

**UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA**

TRABALHO FINAL DE ARQUITETURA DE SOFTWARE 2023-1

**IV/UFGInclui: sistema web
para processar o resultado do PS UFGInclui
Indígenas e Negros Quilombolas**

**GOIÂNIA - GO
2023**

**UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA**

TRABALHO FINAL DE ARQUITETURA DE SOFTWARE 2023-1

**IV/UFGInclui: sistema web
para processar o resultado do PS UFGInclui**

Professor Dr. Jacson Rodrigues Barbosa

Grupo 6

Amadeu Lee 202004743

Fillipe Mendonça Albuquerque 202107677

João Mário Fidelis Martins 202004759

Luca Santos Martins 202004765

GOIÂNIA - GO

2023

SUMÁRIO

PLANEJAMENTO.....	3
1. INTRODUÇÃO.....	4
1.1. Problemática.....	4
1.2. Domínio: Edital UFGInclui.....	5
1.3. Necessidades e Benefícios.....	8
1.4. Desafios.....	9
2. FLUXO DE PROCESSO PARA O SOFTWARE.....	12
2.1. Ações do candidatos.....	12
2.2. Ações do Instituto Verbena.....	12
2.3. Ações do software Módulo de Processamento Ágil (MPA).....	13
2.4. Atividades adicionais a partir do compartilhamento de dados do Instituto Verbena.....	13
2.5. Premissas e detalhes sobre o software.....	14
3. DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS.....	15
3.1. Histórias de Usuários.....	15
3.2. Requisitos Funcionais.....	21
3.3. Requisitos Não Funcionais.....	23
3.4. Restrições.....	25
3.5. Modelo de Casos de Uso (MCU).....	25
4. ARQUITETURA.....	26
4.1. Tradução dos Atributos de Qualidade.....	26
4.1.1. Adequação Funcional.....	27
4.1.2. Eficiência de Desempenho.....	27
4.1.3. Compatibilidade.....	28
4.1.4. Usabilidade.....	28
4.1.5. Confiabilidade.....	29
4.1.6. Segurança.....	29
4.1.7. Manutenibilidade.....	30
4.1.8. Portabilidade.....	30
4.2. Trade-off para o MPA.....	31
4.3. Estilos arquiteturais relevantes.....	32
4.3.1. Arquitetura de Camadas para o back-end.....	32
4.3.2. Arquitetura Híbrida para o front-end.....	33
4.4. Stakeholders, concerns e pontos de vista.....	35
4.5. Por que se usa arquitetura em camadas no back-end?.....	38
4.6. Por que se usa Flux e Clean Architecture no front-end?.....	38
4.6.1. Tecnologias escolhidas.....	40
5. MELHORIAS POSSÍVEIS.....	41
REFERÊNCIAS.....	42

PLANEJAMENTO

O trabalho final em Arquitetura de Software exige a dedicação adicional de 256 horas (com 64h/discente), distribuídas em:

Ano: 2023

19/04: Definição do grupo deste trabalho e discussões sobre a definição do escopo.

15/05: Decisão sobre o uso do MPA, com análise do projeto e critérios de desenvolvimento.

19/06: Apresentação da solução MPA e sobre o detalhamento da arquitetura de software aplicável.

21/06: Discussão sobre os atributos de qualidades utilizados para reforçar a decisão da arquitetura utilizada no back-end e no front-end.

28/06: Discussão sobre o processo de desenvolvimento do projeto e a correlação dos oito grupos de atributos de qualidade da ISO 25010:2015 com as decisões para o back-end e front-end. Surgiram dúvidas sobre a arquitetura global do software MPA e o registro de todos os componentes utilizáveis no cenário viável real e projeção futura (com API das plataformas INEP e IV).

30/06: Decisão sobre a abordagem do projeto e uso do estado atual para registro da arquitetura aplicável ao back-end (camadas) e front-end (Flux híbrido).

07/07: Análise dos atributos de qualidade segundo a ISO 25010:2011, com tradução e seleção dos itens mais relevantes para o MPA.

21/07: Tradução e detalhamento de cada atributo de qualidade para revisão do que de fato será incorporado no MVP do MPA.

28/07: Definição de apenas dois atributos de qualidade na visão do back-end e dois na visão do front-end. Segurança é identificado como um atributos de qualidade transversal.

04/08: Análise dos modelos e pontos de vistas dos stakeholders do projeto.

Em todas as datas houve desenvolvimento e refatoração de código.

Custo estimado

Caso este trabalho utilize como base um valor subsidiado de R\$25,00/h a entrega final implicaria em um custo total de R\$6400,00. No mercado é comum encontrar valores brasileiros que variam de R\$100 a R\$500 por hora técnica.

1. INTRODUÇÃO

O trabalho final de Arquitetura de Software enfatiza o processo de definição da arquitetura de software em seus vários aspectos para um contexto definido pelo grupo. Para este trabalho, no intuito de avançar no desenvolvimento do software Módulo de Processamento Ágil (MPA), o grupo utilizou como base o domínio do edital UFGInclui: Indígenas Brasileiros e Negros Quilombolas.

A partir da análise do domínio, são apresentados conceitos sobre os atributos de qualidade, conforme ISO 25010:2011, e a correlação dos aspectos críticos referentes ao desenvolvimento do back-end e do front-end do MPA. Este processo aborda o desenvolvimento do MPA no estado atual, ou seja, sem considerar como componentes as opções de uso de API das entidades Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) e Instituto Verbena (IV).

Assim, este capítulo apresenta o relato da problemática, com identificação de necessidades e benefícios desejados com a implantação da solução do software de acordo com a perspectiva do cliente-usuário principal. Parte do relato inclui os desafios relevantes para o desenvolvimento do sistema e os riscos preliminares.

O domínio deste trabalho é um sistema web para processar o resultado do Processo Seletivo da chamada UFGInclui, do Instituto Verbena na Universidade Federal de Goiás, a partir dos requisitos listados no Edital 027/2022. Este documento contém 36 páginas e define as regras e processos necessários para o ingresso na UFG dos grupos ali indicados. A definição do domínio limita-se ao sistema capaz de processar os resultados de processos seletivos dessa natureza, sem considerar, no recorte atual do escopo, o uso de comunicação das plataformas INEP e IV, conforme citado anteriormente.

1.1. Problemática

A necessidade de desenvolver a solução deste trabalho parte da percepção de ineficiências no processamento do resultado dos Processos Seletivos UFGInclui na Universidade Federal de Goiás (UFG). A universidade identificou a necessidade de desenvolver um sistema web para processar o resultado do Processo Seletivo UFGInclui: Indígenas Brasileiros e Negros Quilombolas.

Esta versão do Programa UFGInclui determina as regras de ingresso de candidatos indígenas brasileiro e negros quilombolas para o primeiro ou o segundo semestre dos cursos superiores de graduação presenciais listados na UFG. Para

concorrer, os candidatos devem ter cursado o Ensino Médio integralmente em escola pública e realizado o Exame Nacional do Ensino Médio (ENEM). A nota do ENEM é um dos critérios de seleção e, **particularmente no curso de Música – Licenciatura com habilitação em Educação Musical**, há necessidade da realização da Verificação de Habilidades e Conhecimento Específicos (VHCE). Potencialmente poderá haver aplicação de VHCE em outros cursos de acordo com a decisão do IV.

Se houver interesse de indígenas brasileiros e negros quilombolas nos cursos aptos, a UFG abre vagas extras para os aprovados. As vagas extras criadas nos cursos listados da UFG exigem a inscrição de candidatos indígenas brasileiro (1 vaga por semestre, ou seja, 1 vaga no semestre 1 e 1 vaga no semestre 2 do ano do processo seletivo) e negros quilombolas (1 vaga por semestre, ou seja, 1 vaga no semestre 1 e 1 vaga no semestre 2 do ano do processo seletivo) por curso presencial nos campus das regiões contempladas, como Goiânia, Aparecida de Goiânia e Cidade de Goiás. Para concorrer às vagas o candidato deve seguir o regimento descrito no edital.

1.2. Domínio: Edital UFGInclui

A equipe avaliou os requisitos descritos no Edital 027/2022 e retirou os itens considerados fora do escopo deste trabalho em comum acordo com o professor Dr. Jacson Rodrigues. Os itens tachados não são escopo deste trabalho:

- ~~Validar candidato ser indígena ou negro quilombola;~~
- ~~Validar ter cursado Ensino Médio integralmente em escola pública (validar conclusão do 1º, 2º, 3º e até 4º ano concluído em escola pública);~~
- Validar nota do ENEM;
- ~~Validar inscrição de candidato indígena e negro quilombola nos cursos da UFG para abertura de vagas extras dos cursos aptos, com respectiva etnia existente no território brasileiro e especificar a comunidade a qual pertence;~~
- ~~Validar os cursos aptos no Programa UFGInclui: Administração (bacharelado — noturno), Agronomia (bacharelado — integral), Ciência da Computação (bacharelado — integral), Educação Física (grau não definido — integral), Engenharia Civil (bacharelado — integral), Engenharia da Computação (bacharelado — integral), Engenharia Elétrica (bacharelado — integral), Farmácia (bacharelado — integral), Medicina Veterinária (bacharelado —~~

~~integral), Música – Licenciatura com habilitação em Educação Musical, Nutrição (bacharelado – matutino), Química (bacharelado – integral) e Química (licenciatura – integral);~~

- Validar a aprovação na Verificação de Habilidades e Conhecimento Específicos (VHCE) para o curso de Música – Licenciatura com habilitação em Educação Musical;
- ~~Validar inscrição pelo Instituto Verbena (www.institutoverbena.ufg.br);~~
- ~~Validar documentos de identificação;~~
- Validar grupo do curso escolhido na UFG e validar a comparação entre notas dos candidatos, com os respectivos pesos relativos para L: Linguagens, Códigos e suas Tecnologias; CH: Ciências Humanas e suas Tecnologias; CN: Ciências da Natureza e suas Tecnologias; M: Matemática e suas Tecnologias; R: Redação. A maior nota por curso ingressa no primeiro semestre, e a segunda maior nota no segundo semestre. A maior idade é o critério de desempate;
- Validar nota da prova de VHCE;
- Validar interposição de recursos a cada resultado (inscrições homologadas; análise de documentação; relação de candidatos convocados para VHCE, resultado preliminar VHCE, resultado preliminar ENEM, resultado preliminar do processo seletivo);
- Validar os casos omissos decididos pela CGA/UFG e Instituto Verbena/UFG;
- Validar o cronograma publicado para cada edital do processo seletivo UFGInclui;

O domínio do software abordado neste trabalho envolve os interessados em ingressar na UFG através dos editais de inclusão denominados UFGInclui. Trata-se de uma solução que melhora a integração das sociedades de indígenas do Brasil e de negros quilombolas à formação superior, como oferecido pela UFG, além de permitir a análise e ciência de dados do contexto desses candidatos a partir da digitalização do processo seletivo.

O software pode ser utilizado pela equipe técnica do Instituto Verbena, responsável pelo processo seletivo do respectivo edital. Todas as publicações e comunicados com os candidatos, no entanto, continuam sendo de responsabilidade

do Instituto Verbena, sem haver necessidade de alteração do processo atual de divulgação.

Assim, o software será acessado pela web, como um serviço disponível por uma URL. O software deve possuir interface simples, objetiva e amigável, back-end (API) contemplando as regras de negócio do edital e os *end points* relevantes, e com a persistência de dados necessária para se obter o processamento ágil almejado.

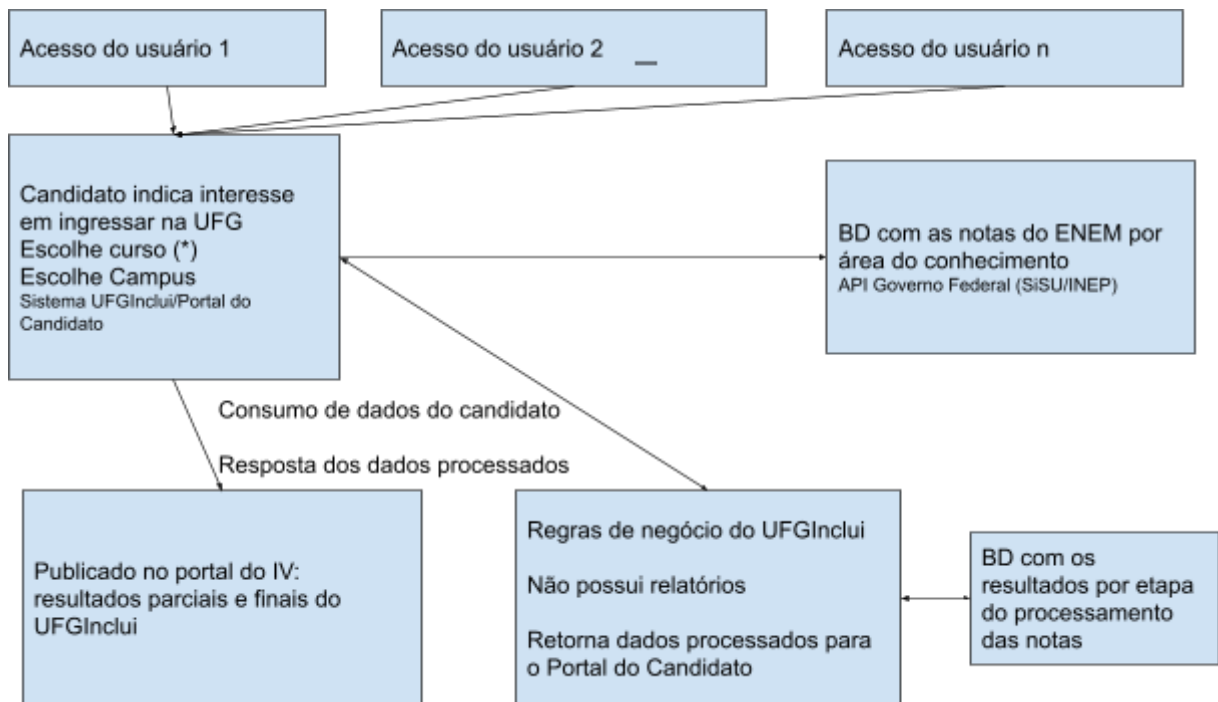
A utilização da solução web exige a contratação de serviços servidores, persistência de dados e segurança necessária para o gerenciamento de dados e suas respectivas comunicações. A partir das análises de requisitos da solução é possível avaliar os estilos arquiteturais aplicáveis ao projeto. A escolha da arquitetura tem relação direto com os atributos de qualidade, sustentabilidade, e os critérios críticos de segurança e performance do software. É na arquitetura que se especifica as estruturas de dados e a integração entre os componentes utilizados no software.

São exemplos de arquiteturas de software: Em camadas; Cliente-servidor; Mestre-escravo; Filtro de Tubo; Broker; P2P (peer-to-peer); Barramento de evento; MVC (model-view-controller); Quadro Negro; Interpretador.

Para atender o escopo, este trabalho está estruturado em:

1. Entendimento do problema, objetivo e componentes;
2. Identificação dos usuários;
3. Requisitos Funcionais e Não Funcionais;
4. Definição dos atributos de qualidade e pontos críticos relevantes ao projeto;
5. Construção do MVP do software;
6. Entregas e evidências no GitHub.

Figura 1 - Arranjo preliminar sobre o processo do software.



Fonte: próprios autores.

1.3. Necessidades e Benefícios

O sistema web a ser desenvolvido para processar o resultado do Processo Seletivo UFGInclui: Indígenas Brasileiros e Negros Quilombolas possui como entrada:

- Dados de cadastro dos candidatos;
- Dados de notas do ENEM dos candidatos;
- Dados de notas de VHCE dos candidatos que fazem esta etapa.

Com esses dados o sistema permitirá:

- Acelerar o processamento de informações dos candidatos para arquivo compatível ao modelo INEP para emissão de notas do ENEM;
- Acelerar o processamento das notas do ENEM;
- Acelerar o processamento das notas de VHCE para os candidatos que participam dessa etapa;
- Acelerar a divulgação de resultados do processo seletivo;
- Ampliar a produção de dados estatísticos sobre o Programa UFGInclui.

Entretanto, existem desafios importantes que precisam ser considerados para o desenvolvimento do software, como:

- O acesso ao sistema do Instituto Verbena;
- O acesso aos dados no Portal do Candidato;
- A integração desses dados com a plataforma do INEP;
- A escolha das tecnologias mais adequadas para o processamento em sistema web, para os dados e a definição das saídas esperadas pelo sistema.

O sistema busca trazer uma grande digitalização do processamento dos dados do Programa UFGInclui, com celeridade, eficiência e capacidade de integração de dados e sistemas. Além disso, com o uso do software, busca-se elevar a agilidade sobre a evolução de processos e trâmites rotineiros sobre o contexto do programa, como, por exemplo, as rotinas de cadastramento de um usuário no programa, levantamento de cursos e vagas disponíveis, registro e recuperação de informações passadas, geração de estatísticas etc.

É importante ressaltar que a segurança do processo seletivo como um todo aumenta com o uso do software, caso as boas práticas e tecnologias de segurança sejam bem planejadas e implementadas no software, com impacto direto na credibilidade, visibilidade e transparência do processo seletivo do Programa UFGInclui.

1.4. Desafios

São desafios do desenvolvimento deste software e da definição das arquiteturas de software:

1. O entendimento do regimento descrito no edital;
2. Compreensão do processo de acesso às notas do ENEM dos candidatos do UFGInclui feita pelo Instituto Verbena;
3. O entendimento do formato dos dados de cadastro para solicitação das notas do ENEM;
4. O entendimento do formato dos dados devolvidos pela INEP;
5. A definição dos atributos de qualidade críticos para o desenvolvimento do software, documentado na arquitetura de software para o back-end e o front-end;

6. A definição das tecnologias que serão utilizadas para desenvolvimento do front-end, back-end e persistência de dados do sistema web;
7. A definição das saídas do sistema, para a lista de candidatos aptos a realizar o VHCE, de candidatos aprovados por curso e campus, e detalhamento das notas de todos os candidatos para ser publicada no Portal do Candidato do Instituto Verbená;
8. A definição da segurança adequada para o processamento dos dados dos candidatos.

Sabe-se que o processo de entrada em Instituições de Ensino Superior (IES) públicas é realizado através do INEP, pelo endereço <<https://www.gov.br/inep/pt-br>>. Lá um candidato que realizou o ENEM consegue utilizar o login do acesso único gov.br no endereço <<https://acessounico.mec.gov.br/>>, selecionar duas IES (primária e secundária) e os cursos que deseja cursar. Com a nota do ENEM existem chamadas dos aprovados além da primeira, desde que existam vagas remanescentes na IES.

O Portal do Candidato do Instituto Verbená na Universidade Federal de Goiás exige um cadastro completo do candidato independente do INEP, ou seja, não há integração das informações e não há disponível uma API para acesso às informações fornecidas pelo INEP sobre o ENEM. Assim, para contornar essa limitação, é necessário o processamento dos dados dos candidatos no Instituto Verbená e persistência no formato aceito pela INEP. Tal arquivo deve ser utilizado para se ter acesso às notas do ENEM.

Também integram o regimento do Programa UFGInclui as regras específicas dos cursos superiores, como:

1. Nos cursos que dependem do VHCE, a nota obtida no teste é adicionada a nota comum do ENEM do candidato;
2. Toda comunicação com o candidato, como a publicação das homologações, de resultados parciais, recursos e resultados finais são de responsabilidade do Instituto Verbená e não do software a ser desenvolvido neste trabalho;
3. As entradas para o software deste trabalho serão em formato .csv ou .ods, e ocorrerão em três momentos: cadastros dos candidatos no Portal do Candidato (filtrados para o edital UFGInclui indígenas e negros quilombolas); notas do ENEM dos candidatos; notas de VHCE dos candidatos.

4. As saídas do software é a lista ordenada de candidatos aprovados, entre indígenas brasileiros e negros quilombolas, com entrada no primeiro e no segundo semestre do ano do processo seletivo, indicando o município, o campus, o curso, o nome completo do aprovado e o número de inscrição daquele candidato no IV.

2. FLUXO DE PROCESSO PARA O SOFTWARE

Este capítulo apresenta o entendimento do fluxo de processo que o software busca apoiar. Ele considera atividades externas ao escopo, como responsabilidades dos candidatos e do Instituto Verbena. Por fim, são listadas as etapas necessárias para o desenvolvimento da aplicação que servirão como requisitos para a definição das funcionalidades e desenvolvimento do sistema.

Figura 2 - Login de acesso ao Portal do Candidato do Instituto Verbena.

A imagem mostra a interface de login do Portal do Candidato do Instituto Verbena. No topo, há o logo do Instituto Verbena com o texto "INSTITUTO VERBENA" e "SELEÇÃO - FORMAÇÃO - AVALIAÇÃO - PESQUISA" abaixo dele. À direita do logo, está o texto "Portal do(a) Candidato(a)". Abaixo disso, há um formulário de login com o título "Entrar" em um botão azul. O formulário contém dois campos de entrada: "CPF" e "Senha". Abaixo dos campos, há um botão azul "Acessar" e um link "Esqueci a senha". Na base do formulário, há um link "Não tenho cadastro".

Fonte: Instituto Verbena.

2.1. Ações do candidatos

O candidato deve:

INÍCIO

1. Candidato fez login
2. Candidato selecionou edital
3. Candidato confirmou inscrição

FIM

2.2. Ações do Instituto Verbena

Os responsáveis pela operacionalização do processo seletivo no Instituto Verbena devem:

INÍCIO

1. Instituto encerra inscrições
2. Instituto homologa e publica inscrições preliminares

3. Instituto permite e resolve recursos
4. Instituto publica inscrições finais
5. Instituto consome serviço MPA

FIM

2.3. Ações do software Módulo de Processamento Ágil (MPA)

O software deve:

INÍCIO

1. MPA recebe arquivo .csv do Instituto Verbena com os dados cadastrais;
2. MPA filtra os dados exigidos pelo INEP e gera um arquivo .txt para o Instituto Verbena solicitar as notas do ENEM;
3. MPA recebe arquivo .csv do INEP com cpf e notas ENEM (mock);
4. (Específico para o trabalho) equipe espelha esquema do banco de dados de candidatos de acordo com o arquivo .csv recebido;
5. MPA analisa necessidade de VHCE;
6. Se sim: MPA retorna lista de candidatos aptos a fazer VHCE por candidato por curso por campus;
 - a. MPA recebe notas de VHCE (detalhada) por candidato por curso por campus → formato do arquivo .csv;
 - b. MPA analisa regras de VHCE (>0 nos critérios e >100 no total);
 - c. MPA retorna para o Instituto Verbena as notas VHCE analisadas.
7. **MPA analisa notas ENEM conforme pesos descritos no quadro 2 por curso por campus;**
8. MPA realiza cálculo da nota final (ENEM + VHCE);
9. MPA ordena candidatos por nota final, considerando os critérios de desempate (idade);
10. MPA retorna candidato e nota final de todos (ENEM + VHCE), indicando aprovados.

FIM

2.4. Atividades adicionais a partir do compartilhamento de dados do Instituto Verbena

1. Definir o design e arquitetura do projeto. Analisar as planilhas .ods e .txt compartilhadas pelo professor Jacson;

2.5. Premissas e detalhes sobre o software

1. O ideal é ter acesso privado à URL do software, porém isso não é escopo deste trabalho;
2. Utilizar as tecnologias JavaScript, Node JS (<https://nodejs.org/en/>) e React (<https://pt-br.reactjs.org/>);
3. Os arquivos .csv devem indicar o edital que ele se aplica para facilitar o processamento de dados das notas do ENEM e VHCE de um mesmo edital em momentos distintos;
4. O front-end conta com uma interface simples, uma página web, com a validação do processamento do fluxo conforme o edital;
 - a. Hipótese: é possível o front-end devolver um código de autorização para que o usuário possa processar a próxima etapa do processo seletivo no software;
 - b. Typescript (TS), React;
5. Back-end: Node JS, Typescript;
6. Database: Postgress (banco de dados relacional), Prisma (Node.JS ORM);
7. O Instituto Verbena irá compartilhar o arquivo .csv com as informações de dados cadastrais: candidatos, campus, cursos e vagas a partir de um banco de dados postgres. Isso permite utilizar os mesmos nome *database schema*;
8. A partir do arquivo .csv dos dados cadastrais o software deve filtrar os dados relevantes exigidos pelo INEP e gerar como saída um arquivo .txt com:
 - a. A lista de CPF dos candidatos;
 - b. A lista de notas do ENEM dos candidatos;
 - c. O número de inscrição do ENEM dos candidatos.
9. O INEP gera um arquivo .csv com todas as notas do ENEM dos candidatos;
10. O Instituto Verbena alimenta o .csv para o software iniciar o processamento dos dados do processo seletivo UFGInclui indígenas e negros quilombolas.

1º arquivo .csv entrada = dados cadastrais do Instituto Verbena;
2º arquivo .csv entrada = notas ENEM geradas pelo INEP;
3º arquivo .csv entrada = notas VHCE.

3. DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS

Este capítulo apresenta o mapeamento de requisitos, comumente separados em requisitos funcionais (RF), requisitos não funcionais (RNF) e ocasionalmente restrições (RE). Os requisitos são mapeados e descritos de forma direta e que reduza o risco de ambiguidade. Aqui também é apresentada uma tabela dos requisitos categorizados, com o respectivo identificador alfanumérico, o assunto, a iteração corrente, a fonte do requisito e a data da última atualização. O artefato Modelo de Casos de Uso (MCU) é apresentado, com as respectivas relações de *include* e *extends*, e os atores relevantes.

O método de Histórias de Usuários (HU) é amplamente utilizado no mapeamento de requisitos, apresentando a ótica do usuário do software e não do processamento de dados realizado pelo sistema. A sua aplicação utiliza um responsável pela HU, a prioridade, uma identificação alfanumérica para cada requisito e algumas combinações de palavras-chaves COMO, DESEJO, PARA QUE ou COMO, QUERO, PARA. Posteriormente é comum adicionar a descrição dos Testes de Aceitação (TA) para cada história de usuário, também com identificação alfanumérica, cenários bons e ruins, e com as palavras-chaves DADO, QUANDO, ENTÃO.

A estrutura deste capítulo se divide em:

Parte 3.1 - Histórias de Usuário (HU);

Parte 3.2 - Requisitos Funcionais (RF);

Parte 3.3 - Requisitos Não Funcionais (RNF);

Parte 3.4 - Restrições (RE);

Parte 3.5 - Modelo de Casos de Uso (MCU).

3.1. Histórias de Usuários

HU 1.0 MPA recebe os dados cadastrais dos candidatos do Instituto Verbena

Responsável: Fillipe

Prioridade: Baixa

Como responsável pelo processo seletivo;

Quero fazer o upload do arquivo .csv com os dados cadastrais dos candidatos do edital UFGInclui, com todas as informações relevantes ao processo seletivo, como nome completo, data de nascimento, CPF, campus e curso;

Para persistir esses dados no MPA **E PARA** acelerar o processamento de dados do processo seletivo **E PARA** que seja realizado o filtro desses dados conforme solicitado pelo INEP para consulta às notas do ENEM.

TA 1.0 Upload do arquivo .csv

TA 1.1 Cenário bom: Upload do arquivo realizado com sucesso

Dado que estou na página web do MPA.

Quando clico no botão <carregar arquivo .csv de dados cadastrais> **E** procuro o respectivo arquivo .csv de dados cadastrais do edital que quero processar os dados no armazenamento local do computador que estou usando **E** clico no botão <carregar>;

Então o software MPA indica <os dados foram carregados com sucesso> **E** informa na tela o <código de autorização para a próxima etapa>.

Tags dados cadastrais IV.

HU 2.0 MPA filtra dados para plataforma INEP

Responsável: Fillipe

Prioridade: Baixa

Como responsável pelo processo seletivo;

Quero fazer o download do arquivo .txt com o formato esperado pelo INEP para consulta de notas do ENEM;

Para consultar as notas do ENEM dos candidatos do processo seletivo UFGInclui.

TA 2.0 Download do arquivo .txt

TA 2.1 Cenário bom: Arquivo .txt baixado com sucesso

Dado que fiz o upload do arquivo .csv de dados cadastrais com sucesso;

Quando clico no botão <carregar> **E** recebo a mensagem <os dados foram carregados com sucesso> **E** a informação na tela do <código de autorização para a próxima etapa>;

Então o MPA permite eu fazer o download do arquivo .txt no computador que estou usando, armazenamento local, com os respectivos filtros esperados pelo INEP para consulta de notas do ENEM.

Tags dados INEP; notas ENEM.

HU 3.0 MPA recebe arquivo .csv do INEP

Responsável: João Mário

Prioridade: Baixa

Como responsável pelo processo seletivo

Quero fazer o upload de um arquivo do tipo .csv contendo as notas e os cpfs dos candidatos do edital

Para que o serviço possa ter acesso a tais informações que são pré-requisitos para o funcionamento dos demais processos

TA 3.0 Upload do arquivo .csv

TA 3.1 Cenário bom: O upload do arquivo foi feito com sucesso

Dado que estou na tela principal do sistema **E** selecionei os arquivos que desejo enviar

Quando eu clicar o botão <fazer upload de arquivos>

Então os arquivos devem ser enviados e uma mensagem deve ser exibida para o usuário com a mensagem “os arquivos foram enviados com sucesso”

Tags arquivo .csv

~~**HU 4.0 (Específico para o trabalho) equipe espelha esquema do banco de dados de candidatos de acordo com o arquivo .csv recebido**~~

~~**Responsável:** João Mário~~

~~**Prioridade:** Baixa~~

~~**Como** desenvolvedor~~

~~**Quero** utilizar o arquivo .csv fornecido~~

~~**Para** popular o banco de dados de dados que será utilizado pelo serviço~~

~~**TA 4.0** espelhamento dos dados no banco de dados~~

~~**TA 4.1 Cenário bom:** todos os dados que estavam no arquivo .csv fornecido agora estão corretamente refletidos no banco de dados de serviço~~

~~**Dado** que possui o arquivo .csv que contém os dados que deverão ser refletidos no banco de dados~~

~~**Quando** for executado o comando `psql \copy` na cli do banco de dados (mais informações sobre essa ação podem ser encontradas neste artigo)~~

~~**Então** os dados do arquivo .csv deverão estar devidamente refletidos no banco de dados do serviço~~

~~**Tags** arquivo .csv~~

HU 5.0 Análise da necessidade de VHCE

Revisão: 2

Responsável: João Mário

Prioridade: Baixa

Como responsável pelo processo seletivo

Quero receber a lista de candidatos que precisam realizar o VHCE

Para poder aplicar o VHCE

TA 5.0 verificação da obrigatoriedade de VHCE para determinado curso

TA 5.1 Cenário bom: o serviço identifica se o curso requer VHCE ou não

Dado que foi realizado o upload das notas do ENEM

Quando eu clico em <carregar>

Então o sistema indica que a lista de candidatos aptos a realizar o VHCE está disponível para download **E** permite o download da lista.

Tags dados do curso analisado

HU 6.0 Envio dos Resultados do VHCE

Responsável: Luca

Prioridade: Baixa

Como responsável pelo processo seletivo

Quero enviar para o MPA a lista de resultados da aplicação do VHCE

Para informar ao sistema as notas dos participantes no teste, o que é necessário para aqueles estudantes nos quais a aplicação do VHCE é obrigatória.

TA 6.1 Cenário bom: Lista enviada com sucesso

Dado que o MPA aguarda retorno, por parte do IV, com o detalhamento das notas do VHCE, por candidato, curso e campus (arquivo CSV)

Quando MPA recebe arquivo CSV

Então MPA armazena notas VHCE dos candidatos **E** realiza uma análise de acordo com as regras definidas no edital (>0 nos critérios e >100 no total) **E** retorna para o Instituto Verbena a análise

Tags notas VHCE

HU 6.1 Resultados da análise inicial das notas do VHCE

Responsável: Luca

Prioridade: Baixa

Como responsável pelo processo seletivo

Quero receber uma lista com os resultados da análise inicial das notas VHCE dos participantes que foram requeridos a aplicação do teste

Para já saber, de antemão, os candidatos que não alcançaram a pontuação mínima, estando assim, já eliminados do processo seletivo.

TA 6.1 Cenário bom: Lista enviada com sucesso

Dado que o IV aguarda retorno, por parte do MPA, com o detalhamento da análise das notas VHCE (arquivo CSV)

Quando MPA possui notas armazenadas

Então MPA realiza uma análise de acordo com as regras definidas no edital (>0 nos critérios e >100 no total) **E** retorna para o Instituto Verbena a análise

Tags notas VHCE

HU 7.0 Cálculo de resultados do edital

Responsável: Marcos

Prioridade: Baixa

Como responsável pelo processo seletivo

Quero definir quais candidatos foram aprovados no edital selecionado

Para que a geração do arquivo seja feita

TA 7.0.1 - Cálculo dos resultados do edital

TA 7.0.2 Cenário bom: a lista de candidatos aprovados gerada no HU 7.1 é correta

Dado que as notas do Enem e do VHCE estejam armazenadas para o edital selecionado

Quando o usuário pressionar o botão “Gerar”

Então o cálculo de quais candidatos foram aprovados é gerado, de acordo com regras de quantidade de vagas por curso

Tags nome do edital, nome do câmpus, curso, nomes dos candidatos aprovados e pontuações

HU 7.1 Geração do arquivo de resultados do edital

Responsável: Marcos

Prioridade: Baixa

Como responsável pelo processo seletivo

Quero gerar a lista de alunos aprovados em um edital

Para que a divulgação no site do Instituto Verbena seja realizada

TA 7.1.1 - Geração do arquivo PDF contendo os resultados

TA 7.1.2 Cenário bom: a geração do arquivo é bem-sucedida, e mostra somente os candidatos aprovados no edital selecionado

Dado que as notas do Enem e do VHCE estejam armazenadas para o edital selecionado

Quando o usuário pressionar o botão “Gerar”

Então o arquivo PDF contendo os resultados calculados no HU 7.0 é gerado

Tags nome do edital, nome do câmpus, curso, nomes dos candidatos aprovados e pontuações

Tabela 1 - Necessidades gerais.

Identificador	Assunto	Iteração	Fonte do requisito	Última atualização
HU1.0	MPA recebe os dados cadastrais dos candidatos do Instituto Verbena	1	Instituto Verbena	15/12/2022
HU2.0	MPA filtra dados para plataforma INEP	1	INEP	15/12/2022
HU3.0	MPA recebe arquivo .csv do INEP	1	INEP	21/12/2022
HU5.0	Análise da necessidade de VHCE	1	Instituto Verbena	21/12/2022
HU6.0	Envio dos Resultados do VHCE para o MPA	1	Instituto Verbena	19/12/2022
HU6.1	Envio dos resultados da análise inicial das notas do VHCE para o IV.	1	Instituto Verbena	19/12/2022
HU 7.0	Cálculo dos resultados finais	1	INEP, Instituto Verbena	21/12/2022
HU 7.1	Geração de arquivo PDF contendo os resultados finais	1	INEP, Instituto Verbena	21/12/2022

Fonte: próprios autores.

3.2. Requisitos Funcionais

RF 5.0 Análise da necessidade de VHCE

Responsável: João Mário

Prioridade: Baixa

Para atender a HU: 5.0

O sistema MPA analisa a obrigatoriedade da aplicação de VHCE no curso que exige essa etapa, conforme as regras do edital. O sistema MPA retorna uma lista de candidatos aptos a realizar o VHCE e disponibiliza a lista para download.

RF 6.0 Recebimento das notas do VHCE

Responsável: Luca S. Martins

Prioridade: Baixa

Para atender a HU: 6.0

O sistema deve ser capaz de receber, através da comunicação com o Instituto Verbena (IV), um documento do formato CSV contendo as notas dos participantes que são requeridos a realização do teste.

RF 6.1 Envio da análise das notas do VHCE

Responsável: Luca S. Martins

Prioridade: Baixa

Para atender a HU: 6.1

O sistema deve ser capaz de enviar ao IV um documento contendo a análise das notas dos candidatos que realizaram o VHCE, indicando se a candidatura vai ser mantida em progresso ou não.

RF 7.0 Cálculo da nota do ENEM do participante

Responsável: Luca S. Martins

Prioridade: Baixa

Para atender a HU: 7.0

O sistema deve ser capaz de calcular a nota ponderada que o participante obteve no ENEM, levando em consideração os pesos relativos ao curso e campus escolhidos.

RF 7.1 Cálculo de resultados do edital

Responsável: Marcos

Prioridade: Baixa

Para atender a HU: 7.1

O sistema deve ser capaz de calcular corretamente os resultados do edital, definindo de forma correta quais candidatos foram aprovados.

RF 7.2 Geração do arquivo de resultados do edital

Responsável: Marcos

Prioridade: Baixa

Para atender a HU: 7.2

O sistema deve ser capaz de gerar um arquivo PDF contendo as informações sobre os aprovados no edital, utilizando os dados gerados no HU 7.0 para tal.

Tabela 2 - Requisitos Funcionais

Identificador	Assunto	Iteração	Fonte do requisito	Última atualização
RF 5.0	Necessidade de VHCE	1	Instituto Verbena	21/12/2022
RF 6.0	Notas VHCE	1	Instituto Verbena	21/12/2022
RF 6.1	Análise notas VHCE	1	Instituto Verbena	21/12/2022
RF 7.0	Cálculo nota ENEM	1	Instituto Verbena	21/12/2022
RF 7.1	Cálculo do resultado geral	1	Instituto Verbena	21/12/2022
RF 7.2	Geração de arquivo de resultado geral	1	Instituto Verbena	21/12/2022

Fonte: próprios autores

3.3. Requisitos Não Funcionais

RNF 1.0 - Forma de acesso

Responsável: Marcos

Prioridade: Baixa

Descrição: O sistema deve estar disponível para acesso na internet por meio de uma URL.

RNF 2.0 - Compatibilidade

Responsável: Marcos

Prioridade: Baixa

Descrição: O sistema deve ser compatível com o navegador Google Chrome.

RNF 3.0 - Segurança na transmissão de dados

Responsável: Marcos

Prioridade: Baixa

Descrição: O protocolo HTTPS deve ser utilizado no acesso ao sistema a fim de garantir a confidencialidade dos dados transmitidos pelo Instituto Verbena ao sistema, visto que os arquivos enviados podem conter informações confidenciais.

RNF 4.0 - Segurança no armazenamento de dados

Responsável: Marcos

Prioridade: Baixa

Descrição: Uma vez que os arquivos fornecidos pelo Instituto Verbena (dados de candidatos, notas ENEM, notas VHCE) sejam processados, tais arquivos devem ser apagados do servidor que hospeda o sistema.

RNF 5.0 - Disponibilidade

Responsável: Marcos

Prioridade: Baixa

Descrição: Por se tratar de um trabalho acadêmico, o sistema deverá estar disponível para uso durante a duração da disciplina de Arquitetura de Software no período acadêmico 2022-2. Após esse prazo, a critério do docente, o código-fonte poderá ser utilizado para manter o sistema no ar em um ambiente de responsabilidade do docente ou da universidade.

RNF 6.0 - Compatibilidade de Interface

Responsável: Marcos

Prioridade: Baixa

Descrição: A interface do sistema deve ser compatível com monitores de resolução 800x600. Resoluções maiores também devem ser suportadas.

3.4. Restrições

RE 1.0 Integração entre Instituto Verbena e INEP

Os dados cadastrais do Instituto Verbena são apresentados em várias tabelas, como: Cadastro_Concurso.ods, Candidato.ods, Cargo.ods, programas.ods e Situacao_Inscricao.ods. Como não há API de integração entre o Portal do Candidato com o INEP, o responsável pelo processo seletivo deve realizar uma atividade manual de compilação dos dados cadastrais para o formato de arquivo aceito pelo INEP. O software MPA não inclui o desenvolvimento de API de integração.

3.5. Modelo de Casos de Uso (MCU)

O Modelo de Casos de Uso para o software MPA é:

Figura 3 - Modelo de Casos de Usos (MCU).



Fonte: próprios autores.

4. ARQUITETURA

Este capítulo foca na descrição da arquitetura (*architecture description* - AD), pontos de vistas, modelos e interesses, conforme ISO 42010:2022. Para isso, as atividades principais contempladas são:

- Listar os atributos de qualidade a partir dos RF e RNF, utilizando ISO 25010:2011;
- Detalhar quais desses atributos de qualidade estão associados a arquitetura do software, abordando os pontos de vista dos usuários do MPA e dos desenvolvedores (back-end e front-end). Isso deve considerar uma análise crítica do MPA como *Minimum Viable Product* (MVP);
- Avaliar se as arquiteturas selecionadas condizem com os atributos de qualidades levantados para cada ponto de vista e interesses, conforme ISO 42010:2022.

Antes de iniciar a descrição da arquitetura este trabalho apresenta três definições do termo que são usados como premissas.

A arquitetura do software é...

- O projeto ou modelagem do sistema de software que delimita a sua estrutura, componentes e interações;
- A fundação e arcabouço de construção do software que fornece uma estrutura geral e diretrizes para o desenvolvimento do sistema;
- Uma estrutura organizada do sistema que determina como as partes devem trabalhar juntas para atingir a funcionalidade desejada.

4.1. Tradução dos Atributos de Qualidade

A tradução dos RF e RNF em atributos de qualidade é a etapa que mais exige atenção na definição da AD. A Figura 4 apresenta os atributos de qualidade de acordo com a ISO 25010:2011.

Figura 4 - Atributos de qualidade.



Fonte: notas de aula de Arquitetura de Software por Jacson Rodrigues 2023-1.

Na ISO 25010:2011 os atributos de qualidade são organizados em oito itens. Cada um com subitens precisam ser completamente atendidos para que aquele atributo de qualidade seja considerado implementado no software.

A arquitetura de um novo software deve ser dividida em uma estratégia para o front-end que não necessariamente se repete no back-end. Para o MPA foram analisados a aplicabilidade dos oito grupos, considerando o back-end inicialmente, e o front-end em um segundo momento. A partir da lista de atributos de qualidade da ISO 25010:2011 todos os itens foram traduzidos para o português e listados nesta seção.

4.1.1. Adequação Funcional

Integridade Funcional

Mede se o software inclui todas as características e funções necessárias para atingir seu objetivo.

Corretude Funcional

Mede se o software realiza o que deveria realizar de forma precisa e confiável.

Adequação Funcional

Mede se as características e funções do software atendem de forma adequada e relevante a expectativa do usuário.

4.1.2. Eficiência de Desempenho

Comportamento no tempo

Mede a capacidade do software responder às interações do usuário ou processos de tarefas, garantindo responsividade.

Utilização de recursos

Mede a capacidade do software de utilizar os recursos disponíveis, como CPU, memória e rede para realizar tarefas sem desperdício.

Capacidade de processamento

Mede a capacidade do software em lidar com uma quantidade específica de tarefas ou concorrência sem lentidão ou gargalos.

4.1.3. Compatibilidade

Coexistência

Mede a capacidade do software em operar sem conflitos com outros softwares ou sistemas.

Interoperabilidade

Mede a capacidade do software em trocar informações com outros sistemas e componentes sem ruídos ou perdas de dados.

4.1.4. Usabilidade

Capacidade de ser reconhecido

Mede a capacidade do software em ter características e funcionalidades devidamente projetadas para serem reconhecidas pelo usuário, facilitando o uso e o entendimento do software.

Capacidade de aprendizado

Mede a capacidade do software de ser aprendido pelo usuário sem muito tempo de treinamento.

Capacidade de operação

Mede a capacidade do software de ser operado pelo usuário sem muito tempo de treinamento, com interações intuitivas e fluidas.

Proteção de erros do usuário

Mede a capacidade do software em minimizar o impacto de erros inseridos pelo usuário, prevenindo erros críticos e fornecendo *feedback*.

Estética da interface do usuário

Mede a capacidade do software em garantir uma experiência frutífera, agradável e atrativa para o usuário.

Acessibilidade

Mede a capacidade do software em acomodar usuários de vários espectros de habilidades e restrições

4.1.5. Confiabilidade

Maturidade

Mede a capacidade do software em ser estável e livre de defeitos até que seja liberado para produção.

Disponibilidade

Mede a capacidade do software em permanecer consistente para o uso sempre que o usuário precisar.

Tolerância a falhas

Mede a capacidade do software em continuar funcionando, ou ter uma perda suave de degradação, quando encontra erros ou falhas.

Recuperabilidade

Mede a capacidade do software em recuperar dados e funcionalidades após um erro ou falha, garantindo a integridade dos dados e permitindo a restauração do sistema.

4.1.6. Segurança

Confidencialidade

Mede a capacidade do software em garantir que informações sensíveis sejam mantidas privadas e acessível apenas a usuários autorizados.

Integridade

Mede a capacidade do software em resguardar o código do software contra alterações não autorizadas.

Não aceitabilidade/repúdio

Mede a capacidade do software em prevenir que os usuários neguem as ações do software, garantindo que as ações e transações sejam verificadas e formalmente vinculadas.

Autenticação

Mede a capacidade do software em verificar a identidade dos usuários e entidades que o acessa, prevenindo fraudes e garantindo acessos legítimos.

Responsabilização

Mede a capacidade do software em responsabilizar o usuário por suas ações no software, garantindo rastreabilidade auditável e detectando potenciais falhas de segurança.

4.1.7. Manutenibilidade

Modularidade

Mede a capacidade do software em ser organizado em módulos separados, autocontidos e com baixo acoplamento, permitindo fácil manutenção e atualização de partes específicas sem afetar todo o sistema.

Reusabilidade

Mede a capacidade do software em utilizar seus componentes e módulos em múltiplos contextos para reduzir o esforço de desenvolvimento de novos recursos e permitindo o reuso de componentes.

Analísabilidade

Mede a capacidade do software de ter o seu código e comportamento facilmente compreendido e analisado por outros desenvolvedores, auxiliando na identificação e correção durante a manutenção.

Modificabilidade

Mede a capacidade do software em ser facilmente modificado e adaptado para acomodar mudanças naturais em sua evolução, como atender novos requisitos sem causar grande impacto no sistema.

Testabilidade

Mede a capacidade do software em facilitar o desenvolvimento e execução de testes, facilitando a detecção de defeitos durante a manutenção.

4.1.8. Portabilidade

Adaptabilidade

Mede a capacidade do software em ajustar as suas funções corretamente para vários ambientes de implantação sem ser necessário modificações significativas.

Instalabilidade

Mede a capacidade do software em ser facilmente instalado e configurado em diferentes sistemas operacionais e ambientes.

Capacidade de substituição

Mede a capacidade do software em ser facilmente substituído por outra versão ou sistema, fornecendo flexibilidade em alterações futuras ou *upgrades*.

4.2. Trade-off para o MPA

Analisando as explicações sobre os atributos de qualidade, um desenvolvedor deseja atingir o máximo de vantagens entre todos aqueles indicados na ISO 25010:2011. Porém, sempre há um *trade-off*, ou seja, uma relação de custo-benefício que não permite a cobertura de todos os benefícios.

O domínio do software no escopo deste projeto não exige alta performance, uma vez que os prazos de processamento dos dados pelo IV respeitam o cronograma do edital, com duração de meses até a publicação do resultado definitivo. Apesar do nome do software ser Módulo de Processamento Ágil (MPA), o edital define prazos de dias para publicação das etapas do processo seletivo em questão em virtude de regras jurídicas típicas, como tempo para publicações no IV, submissão e análise dos recursos dos candidatos.

A compatibilidade e a portabilidade, para a situação atual, também são irrelevantes, uma vez que o MPA deve ser implementado em servidor local da UFG, em um único sistema operacional previamente definido, sem possuir qualquer integração com outros sistemas e sem ser necessária modificação do software para ser executado em vários ambientes, pois ele não será. Por utilizar o upload e download de arquivos .txt, o MPA não exige integração, funcionando como um módulo acelerador de processamento isolado na versão MVP.

Dos oito atributos de qualidade, três foram diretamente descartados nessa análise (Eficiência de Desempenho, Compatibilidade e Portabilidade), restando cinco (Adequação Funcional, Usabilidade, Confiabilidade, Segurança e Manutenibilidade). Na versão MVP, dos cinco atributos de qualidade restantes apenas dois são críticos para cada frente de trabalho, back-end e front-end.

Para o back-end os dois atributos de qualidade críticos são:

- Segurança, sustentada pelo uso do token/middleware, acesso exclusivo por funcionário do IV, clara divisão de responsabilidade por camadas e restrição de acesso ao MPA por IP do IV.

- Manutenibilidade, sustentada por um código que contempla todas as características presentes nesse atributo, como modularidade, reusabilidade, analisabilidade, modificabilidade e testabilidade.

Para o front-end são dois atributos de qualidade críticos:

- Segurança, pelas mesmas razões do back-end, por ser um atributo de qualidade transversal, porém tratado de formas diferentes.
- Usabilidade, pela alimentação direta de arquivos de entrada e saída a cada etapa do processo seletivo, de forma intuitiva, fácil de aprender, simples e objetiva para cada etapa do processo seletivo.

Essas decisões permitem o desenvolvimento de um *Minimum Viable Product* (MVP) capaz de entregar valor ao IV. Em um produto de mercado, o fluxo a ser seguido nesta etapa pode ser visto como:

- Product **Owner (PO)** aponta **requisitos** funcionais e não funcionais;
- Os requisitos são traduzidos pela equipe de desenvolvedores em **atributos de qualidade prioritários**;
- Avalia-se os prós e contras de cada atributos de qualidade para se obter a **melhor arquitetura** para o sistema;
- A seleção dos atributos de qualidade permitem escolher os **estilos arquiteturais**, ou mescla deles, que atendem ao PO;
- Conforme o objetivo do projeto, registra-se **o que se ganha e o que se perde** com as decisões tomadas.

4.3. Estilos arquiteturais relevantes

4.3.1. Arquitetura de Camadas para o back-end

A partir da seleção dos atributos de qualidade para o back-end fica evidente que o padrão arquitetural Arquitetura de Camadas atende integralmente aos benefícios buscados pelo projeto.

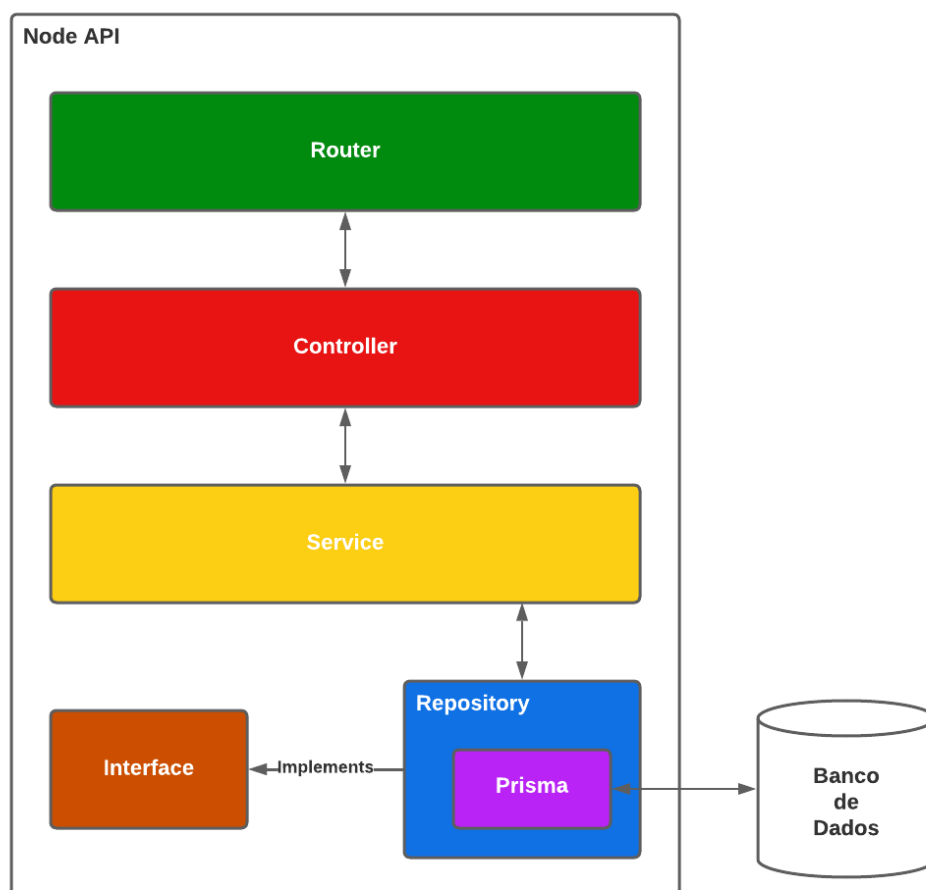
O uso de camadas permite a divisão clara de responsabilidades, fácil entendimento das funções, aplicação de regras próprias de segurança por camada e simplifica os testes, pois cada camada pode ser testada de forma independente, facilitando a identificação e correção de problemas. Os módulos independentes facilitam o desenvolvimento, manutenção e evolução de cada camada de forma

separada. Isso facilita a reutilização de componentes em diferentes partes do sistema.

O uso de camadas busca trabalhar o desacoplamento das camadas de acordo com o arranjo:

- Camada de apresentação: rotas e controladores.
- Camada de negócio: serviços, regras de negócio e interface de desacoplamento com a camada de persistência.
- Camada de persistência: PRISMA ORM e abstração das *queries*.
- Camada do banco de dados: SGBD em PostgreSQL.

Figura 5 - Diagrama de Arquitetura do Sistema.



Fonte: próprios autores.

4.3.2. Arquitetura Híbrida para o front-end

A partir da seleção dos atributos de qualidade para o front-end é necessário uma mescla de padrões arquiteturais, por se basear em dois pilares principais: praticidade no desenvolvimento e escalabilidade. A partir dessa análise sugere-se o uso da mescla do Flux com Clean Architecture.

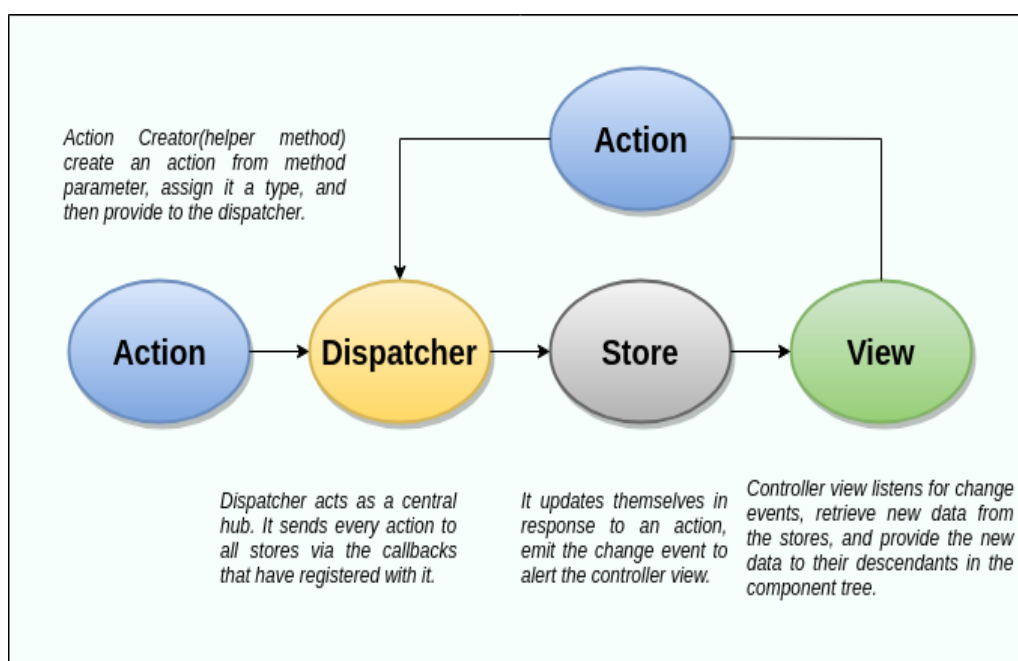
O Flux direciona o desenvolvimento de uma aplicação centralizada nos dados com uso dos componentes: *Views*, *Actions*, *Dispatchers* e *Stores*. Cada um exercendo um papel fundamental para a consistência de dados e praticidade no uso.

Os componentes são:

- *Views*: interfaces de usuário;
- *Actions*: representação de eventos e interações de usuários com a aplicação;
- *Dispatchers*: mensageiros que, através de um dado *Action*, acessam e manipulam uma *Store* para realizar a ação pretendida;
- *Stores*: pontos único de armazenamento e processamento dos dados da aplicação, do qual componentes funcionais e interfaces confiam e consomem.

A Clean Architecture separa interesses e facilita a manutenção do código em uma base "limpa", ou seja, um código que é facilmente mantido, testado, atualizado e compreendido pelos desenvolvedores.

Figura 6 - Arquitetura Flux.



4.4. *Stakeholders*, concerns e pontos de vista

A partir da definição de arquitetura, o projeto exige a tradução dos requisitos funcionais e não funcionais em **atributos de qualidade** para o arquiteto de software, ou seja, é necessário identificar atributos e/ou características mensuráveis entre os requisitos de software que contribuem com a qualidade geral e a performance do sistema para cada ponto de vista e interesse.

Considera-se aqui os pontos de vista e interesses dos *stakeholders*. Cada ponto de vista apresentado gera pelo menos uma visão sobre a arquitetura do software e que utiliza um modelo como base. O modelo 4-1 de Philippe Kruchten. O foco é abordar o que se espera para a documentação. No caso do MPA a documentação serve de referência para instruir os usuários sobre uma primeira melhoria dos processos do IV sobre o UFGInclui e já apresentar possibilidades para um cenário futuro completamente automatizado. O projeto apresenta uma primeira camada de interessados, como:

- Universidade Federal de Goiás (UFG): segurança e eficiência dos processos administrativos do edital UFGInclui: indígenas brasileiros e negros quilombolas;
- Instituto Verbena: automação das etapas do processo seletivo, com redução do uso de planilhas e trabalhos manuais para aplicação das regras definidas no edital. O Instituto Verbena no modelo atual possui equipe reduzida e celebra pouco mais de 1 ano de existência. No processo seletivo UFGInclui, por exemplo, várias etapas são realizadas em planilhas e macros, com consultas manuais das notas de cada candidato para cada ano dos últimos dez anos. Como, no momento atual, não há API no IV ou no INEP, o MPA assume como premissa a integridade e segurança dos dados a cada etapa do processo seletivo, pois eles serão manipulados pelos responsáveis do processo seletivo no IV. Em um momento futuro, com a existência das API do Portal do Candidato/IV e INEP, há necessidade de alta modularidade do MPA para uma manutenção evolutiva que conecte todos esses componentes e mantendo elevado desacoplamento e a aplicação coesa das regras de negócio estabelecidas;

- Os dois responsáveis pelos processos seletivos do UFGInclui, um pelo trabalho administrativo e outro pelo trabalho jurídico: clareza sobre as regras definidas no edital (visão jurídica) e facilidade na execução das etapas do processo seletivo (visão administrativa);
- Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP): para o INEP a saída do MPA deve ser formatada, com a lista de CPF de todos os candidatos em arquivo .txt, gerando como saída o arquivo .txt com todas as notas dos candidatos listados através de um acesso exclusivo do IV/UFG;
- Candidato do processo seletivo no edital UFGInclui: indígenas brasileiros e negros quilombolas: ter seus dados pessoais seguros, aplicados para a execução do processo seletivo, e ter publicidade do resultado do edital. Os candidatos, porém, não têm interação direta com o MPA. Para eles há publicidade dos resultados parciais e definitivos publicados pelo IV, com as respectivas regras de prazos e recursos típicas de editais dessa natureza.
- Desenvolvedores, instaladores e mantenedores do MPA: para a construção do MVP os recursos foram dedicados exclusivamente nas atividades de pesquisa, registro e desenvolvimento de software do back-end e front-end. Para complementar esses pontos de vista é relevante incluir de quem instalar/implantar o MPA na UFG, além dos pontos de vista dos desenvolvedores e usuários.

A partir desse agrupamento preliminar é possível separar a perspectiva que os desenvolvedores, implementadores e utilizadores do MPA sobre o projeto, como:

- Responsáveis pela instalação e gestão do MPA
 - Preocupações: ter a descrição do processo de instalação do MPA no servidor de forma assertiva e que permita a operação correta do software como ele foi projetado.
 - Ponto de vista: contempla a descrição detalhada sobre os requisitos de hardware, configurações de rede, instalação de sistemas fundamentais com as ferramentas e versões necessárias, medidas de segurança a serem tomadas, entre outras ações, que permitam o uso e manutenção do MPA.

Por se tratar de uma instituição pública federal, inúmeros procedimentos e detalhes de implantação precisam ser seguidos e que estão fora do domínio de conhecimento deste grupo de trabalho. De uma forma geral, as tecnologias, componentes e configurações para o uso do MVP são indicadas nos tópicos 4.3, 4.5 e 4.6 de forma descritiva em vez de um Diagrama de Componentes UML.

- **Usuários do MPA no Instituto Verbena**

- Preocupações: ter acesso a um software amigável, fácil de aprender, fácil de usar e que atenda aos requisitos de processamento necessários para o processamento dos dados previstos no edital.
- Ponto de vista: contempla a descrição detalhada sobre os requisitos de interface, funcionalidades e atendimento ao que é solicitado pelo processo seletivo UFGInclui, entre outras ações, que permitam o uso consistente do MPA.

Com o desenvolvimento do MVP, todo o processo de uso do MPA apresentou uma interface com fluxo bem definido, confiável e capaz de ser alimentada e acompanhada a cada etapa do processo seletivo. A interface é web, acessada a partir de um IP definido do Instituto Verbena, com controle de acesso, cadastro de processos seletivos, upload e download de dados via arquivos .txt a partir dos dados das etapas, e download de resultado definitivo para publicação no Portal do Candidato, conforme previsto no edital.

- **Desenvolvedor Back-end**

- Preocupações: desenvolver um software, na camada de regras de negócio, que contemple todas as decisões arquiteturais definidas ao longo do projeto, de forma limpa, desacoplada, coesa, segura e que garanta a integridade dos dados e consistência do uso do MPA.
- Ponto de vista: para o cenário atual, o desenvolvedor back-end dedica a atenção na responsabilidade de cada camada do software, conforme descrito no item 4.3 e 4.5.

Com o desenvolvimento do MVP, todas as regras de negócio contempla a aquisição de informações dos candidatos do edital UFGInclui negros quilombolas e indígenas brasileiros, com uso dos

dados pessoais para geração de arquivo com o CPF. Este arquivo é carregado no site do INEP através de um acesso da UFG para que seja feito o download das notas do ENEM, segmentada, dos últimos dez anos dos candidatos, quando houver. O arquivo INEP é carregado novamente no MPA para avaliação de aptidão para VHCE, seguido do upload da nota de VHCE, cálculo das notas dos candidatos, conforme os critérios de seleção por pesos nas áreas de conhecimento. Por fim, há a publicação do resultado final do processamento, com a emissão dos candidatos aprovados de acordo com as regras do edital.

- **Desenvolvedor Front-end**

- Preocupações: desenvolver um software, na camada de interface, que contemple todas as decisões arquiteturais definidas ao longo do projeto, de forma limpa, amigável, fácil de aprender, fácil de usar, segura e que garanta a integridade dos dados e consistência do uso do MPA.
- Pontos de vista: para o cenário atual, o desenvolvedor front-end dedica a atenção na responsabilidade de cada camada de interface, conforme descrito no item 4.3 e 4.6.

Com o desenvolvimento do MVP, todas as funcionalidades de interação com os usuários contempla a carga inicial de dados dos candidatos do edital UFG. Inclui negros quilombolas e indígenas brasileiros, com arquivos em extensão .txt, .csv e/ou .ods para geração de outros arquivos necessários ao longo do processo definido no back-end.

4.5. Por que se usa arquitetura em camadas no back-end?

A arquitetura em camadas facilita a coesão do código e o desacoplamento, extremamente similar ao padrão OSI de camadas, com organização de um código coeso e definido, pronta para contemplar as várias visões, como usuário, desenvolvedor, implantador e entidades que por ventura venham a interagir com o MPA. Dessa forma, a comunicação entre front-end e back-end irá utilizar a arquitetura cliente-servidor, com as devidas responsabilidades das camadas que tratam os *endpoints*.

4.6. Por que se usa Flux e Clean Architecture no front-end?

A escolha específica das duas arquiteturas é proposital, no sentido de que a primeira exerce um importante papel no front-end quando combinada com a segunda, e vice-versa.

O Flux é escolhido à dedo justamente pelo papel de protagonista que os dados exercem dentro da dinâmica do MPA. Como o sistema é, de forma sucinta, um gerenciador de processo seletivo com diversas etapas, os dados criados e manipulados dentro do fluxo de execução de um processo moldam por completo a maneira como o mesmo vai se desenvolvendo. Dessa forma, ter no front-end (ponte entre o sistema e o usuário manipulador dos dados) uma arquitetura que se baseia por completo na manipulação dessas informações, promovendo consistência e praticidade, é um ganho inestimável.

Clean Architecture, por sua vez, combina muito com a ideologia do Flux no que diz respeito a separação de responsabilidades. A ideia de componentização do Flux em 4 diferentes camadas de tratamento e exibição dos dados, juntamente à uma componentização de outras camadas pelo Clean Architecture, como a camada de serviços, apresentação, estruturação e autenticação e outras diretrizes de desenvolvimento, geram um produto final que é seguro, facilmente escalado, mantido e rapidamente desenvolvido.

Figura 7 - Diagrama de entidades.

Cadastros	
nome_candidato	INT
cpf	VARCHAR
sexo	ENUM
data_nascimento	DATE
estado_civil	ENUM

Cargos	
co_ies_curso	INT
desc_cargo	VARCHAR
desc_grau_academico	ENUM
desc_perodo	ENUM
desc_cidade	VARCHAR
codg_cargo	INT
numr_vagas	INT
numr_grupo	Tipo
desc_regional	VARCHAR

Notas_enem	
numr_enem	INT
cpf	VARCHAR
nome_candidato	VARCHAR
nota_ciencias_natureza	FLOAT
nota_ciencias_humanas	FLOAT
nota_linguagens	FLOAT
nota_matematica	FLOAT
nota_redacao	FLOAT
nota_total	FLOAT

Candidatos	
numg_candidato	INT
cpf	VARCHAR
codg_cargo	INT
cor_raca	ENUM
formacao_escola_publica	BOOLEAN
data_inscricao	DATE
codg_situacao	INT
flag_inscricao	INT
programa	INT
tipo_programa	INT
nome_comunidade	VARCHAR
ano_enem	INT
valor_pf	FLOAT
numr_classificacao	INT
nota_vhce	FLOAT
semestre_ingresso	INT

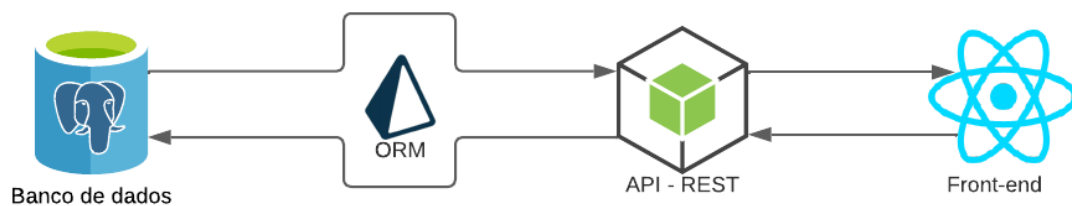
Notas_enem	
num_candidato	INT
ausente	BOOLEAN
nota_1	VARCHAR
nota_2	FLOAT
nota_total	FLOAT

Programas	
codigo	INT
tipo_programa	INT
desc_programa	VARCHAR
categoria_ingresso	VARCHAR

Fonte: próprios autores.

4.6.1. Tecnologias escolhidas

Figura 8 - Tecnologias utilizadas no MVP do MPA.



Fonte: próprios autores.

5. MELHORIAS POSSÍVEIS

No cenário atual o MPA exige grande interação com os responsáveis do processo seletivo UFGInclui negros quilombolas e indígenas brasileiros. Essa decisão é baseada no grau de maturidade dos sistemas que o IV e INEP utilizam, uma vez que nenhum deles possui API de integração com seus respectivos bancos de dados. Assim, para ampliar a automação do MPA recomenda-se o desenvolvimento dessas API e de um novo software para o processamento dos dados do edital. O MPA utiliza uma arquitetura e estratégia de desenvolvimento focada em um MVP, sem a necessidade de atender as integrações com outros sistemas. Porém, para isso, utiliza ainda trabalhos manuais de download e upload em diferentes plataformas, o que amplia a inércia do processo seletivo e fomenta pontos de risco com a utilização de tramitação de dados fora de um sistema, ou seja, sem considerar todas as regras, cuidados e riscos para a comunicação daqueles dados.

Em um cenário ideal, o MPA será alimentado uma única vez pelo responsável do processo seletivo, uma vez que sua arquitetura contemplará as informações do Portal do Candidato do IV, das notas do INEP e das notas de VHCE do IV. Os resultados, parciais ou definitivos, são carregados imediatamente no IV a cada etapa do processo seletivo, cabendo à área jurídica a análise e ações dos recursos tipicamente indicados em editais. A nível de processamento de dados, dessa forma, o MPA de fato implementa uma solução segura, ágil, eficiente e eficaz para o IV-UFG. Em um último momento, este modelo pode ser reutilizado para ferramentas semelhantes de processos seletivos públicos similares ao UFGInclui.

REFERÊNCIAS

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. Capítulo 5: Engenharia de Requisitos. Editora McGraw Hill, 7a. edição, 2011.

TOWARDS DATA SCIENCE, disponível em <<https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nut-shell-a0b47a1e9013>>, consultado em 07 de dezembro de 2022.

UNIVERSIDADE FEDERAL DE GOIÁS, Instituto Verbená, Processo Seletivo Para Ingresso Nos Cursos de Graduação Do Programa UFGInclui - Indígenas e Negros(as) Quilombolas - 2023, disponível em: <https://sistemas.institutoverbena.ufg.br/2023/ps-ufg-inclui/sistema/arquivos/editais/EDITAL_ABERTURA_UFGINCLUI_INDIGENAS_QUILOMBOLAS_2023_Compilado.pdf>, consultado em: 24 de novembro de 2022.

ABNT. ISO 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

ABNT ISO 42010:2022 Software, systems and enterprise — Architecture description.