

Universidade Federal do Amazonas

Instituto de Ciências Exatas e Tecnologia

Engenharia de Software II

Backlog do Produto

Sistema Web

Arnald Bentes Lucas

José Santo de Moura Neto

Nivaldo Yenar Cidade Rego

Introdução

O sistema proposto é uma ferramenta prática e acessível para ajudar alunos a monitorar seu número de faltas em disciplinas específicas e evitar reprovações por excesso de ausências. Na vida acadêmica, existe o desafio de acompanhar os limites de frequência estipulados, pois perder o controle sobre o número de faltas pode levar a consequências como a reprovação direta na disciplina, independentemente do desempenho acadêmico.

Para simplificar essa tarefa, o sistema foi projetado de forma intuitiva, utilizando como base a carga horária total da disciplina e a quantidade de faltas acumuladas. A partir dessas informações, calcula automaticamente se o aluno ainda possui faltas disponíveis, considerando o limite legal de 25% da carga horária total.

A adição de um retorno visual, em formato de porcentagem facilita o entendimento visual da situação do aluno, enquanto o sistema de login e controle de disciplinas específicas garante personalização e segurança no uso. Essa ferramenta é acessível em qualquer dispositivo e contribui para o planejamento acadêmico mais eficiente.

Requisitos do Sistema

Os requisitos do sistema cobrem tanto funcionalidades essenciais quanto critérios de qualidade da ferramenta. As funcionalidades principais, como a inserção da carga horária total e das faltas acumuladas, são vitais para o cálculo preciso do limite de faltas permitido. Além disso, requisitos não funcionais, como uma interface intuitiva e tempo de resposta rápido, asseguram que o sistema ofereça uma experiência de uso eficiente e agradável, independentemente do dispositivo utilizado.

Requisitos Funcionais

Inserção da Carga Horária Total	
RF01	O sistema deve permitir que o aluno insira a carga horária total da disciplina em horas.
	Prioridade: Alta
Inserção da Quantidade de Faltas	
RF02	O sistema deve permitir que o aluno insira o número de faltas já registradas para a disciplina.

	Prioridade: Alta
Cálculo Automático do Limite de Faltas	
RF03	O sistema deve calcular automaticamente o limite máximo de faltas permitido com base na carga horária inserida, considerando 25% como limite.
	Prioridade: Alta
Cálculo Automático do Limite de Faltas	
RF04	O sistema deve informar ao aluno se ele ainda pode faltar ou não, com base no limite de 25% da carga horária total e no número de faltas já registradas.
	Prioridade: Alta
Barra de Progresso de Faltas	
RF05	O sistema deve exibir uma barra de progresso que mostre visualmente a porcentagem de faltas registradas em relação ao limite permitido.
	Prioridade: Alta
Adição e Remoção de Faltas	
RF06	O sistema deve permitir que o aluno adicione ou remova faltas, ajustando automaticamente o total de faltas e atualizando a barra de progresso.
	Prioridade: Alta
Sistema de Login	
RF07	O sistema deve permitir que o usuário faça login utilizando seu nome de usuário e senha.
	Prioridade: Alta
Controle de Acesso	
RF08	O sistema deve garantir que apenas usuários autenticados tenham acesso ao controle de frequência.
	Prioridade: Alta

Associações de Usuário e Disciplinas	
RF09	O sistema deve permitir que cada usuário controle o registro de faltas para suas disciplinas específicas.
	Prioridade: Alta

Requisitos não funcionais

Interface Intuitiva	
RNF01	O sistema deve apresentar uma interface com os campos e botões bem descritos para fácil entendimento do usuário.
	Prioridade: Média
Tempo de Resposta Rápido	
RNF02	O sistema deve processar as informações e exibir o resultado da verificação em até 1 segundo, proporcionando uma experiência ágil.
	Prioridade: Alta
Compatibilidade com Dispositivos Móveis e Desktop	
RNF03	O sistema deve ser responsivo e funcionar adequadamente em dispositivos móveis e desktop.
	Prioridade: Média
Confiabilidade e Precisão	
RNF04	O sistema deve garantir cálculos precisos do limite de faltas e fornecer uma resposta confiável ao usuário.
	Prioridade: Alta
Precisão na Exibição do Progresso de Faltas	
RNF05	A barra de progresso deve refletir com precisão a porcentagem de faltas em relação ao limite permitido.

	Prioridade: Alta
--	-------------------------

Regras de Negócio

As regras de negócio orientam o cálculo do sistema e definem mensagens de alerta para o usuário caso o limite de faltas seja ultrapassado, garantindo a confiabilidade da informação prestada ao aluno e permitindo o planejamento adequado de frequência na disciplina.

Limite de Faltas Permitidas	
RN01	O limite de faltas permitidas é de 25% da carga horária total da disciplina, conforme a legislação brasileira.
	Prioridade: Alta
Excedência do Limite de Faltas	
RN02	Se o número de faltas registradas ultrapassar o limite de 25%, o sistema deve alertar o aluno com uma mensagem indicando a reprovação por faltas.
	Prioridade: Alta

Telas do Sistema

Tela de Cadastro

GFSsystem

HomeLoginCadastrarAdicionar Matéria

Cadastro

Usuário: Obrigatório. 150 caracteres ou menos.
Letras, números e @./+!- _ apenas.

Senha:

- Sua senha não pode ser muito parecida com o resto das suas informações pessoais.
- Sua senha precisa conter pelo menos 8 caracteres.
- Sua senha não pode ser uma senha comumente utilizada.
- Sua senha não pode ser inteiramente numérica.

Confirmação de senha: Informe a mesma senha informada anteriormente, para verificação.

Cadastrar

Já tem conta? [Login](#)

Bem Vindo

Crie sua conta e gerencie suas faltas e anotações

© 2024 GFSsystem. Todos os direitos reservados.

Tela de Login

GFSsystem

HomeLoginCadastrarAdicionar Matéria

Bem Vindo

Entre e gerencie suas faltas e anotações

Login

Usuário:

Senha:

Login

Ainda não é cadastrado? [Cadastrar](#)

© 2024 GFSsystem. Todos os direitos reservados.

Tela Inicial do Sistema

- Sem matérias cadastradas

GFSsystem

HomeLoginCadastrarAdicionar Matéria

Minhas Matérias

Você ainda não adicionou nenhuma matéria.

Adicionar Matéria

© 2024 GFSsystem. Todos os direitos reservados.

- Com matérias cadastradas

Minhas Matérias

[Finalizar Período](#)

Programação Web

Carga Horária: 60h

Dias: Terça e Quinta

Faltas: 0 0.00%

Pode faltar.

[Editar](#)[Faltou](#)[Remover](#)

Tela de Cadastro de Matérias

Adicionar Matéria

Nome:

Carga horária:

Dias semana:

Faltas:

[Salvar](#)

- Campo para inserir o nome da disciplina
- Campo para inserir a carga horária total
- Campo para informar os dias da semana que ocorre a disciplina
- Campo para inserir a quantidade de faltas
- Botão Salvar

Tela de confirmação de exclusão de disciplina

GFSsystem

Home Login Cadastrar Adicionar Matéria

Remover Matéria

Tem certeza que deseja remover "Programação Web"?

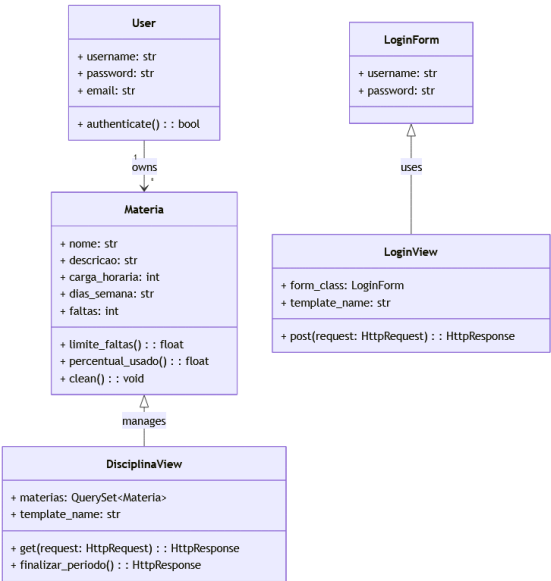
Sim, remover

Cancelar

© 2024 GFSsystem. Todos os direitos reservados.

Diagramas de Classe

Diagrama de classes é uma representação visual da estrutura de um sistema orientado a objetos, mostrando suas classes, atributos, métodos e os relacionamentos entre elas. Ele ajuda a entender como as partes do sistema interagem e organizam-se.



Ferramentas Utilizadas

Django

Django é um framework web robusto e escalável escrito em Python. Ele foi escolhido devido às suas características que facilitam o desenvolvimento rápido e seguro de aplicações web. Alguns recursos importantes que o Django oferece e foram utilizados neste projeto incluem:

- **ORM (Object-Relational Mapping):** Simplifica a interação com o banco de dados utilizando modelos Python.
- **Gerenciamento de Usuários:** Recursos nativos para autenticação e controle de acesso.
- **Segurança:** Proteções contra CSRF, XSS, e SQL Injection embutidas no framework.
- **Admin Interface:** Uma interface administrativa gerada automaticamente para facilitar o gerenciamento de dados do sistema.
- **Escalabilidade:** Preparado para crescer conforme a necessidade, com suporte a ambientes mais complexos.

Python

Python é a linguagem de programação principal do projeto devido à sua simplicidade, legibilidade e vasta gama de bibliotecas e frameworks disponíveis. É amplamente utilizada em aplicações web, ciência de dados, automação e muito mais. Algumas vantagens do Python neste projeto incluem:

- **Comunidade Ativa:** Um ecossistema rico de pacotes e uma grande base de desenvolvedores.
- **Integração com Django:** Python e Django trabalham juntos de maneira fluida para otimizar a produtividade.
- **Bibliotecas Adicionais:** Foram utilizadas bibliotecas nativas e de terceiros para testes, validação e suporte ao desenvolvimento.

Ambiente de Desenvolvimento

- **Visual Studio Code (VS Code):** Escolhido como o editor de código principal, devido à sua leveza e extensibilidade. As principais funcionalidades utilizadas incluem:
 - **Extensões:** Suporte a Django, Python, e formatação automática de código.
 - **Debugging:** Depurador integrado para rastrear erros e monitorar a execução do código.

- **Integração com Git:** Controle de versão diretamente no editor.
- **Terminal Embutido:** Permite executar comandos sem sair do ambiente de desenvolvimento.
- **Git:** Ferramenta de controle de versão usada para rastrear mudanças no código, facilitar a colaboração entre os membros da equipe e permitir o gerenciamento eficiente das versões do projeto.
 - **GitHub:** Repositório remoto utilizado para armazenar e compartilhar o código, além de documentar o progresso do projeto.

Banco de Dados

- **SQLite:**
 - Um banco de dados relacional leve, integrado e pronto para uso com Django, ideal para projetos de pequeno e médio porte. As principais vantagens do SQLite incluem:
 - **Simplicidade:** Nenhuma configuração adicional necessária, ideal para desenvolvimento rápido.
 - **Portabilidade:** O banco de dados é armazenado em um único arquivo, facilitando a migração entre ambientes.
 - **Compatibilidade:** Totalmente suportado pelo ORM do Django.
 - Para futuras expansões, o sistema pode ser facilmente migrado para outros bancos de dados como PostgreSQL ou MySQL, sem alterações significativas no código.

Gerenciamento de Dependências

- **Pip e Virtualenv:**
 - **Pip:** Utilizado para instalar pacotes e dependências do projeto.
 - **Virtualenv:** Ferramenta para isolar o ambiente de desenvolvimento, garantindo que as dependências do projeto não conflitem com outras instalações no sistema.

Outras Ferramentas

- **Postman:** Ferramenta utilizada para testar APIs RESTful e simular requisições HTTP, garantindo a funcionalidade correta das rotas e dados processados no backend.
- **Mermaid:** Utilizado para a criação de diagramas de classe e outros visuais representativos do sistema diretamente no GitHub ou em ferramentas de documentação.
- **Chrome DevTools:** Usado para inspecionar e depurar o frontend do sistema, garantindo responsividade e comportamento esperado nos navegadores.

Melhorias Planejadas

- Adicionar testes automatizados para garantir a confiabilidade do sistema.
- Melhorar a personalização da interface com base no feedback dos usuários.

Modificações Realizadas

- Implementação do sistema de login.
- Controle de usuários para permitir gerenciamento das disciplinas matriculadas.
- Adição de retorno visual em formato de barras indicando a quantidade de aulas já faltadas, comparando com o limite total de faltas permitido.