

# **Lock**



Soluções em Tecnologia

---

## **Especificação de Design de Software**

### **PROLOCK**

LUCAS FERREIRA DE SOUZA SANTOS  
MATHEUS BATISTA SILVA  
PAULO ROBERTO DE JESUS GONÇALVES

**São Mateus - ES**  
**13/03/2022**

## ***Histórico de Revisões***

<b>Versão</b>	<b>Comentário</b>	<b>Data</b>
1.0	Levantamento de requisitos para o desenvolvimento do design do software	28/02/2022
1.1	Readequação das definições de interface.	05/03/2022
1.2	Revisão e adequação com os demais artefatos.	12/03/2022

## **Sumário**

1	Objetivos deste Documento	3
2	Requisitos Tecnológicos e Plataforma de Implementação	3
3	Arquitetura do Sistema	3
3.1	Pacote Gerenciamento do usuário	4
3.2	Pacote GUI	4
3.3	Pacote Gestão de Funções	5
3.4	Pacote Controle de Informações	6
3.5	Pacote Resultado	6
4	Observações	7

Organização: Lock soluções em tecnologia	Projeto: PROLOCK	
Documento de Especificação de Design de Software	Versão: 1.0	Data: 13/03/2022

## 1. Objetivos deste Documento

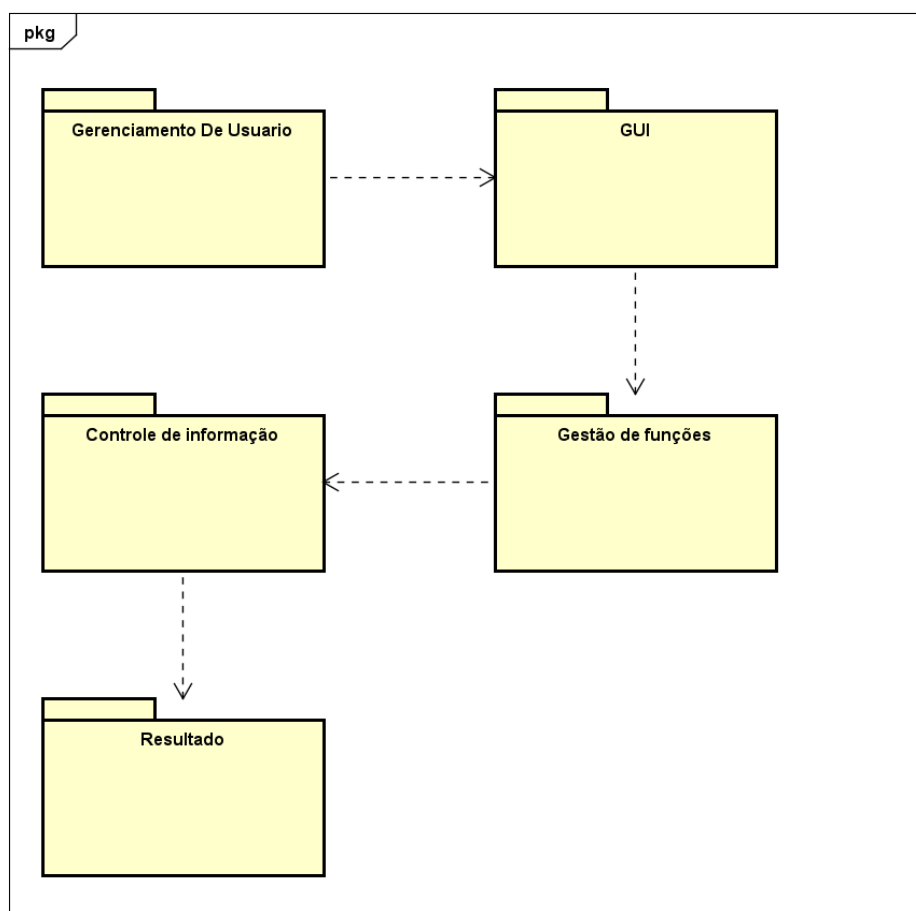
Documento que contempla os requisitos identificados nos modelos de requisitos construídos ao longo do projeto. Auxilia quem ficará responsável por testar, criar e manter nosso software. Retrata aspectos de todas funções, assim como a interface, e compreende os detalhes comportamentais e de dados, seguindo a perspectiva dos implementadores de forma que seja transparente e instrutivo para a compreensão do leitor.

## 2. Requisitos Tecnológicos e Plataforma de Implementação

Como já foi dito anteriormente, o Software será uma plataforma web desenvolvida em Python 3, usando o microframework Flask, para o back-end e para o sistema de alocação dos professores. E HTML5, CSS3 e BootStrap para o front-end. O layout do front-end irá se constituir em uma página de login, após fazer o login você terá a opção de cadastrar, alterar, excluir um professor ou disciplina. Também terá a opção de alocar as disciplinas aos professores. Iremos utilizar um banco de dados relacional, não iremos utilizar um volume muito grande de informações, então um banco de dados relacional é a melhor opção para esse projeto. O MySQL foi a escolha. O sistema ficará hospedado no servidor interno da instituição.

## 3. Arquitetura do Sistema

Usando um pacote de Gerenciamento de Usuário que engloba as classes: Pessoa, Cadastro, Administrador e Login, onde há o tratamento de dados do usuário, que será ligado por meio do pacote de interface GUI aos pacotes gestão de funções, controle de informação e Resultado.



Organização: Lock soluções em tecnologia	Projeto: PROLOCK
Documento de Especificação de Design de Software	Versão: 1.0
	Data: 13/03/2022

### 3.1. Pacote Gerenciamento de usuário.

Com base no diagrama de classes e de pacote apresentados, traremos uma concepção mais vasta e completa do tema, após isto é feito uma descrição da especificação do pacote de gerenciamento de usuário, sendo que os métodos e atributos ali envolvidos estão diretamente ligados a dados comumente necessários para realizar registro de um usuário e sua autenticação de acesso.

A promoção da interface é no intuito de contribuir com a experiência de exercitação do usuário. Sua implantação simplifica os processos tornando a assimilação da ideia muito mais eficaz assim como agilizando os métodos nas ocorrências de execução.

Classe: Gerenciamento de usuário		
Objetivo	Informações básicas ao administrador, que são exigências obrigatórias para a efetivação do cadastro. Procedimentos recorrentes a um ingressante que busca se cadastrar no sistema, elaborando uma senha de segurança única e confidencial, e também um “logout” por meio da numeração de CPF, a senha prescrita pelo usuário e um verificador de autenticação., para validar a entrada.	
Descrição de Atributos		
nome	String	Nome do administrador.
email	String	endereço de email do administrador.
telefone	String	Número do telefone para contato com o Administrador.
DataNascimento	String	Data de nascimento do Administrador.
id	Int	Identificador do Administrador.
senha	String	Senha individual, definida por um Administrador para acessar o sistema.
RG	String	Documento pessoal de identificação oficial do usuário.
CPF	String	Documento de identificação de contribuinte, emitido pela Receita Federal.
Descrição de Operações		
setSenha()	Método utilizado para cadastrar senha do Administrador no sistema, permitindo assim o seu acesso ao sistema.	
setSenha(senha: Administrador)	Método que requer a inserção da senha cadastrada pelo Administrador.	
setCPF(CPF: Administrador)	Método que requer a inserção do CPF cadastrado pelo Administrador.	
ValidarAutenticação()	Método que o sistema utiliza para comparar os dados inseridos com os dados previamente dispostos na base de dados, liberando o acesso ou requisitando novamente a digitação.	

### 3.2. Pacote GUI:

Classe: GUI		
Objetivo	As classes de interface são usadas para gerar a conexão entre o usuário e o código. Sem o GUI, seria praticamente impossível o manuseio por todos os usuários, já que nem todos teriam o entendimento de linhas de código.	
Descrição de Atributos		
Não há atributos.	—	—
Descrição de Operações		
Sem operações.	—	

Organização: Lock soluções em tecnologia	Projeto: PROLOCK	
Documento de Especificação de Design de Software	Versão: 1.0	Data: 13/03/2022

### 3.3. Pacote Gestão de Funções:

Classe: Gestão de Funções		
Objetivo	Pacote responsável por gerir as funções dos sistemas após o login do administrador. As funções que o sistema oferece são de atualizar os dados de um professor ou disciplina, excluir um professor ou disciplina e adicionar professor e disciplina. Essas alterações são feitas diretamente na base de dados. Além dessas funções, temos a classe de alocação dos professores.	
Descrição de Atributos		
codigo_professor	Int	Variável responsável por receber o código do professor para posterior busca na base de dados.
nome_professo	String	Variável que recebe o nome do professor para posterior cadastro na base de dados.
demanda_professor	Int	Variável que recebe a demanda do professor para posterior cadastro na base de dados (usada somente se necessário).
area_de_formacao	String	Variável que recebe a área de formação do professor para posterior cadastro na base de dados (usada somente se necessário).
Restricao	Boolean	Variável que identifica se o professor possui alguma restrição, independente de qual seja. Caso possua alguma, será considerado no momento da alocação.
codigo_disciplina	Int	Variável responsável por receber o código da disciplina para busca na base de dados.
nome_disciplina	String	Variável que recebe o nome da disciplina para posterior inserção na base de dados.
demanda_disciplina	Int	Variável que recebe a demanda da disciplina para posterior cadastro na base de dados.
Descrição de Operações		
Busca_professor()	Método que irá identificar se o professor existe na base de dados, importante pois não se deve ter duplicatas no registro.	
Busca_disciplina	Método que irá identificar se a disciplina existe na base de dados, importante pois não se deve ter duplicatas no registro.	
insere_professor()	Método que irá inserir na base de dados o professor, após fazer uma busca e identificar que o professor já não é cadastrado ele pode ser inserido sem problemas na base de dados.	
insere_disciplinas()	Método que irá inserir na base de dados a disciplina, após fazer uma busca e identificar que a disciplina já não é cadastrada ele pode ser inserido sem problemas na base de dados.	
Exclui_professor()	Método que exclui o professor da base de dados. Após verificar que o professor existe na base de dados com o método de busca ele será excluído. Caso não exista o programa retorna uma mensagem de aviso.	
Exclui_disciplina()	Método que exclui a disciplina da base de dados. Após verificar que a disciplina existe na base de dados com o método de busca ele será excluído. Caso não exista o programa retorna uma mensagem de aviso.	
edita_professor()	Método que irá editar os dados de um professor na base de dados. Após verificar que o professor é cadastrado pode-se editar o campo desejado. Caso o professor não exista o programa retorna um aviso.	
edita_disciplina()	Método que irá editar os dados de uma disciplina na base de dados. Após verificar que a disciplina é cadastrada pode-se editar o campo desejado. Caso o professor não exista o programa retorna um aviso.	

Organização: Lock soluções em tecnologia	Projeto: PROLOCK
Documento de Especificação de Design de Software	Versão: 1.0
	Data: 13/03/2022

### 3.4. Pacote Controle de informação:

Classe: Controle de Informação	
<b>Objetivo</b>	Classe responsável pela alocação temporária de dados sobre professores e disciplinas, assim como as restrições associadas.
<b>Descrição de Atributos</b>	
Não há atributos.	—
<b>Descrição de Operações</b>	
Alocar() : void	Esse método permite acessar os atributos e métodos de outras classes para realizar o processamento dos dados utilizando os recursos da biblioteca de otimização selecionada e implementada pelos desenvolvedores do projeto. A otimização considera as variáveis e restrições associadas gera o melhor caso como saída.

### 3.5. Pacote Resultado:

Classe: Resultado		
Objetivo	Após a execução da função de Alocação, aqui será exibido o resultado desse processo em forma de tabela, relacionando os professores com suas respectivas disciplinas alocadas.	
Descrição de Atributos		
codigo_professor	int	Variável que recebe o código do professor alocado.
codigo_disciplina	int	Variável que recebe o código da disciplina alocada.
nome_professor	String	Variável que recebe o nome do professor alocado
nome_disciplina	String	Variável que recebe o nome da disciplina alocada.
Descrição de Operações		
Resultado()	Fará a captura desses dados na respectiva tabela da base de dados, mostrando os dados que foram alocados de forma clara, para que o usuário possa analisar os dados.	

Classe: Alocação	
<b>Objetivo</b>	Classe responsável pela alocação temporária de dados sobre professores e disciplinas, assim como as restrições associadas.
<b>Descrição de Atributos</b>	
Não há atributos.	—
<b>Descrição de Operações</b>	
Alocar() : void	Esse método permite acessar os atributos e métodos de outras classes para realizar o processamento dos dados utilizando os recursos da biblioteca de otimização selecionada e implementada pelos desenvolvedores do projeto. A otimização considera as variáveis e restrições associadas gera o melhor caso como saída.

Organização: Lock soluções em tecnologia	Projeto: PROLOCK
Documento de Especificação de Design de Software	Versão: 1.0
	Data: 13/03/2022

Classe: Cadastro		
Objetivo	Classe responsável pelo cadastro de usuários Administradores no sistema. A classe invoca os atributos da classe pessoa.	
Descrição de Atributos		
DadosPessoa	pessoa	invoca os atributos da classe pessoa.
Descrição de Operações		
setSenha()	Captura as variáveis da classe.	

## 4. Observações

Para guiar de forma instrutiva o usuário a respeito do manuseio do programa, será feita também a adição de um item denominado "Ajuda", onde ele poderá sanar as dúvidas sobre como proceder por meio de textualização.

Há também a possibilidade do usuário experimentar um tutorial de como usar o programa, com um exemplo de gravação de tela feito por um membro da equipe, neste exemplo iremos explicar os ícones e como eles atuam no programa.

Ao fim do uso da execução do programa será feito um pedido na interface perguntando ao usuário qual a avaliação dele a respeito da experiência, e lhe pedir sugestões de como está evoluindo o sistema.