



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ALOCAÍ

DOCENTE: RAONI KULESZA

DISCENTES:

CAMILA EDUARDO COSTA DE VASCONCELOS
LUCAS FARIAS DE MEDEIROS
LUERSON DE ALBUQUERQUE SILVA FILHO
LUIZ HENRIQUE QUEIROZ DE ALBUQUERQUE SILVA
MAURO COIMBRA BARRETO COSTA NETO

PERÍODO: 2024.2

Histórico de Revisões

Data	Versão	Descrição	Autor(es)
05/02/2025	1.00	Versão inicial do documento de requisitos	Camila Costa, Lucas Farias, Luiz Queiroz, Luerson Filho, Mauro Coimbra
06/02/2025	1.01	Adição de Requisitos Funcionais após entrevistas	Camila Costa, Lucas Farias, Luiz Queiroz, Luerson Filho, Mauro Coimbra
07/02/2025	1.02	Revisão geral do documento	Camila Costa, Lucas Farias, Luiz Queiroz, Luerson Filho, Mauro Coimbra

Sumário

1. Introdução.....	2
2. Descrição geral.....	4
3. Glossário.....	8
4. Elicitação de Requisitos.....	9
5. Especificação de Requisitos.....	11

1. Introdução

1.1. Propósito do documento

Este documento tem como propósito a especificação do projeto **ALOCAÍ**, incluindo uma descrição geral, a elicitación de requisitos, a análise e especificação desses requisitos, a análise dos casos de uso, uma descrição da interface com o usuário e a definição dos diagramas e arquitetura do projeto.

1.2. Visão geral do documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto, além de conter uma lista de referências para outros documentos relacionados. As demais seções apresentam a especificação do sistema **ALOCAÍ** e estão organizadas como descrito abaixo.

- Seção 2 – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- Seção 3 – Glossário: apresenta definições de todas as entidades referenciadas neste documento. É uma seção centralizadora de informações de dados do negócio, servindo como uma espécie de dicionário para este documento.
- Seção 4 – Elicitación de Requisitos: especifica os processos aplicados para elicitación de requisitos do sistema, com detalhamentos das formas e contribuições de cada método.
- Seção 5 – Especificação de Requisitos: apresenta todos os requisitos funcionais e não funcionais do sistema, juntamente com o nível de importância de cada um.

1.2. Documentos relacionados

Documentos relacionados ao **ALOCAÍ** e/ou mencionados nas seções a seguir:

1. Glossário; Seção 3 deste documento.
2. ANEXO A; Data: 06/02/2025; Responsável pelo documento: Luerson.
3. ANEXO B; Data: 06/02/2025; Responsável pelo documento: Luerson.
4. ANEXO C; Data: 07/02/2025; Responsável pelo documento: Luerson.

2. Descrição geral

A aplicação tem como objetivo tornar acessível aos coordenadores de cursos a alocação eficiente de disciplinas e professores, sem a necessidade de conhecimento prévio em computação ou pesquisa operacional. Para isso, oferece uma solução intuitiva, específica para o contexto acadêmico, permitindo que os coordenadores realizem alocações otimizadas de horários de professores sem precisar lidar com modelagem matemática complexa.

O sistema utiliza algoritmos consagrados, como o Simplex e técnicas de Programação Inteira Mista (MIP), para resolver problemas de otimização de forma eficiente. A modelagem matemática é gerada automaticamente com base nos dados fornecidos pelo usuário, eliminando a necessidade de conhecimentos técnicos avançados.

Essa abordagem simplifica o processo de alocação de disciplinas e professores, reduzindo a carga administrativa dos coordenadores e garantindo uma distribuição eficiente de horários e docentes, de forma acessível e escalável para a instituição.

2.1. Motivação

A criação de uma aplicação para alocação de disciplinas e professores visa atender à crescente demanda por soluções eficientes e acessíveis para instituições de ensino, facilitando o planejamento acadêmico e a gestão de horários. Muitas vezes, as equipes de gestão enfrentam desafios complexos ao tentar alocar recursos humanos de forma eficaz. A aplicação propõe simplificar esse processo, oferecendo uma interface intuitiva e funcionalidades que permitam aos usuários, independentemente de sua familiaridade com conceitos técnicos, realizar alocações de pessoal de maneira estratégica e eficiente.

2.2. Problemas identificados

Muitas instituições de ensino enfrentam desafios ao alocar disciplinas e professores de maneira eficiente, especialmente quando não contam com ferramentas especializadas ou quando os coordenadores não possuem conhecimento técnico em otimização. A falta de um planejamento adequado pode resultar em choques de horário, subutilização de recursos e dificuldades na organização acadêmica.

Além disso, a alocação de disciplinas e professores frequentemente se torna um processo complexo e sujeito a erros, especialmente ao lidar com múltiplas restrições acadêmicas e ajustes de última hora. Isso pode gerar choques de horário, dificuldade no cumprimento da carga horária dos professores e uma

distribuição ineficaz dos recursos disponíveis.

Com essa aplicação, espera-se eliminar essas ineficiências, proporcionando uma solução acessível e automatizada para a gestão acadêmica.

2.3. Visão geral do sistema

O sistema proposto tem como objetivo principal facilitar a alocação eficiente de disciplinas e professores, permitindo que os coordenadores distribuam horários de forma otimizada, sem a necessidade de conhecimento técnico avançado em pesquisa operacional ou computação. Para isso, a aplicação oferecerá funcionalidades como o cadastro e gerenciamento de professores e disciplinas, permitindo registrar docentes, definir suas restrições de horário e alocar recursos de forma eficiente.

A visualização e o ajuste das alocações serão acessíveis exclusivamente para coordenadores ou usuários com nível de autoridade equivalente. Além disso, o sistema contará com um painel de controle que permitirá o monitoramento em tempo real da distribuição de horários, possibilitando ajustes rápidos sempre que houver mudanças na disponibilidade de professores.

No entanto, algumas funcionalidades não estarão incluídas no escopo do sistema. O gerenciamento de desempenho dos professores e análises detalhadas sobre carga horária não serão contemplados, uma vez que esse tipo de avaliação exige ferramentas especializadas. Além disso, a aplicação não terá integração com sistemas acadêmicos externos, como plataformas de gestão administrativa, nem oferecerá funcionalidades de comunicação interna, como chats ou notificações automáticas para os docentes.

O sistema será totalmente independente, sem necessidade de integração com outras plataformas. No entanto, em versões futuras, poderão ser exploradas possibilidades de conexão com sistemas acadêmicos externos para ampliar suas capacidades e atender a demandas mais específicas.

2.4. Usuários do sistema

A aplicação foi projetada para atender coordenadores de cursos e gestores acadêmicos que precisam realizar a alocação eficiente de disciplinas e professores, mas que não possuem conhecimento técnico em computação ou pesquisa operacional. Esses usuários podem incluir coordenadores de cursos, administradores acadêmicos e outros profissionais envolvidos na gestão de horários e recursos da instituição.

A produtividade dos usuários alvo do sistema costuma ser muito limitada devido ao trabalho manual

necessário para realizar a alocação de horários e professores, independentemente do contexto acadêmico. O sistema proposto visa tornar esse processo muito mais rápido e eficiente, permitindo que os usuários utilizem o tempo economizado em outras atividades significativas, como planejamento curricular e atendimento aos alunos.

Os principais desafios enfrentados por esses usuários incluem:

- **Falta de conhecimento técnico:** Muitos coordenadores e gestores acadêmicos não estão familiarizados com modelagem matemática ou algoritmos de otimização, tornando difícil o uso de ferramentas complexas de programação.
- **Processos manuais e ineficientes:** A alocação de horários e professores frequentemente é feita de forma empírica ou por tentativa e erro, resultando em desperdício de tempo e recursos.
- **Dificuldade em lidar com restrições complexas:** É necessário considerar múltiplas variáveis e restrições (como carga horária e disponibilidade dos professores), o que pode ser desafiador sem suporte tecnológico.
- **Necessidade de escalabilidade:** Instituições de ensino precisam de soluções que possam ser aplicadas a diferentes níveis de complexidade, desde pequenos cursos até grandes centros acadêmicos.

Em versões posteriores do projeto, coordenadores e gestores poderão permitir que os professores acessem seus próprios cadastros e visualizem suas alocações, recebendo avisos automaticamente sempre que houver mudanças nos horários. Isso reduzirá a responsabilidade do administrador em avisar constantemente os docentes sobre alterações nos horários.

2.5. Suposições e restrições gerais

Há dois limitadores importantes a serem considerados pelo sistema: 1) memória principal exigida e 2) tempo de execução dos modelos de programação linear inteira.

A memória principal se apresenta como um possível limitador do sistema, dado que o uso de memória tende a crescer rapidamente à medida que o número de disciplinas e professores a serem alocados aumenta. Esse crescimento não linear ocorre à medida que o número de restrições e variáveis no processo de alocação se expande, o que pode impactar a performance do sistema conforme a

complexidade da alocação acadêmica aumenta.

Já o tempo de execução do modelo se mostra como um ponto de atenção devido ao tempo que os algoritmos, mesmo considerados estado da arte, podem demorar para resolver os problemas. Uma das principais propostas do sistema é ser eficiente, então deixar o usuário esperando demais é algo totalmente indesejado. Essa situação, contudo, só costuma acontecer quando há muitas alocações para serem feitas e, portanto, só deve se tornar algo preocupante quando for necessário escalar os servidores para suportarem grandes requisições.

3. Glossário

Perfil do Colaborador

- Representa os dados de um colaborador cadastrado pelo gestor.
- A carga horária diária máxima de um colaborador é de 2 turnos.
- **Atributos:**
 - **Nome Completo:** Campo alfabético de até 100 caracteres.
 - **ID:** Identificador numérico, geralmente gerado automaticamente pelo sistema da empresa no momento da admissão do funcionário.
 - **Disciplinas:** Identificadores textuais das disciplinas que o professor leciona.
 - **Turnos:** Identificadores numéricos dos turnos (manhã, tarde ou noite) e dias da semana (de segunda a sexta-feira) para os quais o profissional está alocado.

Perfil da Disciplina

- Representa os dados de uma determinada disciplina a ser alocada.
- **Atributos:**
 - **Nome:** Campo alfabético.
 - **ID:** identificador numérico único.
 - **Turnos:** turnos para os quais a disciplina está alocada.

Perfil do Turno

- Representa os dados de um turno em específico.
- **Atributos:**
 - **Dia da Semana:** dia do turno.
 - **ID:** identificador numérico único.
 - **Colaboradores:** lista de colaboradores alocados para o turno.

Planilha de Colaboradores

- Representa os dados organizados dos professores em uma planilha exportada pelo programa.
- **Atributos:**
 - **Todas as informações do Perfil do Colaborador**, com cada dado armazenado como um atributo independente.

Planilha de Disciplinas

- Representa os dados organizados das disciplinas em uma planilha exportada pelo programa.
- **Atributos:**
 - **Todas as informações do Perfil da Disciplina**, com cada dado armazenado como um atributo independente.

Modelagem Matemática

Representação de um problema do mundo real por meio de expressões matemáticas, permitindo sua análise e solução utilizando técnicas quantitativas.

Programação Linear Inteira Mista (MIP)

Técnica de otimização que envolve variáveis contínuas (fracionárias) e inteiras, usada para resolver problemas de alocação de recursos, planejamento e logística.

Simplex

Algoritmo clássico para resolver problemas de **programação linear**, otimizando funções sujeitas a restrições matemáticas.

Pesquisa Operacional

Área da matemática aplicada que utiliza modelos quantitativos para auxiliar na tomada de decisões e otimizar processos no mundo real.

Restrições

Condições que limitam ou orientam a solução de um problema de otimização.

Exemplo: Em um problema de maximização de vendas, o estoque disponível pode ser uma restrição, pois impede vendas além da quantidade existente.

Problema de otimização

Problema matemático que busca encontrar a melhor solução possível dentro de um conjunto de alternativas viáveis, maximizando ou minimizando uma determinada função objetivo, sujeita a restrições. No contexto da alocação de horários e professores, envolve a distribuição eficiente de recursos (salas, professores e horários) considerando restrições acadêmicas e institucionais.

Turno

Valor numérico referente a um turno da semana, contido em [0, 15]. O valor é incremental com base no dia/horário, considerando os dias de segunda a sexta nos turnos Manhã, Tarde e Noite. **Exemplo:** Segunda-Manhã -> 0, Segunda-Tarde->1, Segunda-Noite->2, Terça-Manhã->3...

4. Elicitação de Requisitos

4.1. Técnica 1: Entrevista

Descrição: A primeira técnica adotada foi a realização de entrevistas com especialistas em alocação acadêmica e otimização de recursos. Foram consultados membros da equipe do Sistema de Apoio à Decisão (SACI) do Centro de Informática (CI) da UFPB, um ex-coordenador do curso de Ciência da Computação e uma desenvolvedora com experiência na organização de escalas de trabalho. A escolha dessa abordagem se deve ao fato de que esses profissionais possuem experiência direta com sistemas de alocação e otimização, fornecendo insights valiosos para o desenvolvimento do nosso projeto.

Detalhamento: As entrevistas ocorreram nos dias 6 e 7 de fevereiro de 2025, no Centro de Informática da UFPB. O objetivo principal foi entender as diferentes estratégias e desafios envolvidos na alocação de disciplinas, professores e salas, identificando práticas eficazes que possam ser aplicadas ao nosso sistema. Durante as discussões, foram abordadas funcionalidades, limitações e boas práticas que contribuíram para refinar a definição dos requisitos do projeto.

Resultados das entrevistas:

- **Entrevista com Gustavo Motta** (06/02/25, 12h): O professor destacou a necessidade de respeitar os horários disponíveis dos professores e garantir um mecanismo de revisão iterativa da alocação.
- **Entrevista com Cecília** (06/02/25, 16h): A desenvolvedora ressaltou a importância da integração com dados prévios e da personalização de restrições no sistema.

Os documentos completos das entrevistas estão disponíveis para consulta e serviram como base para a definição dos requisitos do projeto.

4.2. Técnica 2: Brainstorming

Descrição:

A segunda técnica utilizada foi o brainstorming, realizado com a equipe interna do projeto. Essa abordagem foi escolhida por sua natureza colaborativa, permitindo que todos os membros compartilhassem suas ideias sobre as funcionalidades e melhorias que a aplicação poderia trazer. O

brainstorming ajudou a identificar diferentes perspectivas e a gerar ideias inovadoras para o sistema.

Detalhamento:

A sessão de brainstorming ocorreu no dia 15 de janeiro de 2025, às 10h, em uma reunião online via Google Meet. Participaram do encontro todos os membros da equipe do projeto, composta por estudantes da disciplina, incluindo desenvolvedores, designers e analistas de requisitos. O objetivo principal da discussão foi levantar as principais funcionalidades que o sistema deveria possuir e explorar soluções para os desafios identificados.

Inicialmente, o foco do projeto era oferecer um ambiente otimizado para desenvolvedores especializados, permitindo que rodassem modelos matemáticos sem se preocupar com a configuração do servidor ou com limitações de memória. No entanto, durante a reunião, percebeu-se que esse público-alvo já possuía conhecimento técnico suficiente para configurar suas próprias soluções, reduzindo o impacto da aplicação para esse grupo.

A partir dessa reflexão, a equipe redefiniu o escopo do projeto para priorizar a acessibilidade. A nova abordagem buscou tornar a alocação de pessoal mais intuitiva para usuários sem experiência com pesquisa operacional, oferecendo uma solução simplificada para a gestão de escalas de trabalho. No entanto, ao avaliar a viabilidade dessa abordagem, identificou-se que seria desafiador desenvolver uma aplicação completamente funcional e prática que atendesse simultaneamente a diferentes grupos com necessidades variadas.

Diante disso, a equipe decidiu focar em um público-alvo mais específico: o contexto escolar. Esse direcionamento permitiu adaptar a aplicação para um cenário com regras bem definidas, facilitando tanto o desenvolvimento quanto a adoção do sistema. A decisão foi consolidada em uma reunião posterior, também realizada via Google Meet, onde foram alinhadas as direções estratégicas do projeto.

Os insights gerados durante essas sessões foram documentados, analisados e refinados para garantir que o sistema atendesse às necessidades do novo público-alvo, priorizando usabilidade e eficiência na alocação de professores e turmas.

4.3. Técnica 3: Levantamento de domínio

Cenário 1: Alocação de horários/professores na educação

No caso de um sistema com escopo acadêmico, foi considerado importante desenvolver uma solução

para a alocação de disciplinas e professores a horários, com base no conhecimento acumulado pelo Centro de Informática (CI) da UFPB. O CI possui estudos científicos focados em otimização de alocação de recursos em ambientes universitários e escolares, o que viabiliza a aplicação de métodos avançados de otimização no contexto de alocação de horários, sem que o sistema se restrinja a um único centro acadêmico.

O problema de alocação de horários é conhecido na literatura como **University Course Timetabling Problem** e envolve restrições tanto **hard** quanto **soft**. As restrições **hard** garantem a viabilidade da solução gerada e incluem condições como:

- Cada disciplina deve ser alocada a uma quantidade de horas igual à sua carga horária semanal.
- Não pode haver choque de horários entre disciplinas do mesmo semestre, com o mesmo professor ou que sejam alocadas na mesma sala.
- Cada turma deve ser alocada a uma sala que consiga acomodar a quantidade de alunos matriculados.

Já as restrições **soft** visam melhorar a qualidade da alocação, por exemplo:

- Diminuir a quantidade de dias que os alunos/professores precisam ir à faculdade.
- Minimizar a quantidade de horários vazios para os alunos em um mesmo dia.
- Minimizar a alocação de aulas nas sextas-feiras (no caso do Centro de Informática).

Para o sistema funcionar, é necessário coletar informações sobre professores e disciplinas. Essas informações podem ser organizadas em tabelas como:

- **Tabela de professores:**
 - Nome completo, matrícula, horários disponíveis, disciplinas lecionadas, etc.
- **Tabela de disciplinas:**
 - Nome da disciplina, quantidade de créditos, curso(s) a que pertence, turno(s), etc.

O sistema será acessado por coordenadores de cursos, que poderão gerenciar o CRUD de professores, disciplinas e salas de seus respectivos centros. As informações dos coordenadores incluirão nome completo, login, senha e dados do centro ao qual pertencem.

Esse levantamento de domínio foi crucial para garantir que o sistema atendesse às necessidades

específicas da alocação de horários no contexto acadêmico em geral, utilizando o know-how do Centro de Informática da UFPB para a aplicação de técnicas de otimização bem estabelecidas.

4.4. Considerações Finais

Ambas as técnicas foram fundamentais para a definição dos nossos requisitos funcionais. O brainstorming nos permitiu esclarecer e estruturar melhor o sistema que tínhamos em mente, destacando as funcionalidades mais relevantes (com foco nos 20% mais importantes). Já as entrevistas com especialistas em alocação de pessoal, incluindo membros do Sistema de Apoio à Decisão (SACI), um ex-coordenador do curso de Ciência da Computação e um desenvolvedor experiente na organização de escalas de trabalho, trouxeram perspectivas valiosas. Esses encontros nos ajudaram a identificar pontos que não havíamos considerado sozinhos e a refinar quais requisitos deveriam, de fato, compor os 20% mais essenciais.

A partir das entrevistas e do brainstorming, algumas necessidades específicas foram destacadas. O professor Gustavo Motta enfatizou a importância de minimizar mudanças entre períodos letivos e garantir revisões iterativas na alocação. Cecília, desenvolvedora do Log-UFPB, reforçou a necessidade de permitir personalização de restrições e integração com dados prévios. Taylor Cantalice, com experiência na alocação de pessoal em supermercados, evidenciou a importância da estabilidade nas escalas de trabalho e da otimização de custos.

Apesar das particularidades de cada contexto, foram identificados interesses comuns entre as áreas entrevistadas, como a necessidade de uma solução flexível, capaz de respeitar regras de negócios específicas, como horários de trabalho, restrições individuais e a minimização de alterações entre ciclos de planejamento. Além disso, a importância de oferecer mecanismos de revisão iterativa apareceu em todas as áreas, evidenciando que a alocação raramente é um processo definitivo e demanda ajustes constantes.

Com base nas informações coletadas nas entrevistas e brainstorming, a lista a seguir descreve as funcionalidades relevantes para o escopo do sistema, bem como aquelas que foram descartadas por não se alinharem ao objetivo do projeto:

Funcionalidades Relevantes:

- **Flexibilidade na Alocação:** O sistema deve ser flexível o suficiente para lidar com diferentes tipos de alocação de pessoal (professores e disciplinas).
- **Respeito aos horários individuais:** Cada professor deve ser alocado de acordo com sua

disponibilidade de horários.

- **Importação de dados prévios:** O sistema deve ser capaz de integrar informações pré-existentes, como dados de horários anteriores ou registros de alocações passadas.
- **Hierarquização de Restrições (Hard/Soft):** O sistema deve permitir a definição de restrições mais rígidas (hard) e mais flexíveis (soft) para melhorar a alocação.
- **Revisão Iterativa:** A alocação deve ser um processo iterativo, permitindo ajustes constantes conforme surgem novas necessidades ou mudanças.
- **Personalização de Restrições:** O sistema deve permitir a personalização de restrições específicas de cada instituição, como requisitos de horários e disponibilidade dos professores.

Funcionalidades Irrelevantes:

- **Ferramentas para Desenvolvedores:** O foco do sistema não será em atender a desenvolvedores com experiência em modelagem matemática ou a criação de soluções complexas de otimização.
- **Gerenciamento de Desempenho:** O sistema não será utilizado para medir ou gerenciar o desempenho individual dos professores.
- **Integração com Sistemas de Folha de Pagamento:** Aspectos financeiros como integração com a folha de pagamento dos professores foram descartados, pois não são essenciais para o processo de alocação.
- **Priorização de Disciplinas para Professores Concursados:** A priorização de alocação de disciplinas para professores concursados é uma demanda específica que não se encaixa no escopo inicial.
- **Controle de Desempenho ou Avaliação de Professores:** O sistema não incluirá funcionalidades voltadas para avaliar o desempenho dos professores, como integração com sistemas de avaliação de desempenho.

- **Gerenciamento de Comunicação Interna entre Funcionários:** O sistema não incluirá recursos de comunicação interna entre os professores ou entre professores e alunos, como chats ou notificações.

Por outro lado, algumas funcionalidades foram consideradas específicas demais para uma aplicação generalista no contexto escolar, e, portanto, ficaram de fora do escopo inicial. No caso acadêmico, por exemplo, existem demandas como a priorização de disciplinas estratégicas para professores concursados, que embora sejam importantes, não são o foco do sistema proposto. Da mesma forma, regras detalhadas sobre distribuição de folgas entre os professores, como ocorre em outros setores, foram descartadas, já que o objetivo é simplificar a alocação, sem entrar em complexidades que comprometam a usabilidade e a eficácia do sistema.

Além disso, aspectos como gerenciamento de desempenho individual, integração com sistemas de folha de pagamento e comunicação interna entre professores foram descartados, pois extrapolam o objetivo central da aplicação. O foco da solução está na alocação de pessoal com base em restrições predefinidas, garantindo um processo mais eficiente de organização das grades de horários.

Com base nesses insights, funcionalidades como respeito aos horários individuais dos professores, importação de dados prévios e hierarquização de restrições (hard/soft) foram priorizadas. Por outro lado, o brainstorming permitiu descartar funcionalidades que não agregam valor ao público-alvo inicial, como ferramentas avançadas para desenvolvedores com experiência em modelagem matemática.

Desse modo, as técnicas de elicitação foram essenciais para refinar nosso escopo, garantindo um sistema mais eficiente e alinhado com as reais necessidades dos usuários, voltado para um contexto acadêmico de gestão de alocação de professores e turmas.

5. Análise de Requisitos

5.1. Requisitos funcionais

[RF 01] Cadastro de Funcionários Não-Usuários

Descrição: O gestor responsável pela alocação deve poder cadastrar os professores no sistema, incluindo as informações como descritas no *Perfil do Colaborador*. O cadastro deve ser acessível apenas para administradores do sistema.

Casos de uso relacionados: UC01

Prioridade: Essencial

[RF 03] Exportação dos Dados em CSV

Descrição: O usuário deve conseguir exportar as soluções geradas pelo sistema para uma planilha no formato CSV, permitindo a gestão das informações fora da aplicação. A exportação deve incluir, para cada professor, as turmas e horários atribuídos.

Casos de uso relacionados: UC03

Prioridade: Essencial

[RF 04] Definir os Horários Possíveis de Cada Professor

Descrição: O gestor deve poder configurar os horários em que cada professor pode lecionar, respeitando restrições previamente definidas. O sistema só poderá alocar um professor em um horário se ele estiver disponível.

Casos de uso relacionados: UC04

Prioridade: Essencial

[RF 06] Tela de Alocação Final

Descrição: O sistema deve apresentar uma tela onde o gestor possa visualizar a distribuição dos professores entre as disciplinas.

Casos de uso relacionados: ainda não preenchido

Prioridade: Essencial

[RF 07] Funcionalidade de Busca de Funcionários

Descrição: O sistema deve oferecer um mecanismo de busca para localizar professores cadastrados rapidamente. A busca pode ser pelo identificador numérico como descrito em *Perfil do Colaborador*.

Casos de uso relacionados: ainda não preenchido

Prioridade: Essencial

[RF 08] Geração de Relatórios

Descrição: O sistema deve permitir a geração de relatórios com dados sobre a alocação, incluindo as alocações dos professores.

Casos de uso relacionados: ainda não preenchido

Prioridade: Essencial

[RF 09] Registro de Histórico de Alocações

Descrição: O sistema deve armazenar o histórico das alocações anteriores, permitindo que o gestor consulte e recupere distribuições passadas. Deve ser possível visualizar alterações feitas ao longo do tempo e restaurar versões anteriores da alocação.

Casos de uso relacionados: ainda não preenchido

Prioridade: Desejável

[RF 10] Cadastro de Disciplinas

Descrição: O gestor responsável pela alocação deve poder cadastrar as disciplinas no sistema, incluindo informações como nome e turnos. O cadastro deve ser acessível apenas para administradores do sistema.

Casos de uso relacionados: UC07

Prioridade: Essencial

[RF 11] Funcionalidade de Busca de Disciplinas

Descrição: O sistema deve oferecer um mecanismo de busca para localizar disciplinas cadastradas rapidamente. A busca pode ser feita por nome ou turno. Deve incluir filtros para facilitar a organização da

alocação.

Casos de uso relacionados: ainda não preenchido

Prioridade: Essencial

5.2. Requisitos não funcionais

Acessibilidade

[RNF 01] O Sistema Deve Ser Acessível Via Navegador

Descrição: O sistema deve ser acessível de forma online, via navegador. Mais especificamente, o sistema terá suporte à versão do Google Chrome mais recente no momento da escrita deste documento (Versão 132.0.6834.160 64 bits).

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

[RNF 02] Acesso a Servidor Online

Descrição: Para garantir a acessibilidade e permitir que todos os usuários rodem a aplicação, o sistema deve utilizar um servidor online. Inicialmente, esse servidor será fornecido pelos próprios desenvolvedores, com 32GB de memória RAM e processador i5-11400H, mas ao final do desenvolvimento, espera-se utilizar máquinas fornecidas pela AWS.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

Confiabilidade

[RNF 03] Algoritmo de Solução

Descrição: O sistema deve utilizar a biblioteca **Python-MIP 1.15.0** para resolver os problemas de alocação de horários. A aplicação deve ser capaz de receber os dados fornecidos pelo usuário e convertê-los para um formato adequado ao resolvidor.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

Segurança

[RNF 04] Segurança dos Dados do Usuário Gestor

Descrição: O sistema deve garantir a **confidencialidade, integridade e disponibilidade** dos dados do usuário gestor. Deve implementar mecanismos de **autenticação segura, controle de acesso baseado em permissões, criptografia de dados sensíveis** e **auditoria de ações administrativas**.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

[RNF 05] Proteção Contra Acesso Não Autorizado

Descrição: O sistema deve incluir medidas para evitar acessos não autorizados, como **verificação de sessão, tempo limite de inatividade e proteção contra ataques de injeção de código**.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

Carga e Escalabilidade

[RNF 06] Suporte a Carga Crescente

Descrição: O sistema deve ser capaz de lidar com **um mínimo de 500 usuários simultâneos** na fase inicial. Deve haver suporte para **escalabilidade**, permitindo aumento da capacidade conforme a demanda crescer.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

[RNF 07] Tempo Máximo de Geração de Alocação

Descrição: A geração de uma nova alocação não deve exceder **5 segundos para cenários com até 100 professores e 15 segundos para cenários com até 500 professores**.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

Formato de Dados - Entrada e Saída

[RNF 08] Formato dos Arquivos de Entrada

Descrição: O sistema deve suportar a importação de arquivos no formato CSV, contendo informações sobre disciplinas, professores e seus horários disponíveis.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

[RNF 09] Formato dos Arquivos de Saída

Descrição: O sistema deve permitir a **exportação das soluções geradas no formato CSV**.

Casos de uso relacionados: Ainda não definidos

Prioridade: Essencial

5.3 Casos de uso

Cadastro de Professor (UC01)

Atores: Administrador e professor associado

Pré-condições: Acesso ao servidor, administrador logado no sistema

Fluxo de Execução

1. O administrador abre a tela “**Cadastro de Professores**”
2. O administrador seleciona a opção “**Adicionar Professor**” (S1)
3. O administrador preenche os dados do professor
4. O administrador confirma a solicitação de cadastro
5. O sistema verifica a consistência dos dados (F1)
6. O sistema confirma a aceitação do cadastro
7. O sistema pede para o administrador confirmar o cadastro
8. O administrador confirma o cadastro

Fluxos Alternativos

Dados Inválidos

1. Em (F1), o sistema identifica inconsistência nos dados
2. O sistema exibe uma tela informando a inconsistência ao administrador
3. O administrador fecha a tela de aviso
4. O administrador volta para (S1)

Professor Já Cadastrado

1. Em (F1), o sistema identifica que o professor já foi cadastrado
2. O sistema exibe uma tela informando que o professor já está cadastrado
3. O administrador fecha a tela de aviso
4. O administrador volta para (S1)

Pós-condições

O professor está cadastrado no sistema e disponível para alocação de aulas.

Gerar Alocação de Aulas (UC03)

Atores: Administrador e professor associado

Pré-condições: Acesso ao servidor, administrador logado no sistema, dados de professores e horários corretamente preenchidos

Fluxo de Execução

1. O administrador acessa a **tela principal** (F3)
2. O administrador seleciona a opção “**Gerar Alocação de Aulas**”
3. O sistema gera uma solução baseada nos dados fornecidos pelo administrador (F4)
4. O sistema exporta os dados no formato **Excel**, contendo o nome do professor e os horários alocados
5. O sistema baixa o arquivo **Excel** para o administrador

Fluxos Alternativos

Não Há Solução Viável

1. Após (F4), o sistema não consegue gerar uma solução viável devido às restrições impostas pelos dados
2. O sistema exibe uma tela informando que **não foi possível encontrar uma solução** com os dados fornecidos
3. O administrador fecha a tela de aviso
4. O administrador volta para (F3)

Dados Faltantes

1. Após (F4), o sistema reconhece que informações necessárias para gerar a alocação não foram preenchidas
2. O sistema exibe uma tela informando **quais dados precisam ser preenchidos**
3. O administrador fecha a tela de aviso
4. O administrador volta para (F3)

Pós-condições

Após a execução do fluxo, o administrador deve ter acesso ao **arquivo Excel** com a alocação de professores ou estar informado sobre quais dados precisam ser corrigidos.

Definir Horários Disponíveis para Professores (UC04)

Atores: Administrador e professor associado

Pré-condições: Acesso ao servidor, administrador logado no sistema, pelo menos um professor cadastrado

Fluxo de Execução

1. O administrador abre a tela **“Cadastro de Professores”** (F5)
2. O administrador seleciona um professor da **lista de professores** (F6)
3. O sistema exibe uma tela com **checkboxes** representando cada horário de aula disponível (F7)
4. O administrador seleciona os horários de disponibilidade do professor
5. O sistema solicita a confirmação da seleção
6. O administrador confirma a seleção
7. O sistema atualiza os dados do professor

Fluxos Alternativos

Nenhum Horário Selecionado

1. O administrador **não seleciona nenhum horário** antes de confirmar
2. O sistema exibe uma tela de aviso informando que **nenhum horário foi selecionado**
3. O administrador fecha a tela de aviso
4. O administrador volta para (F6)

Pós-condições

Após a execução do fluxo, o sistema deve estar atualizado com **os novos horários de disponibilidade do professor**.

Cadastrar Disciplina (UC07)

Atores: Administrador

Pré-condições: Acesso ao servidor, administrador logado no sistema

Fluxo de Execução

1. O administrador abre a tela “**Cadastro de Disciplinas**” (F5)
2. O administrador informa o nome da disciplina
3. O administrador informa os turnos da disciplina (valores de 0-15 respectivos aos turnos da semana)
4. O administrador confirma o cadastro
5. O sistema atualiza os dados das disciplinas

Fluxos Alternativos

Turno inválido

1. O administrador informa um turno fora do espaço de 0-15
2. O sistema exibe uma tela de aviso informando que **aquele turno é inválido**
3. O administrador fecha a tela de aviso
4. O administrador volta para (F5)

Pós-condições

Após a execução do fluxo, o sistema deve estar atualizado com **a nova disciplina e seus respectivos turnos**

ANEXO A

Entrevista com Gustavo Motta

O professor entrevistado foi **Gustavo Motta**, ex-coordenador do curso de Ciência da Computação. A seguir, estão os principais pontos levantados na entrevista:

1. Solicitação de Alocação

- A alocação de professores e turmas ocorre na forma de **(professor x turma) para um determinado horário**.

2. Inércia na Alocação

- A cada período letivo, é necessário refazer o **scheduling** dos professores e turmas.
- O ideal seria **aproveitar a alocação do período anterior**, realizando o menor número possível de mudanças.

3. Respeito aos Horários dos Professores

- É essencial garantir que **os horários disponíveis dos professores sejam respeitados** ao gerar a alocação.

4. Mapeamento de Conflitos

- O sistema deve ser capaz de **identificar potenciais conflitos na alocação**, permitindo que o professor realize ajustes.
- Caso uma solução inviável seja gerada, o sistema deve informar **exatamente o motivo da inviabilidade**.

5. Capacidade de Revisão da Alocação

- A alocação de pessoal é um **processo iterativo**.
- Muitas vezes, um coordenador elabora um scheduling adequado, mas o departamento pode rejeitá-lo e solicitar alterações.
- Em alguns casos, professores de outros departamentos estão envolvidos, exigindo aprovação de setores como o **Departamento de Matemática**.

6. Minimização de Deslocamento

- O sistema deve buscar soluções que **minimizem o deslocamento** dos professores e alunos.
- Exemplo: se um professor ministra duas aulas no mesmo dia, é preferível que **ambas ocorram no mesmo turno** (manhã ou tarde), ao invés de serem distribuídas ao longo do dia.

7. Preferências dos Alunos

- Alguns alunos possuem **necessidades específicas ou turmas preferenciais**.
- É necessário estudar como essa demanda pode ser transformada em um **requisito funcional** para o sistema.

ANEXO B

Resumo da Entrevista com Cecília (Desenvolvedora Log - UFPB)

Nosso projeto (interface para alocação de pessoal em múltiplos contextos) possui pontos em comum com o trabalho desenvolvido por **Cecília**, que visa **alocar professores em salas de aula** para o Departamento de Engenharia de Produção da UFPB. A seguir, os principais insights obtidos na entrevista:

Principais Requisitos e Diferenciais do Sistema

1. Integração com Dados Prévios

- O sistema deve permitir o **uso de respostas anteriores**, importando dados de **planilhas ou PDFs** já existentes.

2. Personalização das Restrições

- **Checkbox para seleção:** Interface com opções para **marcar restrições desejadas**, como tempo de descanso ou dias mínimos de trabalho.
- **Respeito às individualidades:** O sistema deve considerar **diferenças e impossibilidades específicas** de cada professor, como questões de saúde e disponibilidade.

3. Hierarquia de Prioridades

- **Restrições HARD (obrigatórias):** Como obrigações contratuais ou limitações físicas que devem ser respeitadas.
- **Restrições SOFT (preferências):** Como **minimizar dias de trabalho** ou priorizar determinadas salas.
- **Preferência para professores concursados:** Os professores concursados devem **ter prioridade na alocação** de disciplinas estratégicas.

4. Funcionalidades Práticas

- **Exportação em Excel:** O sistema deve garantir **compatibilidade com planilhas**, facilitando a gestão.
- **Tempo de descanso:** Respeitar **intervalos mínimos entre aulas** ou dias de trabalho.
- **Otimização de rotina:** Alguns professores desejam **minimizar a quantidade de dias**

trabalhados, preferindo concentrar aulas em **menos dias**.