"> • Peso • Altura
F2.3	Atualização dos dados dos jogadores
	<p>Usuário – Poderá atualizar os dados dos jogadores utilizando requisições API via uma interface. O usuário enviará um JSON especificando o identificador do jogador e as informações a serem modificadas para que a API realize a atualização na tabela "jogadores".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador do Jogador • Nome • Apelido • Data de Nascimento • Número da Camisa • Posição • Peso • Altura
F2.4	Deleção dos dados dos jogadores
	<p>Usuário – Poderá remover dados dos jogadores por meio de requisições API via uma interface. O usuário enviará um JSON especificando apenas o identificador do jogador que deseja excluir, permitindo que a API remova os dados correspondentes da tabela "jogadores".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador

Necessidade 3		Benefício
Controle de dados das estatísticas dos jogadores		Crítico
Id	Descrição das Funcionalidades/atores envolvidos	

Func.	
F3.1	Listagem de estatísticas dos jogadores
	<p>Usuário – Poderá consultar as estatísticas dos jogadores através de requisições API via uma interface. A API retornará os dados da tabela "estatísticas_jogadores" estruturados em formato JSON.</p> <p>Exemplo de resposta: {data:[{ id: 1, pontos: 25, assistencias: 5, rebotes: 10, roubos: 2, bloqueios: 1, cestaTresPontos: 3, cestasDentroPerimetro: 8, lancesLivres: 6 }]}</p>
F3.2	Criação de estatísticas dos jogadores
	<p>Usuário – Poderá criar novos registros de estatísticas dos jogadores por meio de requisições API via uma interface. O usuário enviará um JSON como entrada, contendo os dados necessários para que a API insira essas informações na tabela "estatísticas_jogadores".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador • Pontos • Assistências • Rebotes • Roubos • Bloqueios • Cestas de Três Pontos • Cestas Dentro do Perímetro • Lances Livres
F3.3	Atualização das estatísticas dos jogadores
	<p>Usuário – Poderá atualizar as estatísticas dos jogadores utilizando requisições API via uma interface. O usuário enviará um JSON especificando o identificador do jogador e as informações a serem modificadas para que a API realize a atualização na tabela "estatísticas_jogadores".</p>

	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador • Pontos • Assistências • Rebotes • Roubos • Bloqueios • Cestas de Três Pontos • Cestas Dentro do Perímetro • Lances Livres
F3.4	Deleção de estatísticas dos jogadores
	<p>Usuário – Poderá remover registros de estatísticas dos jogadores por meio de requisições API via uma interface. O usuário enviará um JSON especificando apenas o identificador do jogador cujas estatísticas deseja excluir, permitindo que a API remova os dados correspondentes da tabela "estatísticas_jogadores".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador

Necessidade 4		Benefício
Controle de dados estatísticos dos Jogos		Crítico
Id Func.	Descrição das Funcionalidades/atores envolvidos	
F4.1	Listagem de jogos	
	<p>Usuário – Poderá consultar os dados dos jogos através de requisições API via uma interface. A API retornará os dados da tabela "jogos" estruturados em formato JSON.</p> <p>Exemplo de resposta: {data:[{ id: 1, confronto: "Time A vs Time</p>	

	B" }}}
F4.2	Criação de jogos
	Usuário – Poderá criar novos registros de jogos por meio de requisições API via uma interface. O usuário enviará um JSON como entrada, contendo os dados necessários para que a API insira essas informações na tabela "jogos".
	Dados necessários: <ul style="list-style-type: none"> • Confronto. • Identificador.
F4.3	Atualização de jogos
	Usuário – Poderá atualizar os dados dos jogos utilizando requisições API via uma interface. O usuário enviará um JSON especificando o identificador do jogo e as informações a serem modificadas para que a API realize a atualização na tabela "jogos".
	Dados necessários: <ul style="list-style-type: none"> • Identificador do Jogo; • Confronto.
F4.4	Deleção de jogos
	Usuário – Poderá remover registros de jogos por meio de requisições API via uma interface. O usuário enviará um JSON especificando apenas o identificador do jogo que deseja excluir, permitindo que a API remova os dados correspondentes da tabela "jogos".
	Dados necessários: <ul style="list-style-type: none"> • Identificador do Jogo

Necessidade 5	Benefício
---------------	-----------

Controle de dados dos times		Crítico
Id Func.	Descrição das funcionalidades, autores envolvidos e discriminação dos dados necessários	
F5.1	Listagem de times	
	<p>Usuário – Poderá consultar os dados dos times através de requisições API via uma interface. A API retornará os dados da tabela "times" estruturados em formato JSON.</p> <p>Exemplo de resposta: {data:[{ id: 1, nomeCompleto: "Clube Atlético", apelido: "Atlético", abreviacao: "ATL", anoFundado: 1920, cidade: "São Paulo", estado: "SP" }]}</p>	
F5.2	Criação de times	
	<p>Usuário – Poderá criar novos registros de times por meio de requisições API via uma interface. O usuário enviará um JSON como entrada, contendo os dados necessários para que a API insira essas informações na tabela "times".</p>	
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Nome Completo • Apelido • Abreviação • Ano Fundado • Cidade • Estado 	
F5.3	Atualização de times	
	<p>Usuário – Poderá atualizar os dados dos times utilizando requisições API via uma interface. O usuário enviará um JSON especificando o identificador do time e as informações a serem modificadas para que a API realize a atualização na tabela "times".</p>	

	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador do Time • Nome Completo • Apelido • Abreviação • Ano Fundado • Cidade • Estado
F5.4	Deleção de times
	<p>Usuário – Poderá remover registros de times por meio de requisições API via uma interface. O usuário enviará um JSON especificando apenas o identificador do time que deseja excluir, permitindo que a API remova os dados correspondentes da tabela "times".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador do Time

Necessidade 6		Benefício
Listagem de estatísticas dos times		Crítico
Id Func.	Descrição das funcionalidades, autores envolvidos e discriminação dos dados necessários	
F6.1	Listagem de estatísticas dos times	
	<p>Usuário – Poderá consultar as estatísticas dos times através de requisições API via uma interface. A API retornará os dados da tabela "estatísticas_times" estruturados em formato JSON.</p> <p>Exemplo de resposta: {data:[{ id: 1, roubos: 8, wl: "10-2", bloqueios: 3, lancesLivres: 12, rebotes: 40, pontos: 85, assistencias: 20, cestasDentroPerimetro: 25, cestasTresPontos: 8 }]}</p>	

F6.2	Criação de estatísticas dos times
	Usuário – Poderá criar novos registros de estatísticas dos times por meio de requisições API via uma interface. O usuário enviará um JSON como entrada, contendo os dados necessários para que a API insira essas informações na tabela "estatísticas_times".
	Dados necessários: <ul style="list-style-type: none"> • Roubos • WL (Win-Loss) • Bloqueios • Lances Livres • Rebotes • Pontos • Assistências • Cestas Dentro do Perímetro • Cestas de Três Pontos
F 6.3	Atualização de estatísticas dos times
	Usuário – Poderá atualizar as estatísticas dos times utilizando requisições API via uma interface. O usuário enviará um JSON especificando o identificador do time e as informações a serem modificadas para que a API realize a atualização na tabela "estatísticas_times".

	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador do Time • Roubos • WL (Win-Loss) • Bloqueios • Lances Livres • Rebotes • Pontos • Assistências • Cestas Dentro do Perímetro • Cestas de Três Pontos
F 6.4	Deleção de estatísticas dos times
	<p>Usuário – Poderá remover registros de estatísticas dos times por meio de requisições API via uma interface. O usuário enviará um JSON especificando apenas o identificador do time cujas estatísticas deseja excluir, permitindo que a API remova os dados correspondentes da tabela "estatísticas_times".</p>
	<p>Dados necessários:</p> <ul style="list-style-type: none"> • Identificador do Time

9. Licenciamento e Instalação

O licenciamento do sistema será fornecido conforme as condições estabelecidas na proposta, de acordo com o modelo de pagamento escolhido pelo cliente. O cliente receberá uma licença de uso não exclusiva, intransferível e válida durante todo o período de desenvolvimento, abrangendo todas as versões e atualizações lançadas até a conclusão do projeto.

9.1. Licenciamento:

A licença do sistema é concedida para uso interno da empresa, com o número de usuários definido na proposta inicial. Caso o cliente deseje ampliar o número de licenças ou alterar o tipo de uso, será necessário firmar um novo contrato de licenciamento. O cliente terá acesso a atualizações e correções do

sistema durante o período de suporte de 90 dias após a entrega. Após esse período, qualquer suporte será cobrado conforme acordado.

9.2. Instalação:

A instalação do sistema será realizada pela equipe de desenvolvimento na infraestrutura de TI do cliente, conforme os requisitos definidos. O processo inclui a configuração do ambiente, instalação dos arquivos do sistema e ajustes necessários para integração com sistemas existentes, quando aplicável. Durante a instalação, será fornecido suporte para garantir o funcionamento adequado e atender a todas as dependências de software e hardware. O cliente é responsável por garantir as condições adequadas para a instalação, incluindo servidores, acesso à rede e permissões necessárias.

10. Interligação com Outros Sistemas

O NBA Prediction Database já possui integração com um sistema automatizado de extração de dados baseado no processo ETL (Extract, Transform, Load). Esse sistema utiliza scripts em Python para consultar APIs, extrair informações relevantes, transformá-las e carregá-las diretamente no banco de dados, garantindo a atualização eficiente e consistente das informações.

Embora futuras integrações estejam planejadas, como a conexão com um software de previsões com inteligência artificial, estas serão implementadas em versões posteriores. A arquitetura do sistema, desenvolvida com alta coesão e baixo acoplamento, assegura que essas expansões possam ser realizadas de forma eficiente e escalável, mantendo a robustez e a confiabilidade do banco de dados.

11. Proposta de Solução Tecnológica Escolhida

O sistema será desenvolvido utilizando Java como linguagem de programação principal, com a capacidade de realizar conexão com uma base de dados focada em informações da NBA. O sistema permitirá o cadastro, modificação e exclusão de dados de apostas e usuários, utilizando a mesma base de dados do sistema.

O banco de dados escolhido para o projeto será o PostgreSQL, garantindo suporte eficiente às transações e consultas realizadas pelo sistema. O acesso à internet será essencial

para todas as funcionalidades, e o backup dos dados será realizado na nuvem, utilizando a conta de email cadastrada no sistema.

Além disso, o sistema terá a capacidade de se conectar a um sistema de ETL (Extract, Transform, Load), que permitirá a coleta automática de dados relacionados à NBA. Esse sistema de ETL será responsável por extrair os dados de fontes externas, transformá-los no formato adequado e carregá-los diretamente na base de dados do projeto. Com isso, o sistema garantirá que as informações, como estatísticas e resultados de jogos, sejam atualizadas de forma automática e em tempo real, oferecendo aos apostadores uma base de dados sempre atualizada. Neste projeto, foram implementadas duas abordagens para popular o banco de dados. A primeira utiliza um processo de ETL, no qual uma API é consultada por meio de scripts em Python para extrair, transformar e carregar os dados estruturados, inserindo-os adequadamente no banco de dados. A segunda abordagem oferece um controle mais direto e manual dos dados através de uma API com endpoints que permitem a comunicação direta com o banco de dados. Além disso, há uma interligação entre ambas as versões, permitindo que uma chamada de endpoint acione o processo de ETL, especificando uma temporada para a qual todos os dados relevantes serão inseridos no sistema de maneira automatizada e integrada.

Para o desenvolvimento, serão utilizadas diversas ferramentas, como VSCode, pgAdmin4, SpringBoot, DBeaver e Postman, para testar e gerenciar as funcionalidades do sistema. A integração com o sistema de ETL permitirá maior eficiência no gerenciamento de dados, reduzindo a necessidade de inserções manuais e minimizando erros, além de possibilitar que o sistema ofereça análises e insights mais rápidos, otimizando o processo de consulta e suporte para as decisões de apostas.

12. Valor da Proposta

A proposta para o desenvolvimento do sistema inclui os seguintes itens:

- Levantamento de Requisitos;
- Elaboração do Projeto do Sistema;
- Desenvolvimento e Codificação do Sistema;
- Testes e Validação do Sistema;
- Documentação Técnica Completa;
- Implantação do Sistema;
- Suporte, Correções e Atualizações por até 90 dias após a entrega.

O valor total desta proposta é de **R\$ 1.200,00 (Mil e duzentos reais)** mensais durante o período de desenvolvimento para cada desenvolvedor. O pagamento pode ser realizado de uma das seguintes formas:

a) Pagamento integral à vista:

1 x R\$ 1.200,00 no ato da aprovação da proposta.

b) Pagamento de entrada + contra-entrega:

1 x R\$ 600,00 no ato da aprovação da proposta.

1 x R\$ 600,00 (EM CHEQUE) com vencimento para 30 dias a partir da data de aprovação da proposta.

c) Pagamento de entrada + duas parcelas iguais:

1 x R\$700,00 no ato da aprovação.

2 x R\$ 250,00 (EM CHEQUES) com vencimento para 30 e 60 dias a partir da data de aprovação da proposta.

Obs.: O cliente poderá apresentar uma contraproposta para análise.

13. Cronograma de Execução

O cronograma abaixo exhibe uma estimativa do desenvolvimento do sistema.
Considerando a proposta aceita até o mês de Setembro de 2024.

ATIVIDADE / DATA	ABR.	MAIO.	AGO	SET.	OUT.	NOV.	DEZ.	JAN.
Análise de Requisitos			.					
Projeto do Sistema								
Codificação								
Testes e Validação								
Documentação								

Implantação								
Correções/Atualizações								
Defesa								

14. Termo de Aceitação

Eu, Artur Luiz Torres de Oliveira, analista de dados, certifico que estou ciente e de acordo com a proposta acima apresentada, tanto no que se refere às funcionalidades apresentadas no item 4 deste documento, quanto ao valor da proposta, presente no item 6, e com o cronograma de execução do item 7.

Assinatura do Cliente

Assinatura do Orientador

Assinatura dos Desenvolvedores

15. Restrições

15.1. Limitações Técnicas:

O banco de dados precisa lidar com o aumento do volume de dados ao longo das temporadas, o que pode impactar a performance em consultas mais complexas, exigindo otimizações contínuas para garantir eficiência. Ademais, a integração com fontes externas pode enfrentar problemas de consistência, demandando manutenção constante nas APIs de coleta de dados. O volume crescente de estatísticas dos jogadores, também, pode exigir alterações na estrutura do banco de dados para manter um bom desempenho.

15.2. Limitações Operacionais:

A falta de recursos especializados em banco de dados e integração de APIs pode ser um obstáculo, já que a equipe pode ter dificuldades em cumprir prazos ou entregar soluções eficientes. Também é necessário considerar os custos com infraestrutura, visto que à medida que o banco de dados cresce torna-se necessário mais armazenamento e poder de processamento. Outrossim, a atualização contínua dos dados, que exige

esforço constante para garantir que as informações estejam sempre corretas e atualizadas.

15.3. Limitações de Acesso à Dados:

O projeto depende de fontes externas para manter os dados atualizados, e qualquer mudança nas políticas de acesso ou nos custos desses dados pode afetar diretamente o andamento do projeto. Se houver alterações no fornecimento de dados, será necessário adaptar a coleta e o processo de integração.

16. Documentação

16.1. Ajuda on-line

Sistema de ajuda para auxiliar o usuário de forma a entender as funcionalidades da API e tirar dúvidas sobre determinados assuntos.