

PUC MINAS

1a Avaliação de ENGENHARIA DE SOFTWARE II - VALOR: 25 DATA: 03/04/2024

21,5

NOME:

Caio Gomes Alcântara Glória

A parte de gerenciamento de acesso foi desenvolvida por uma empresa para um grande projeto. No gerenciamento de acesso o usuário é cadastrado pelo gerente e o gerente associa um perfil de acesso do usuário para as funcionalidades do projeto. Cada funcionalidade pode ter os seguintes perfis: **apenas leitura**; **acesso completo**; **apenas alteração**. Uma funcionalidade pode possuir 1 ou muitos serviços associados. Sendo o protótipo abaixo referente ao cadastro de usuários:

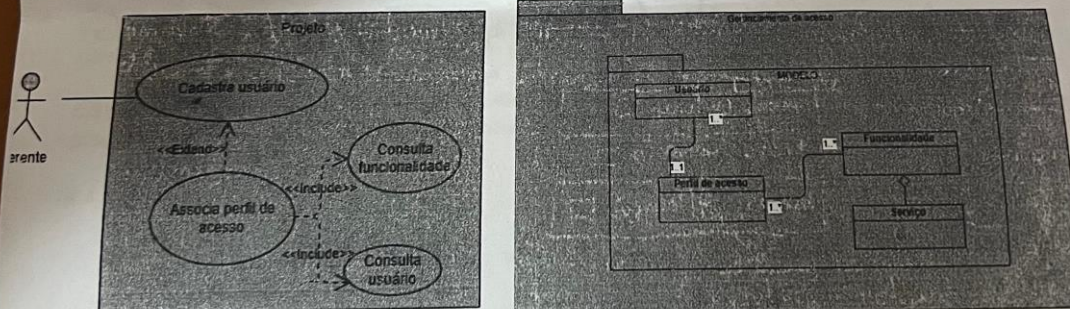
**Cadastro de Usuário**

digite o nome ...   Consultar

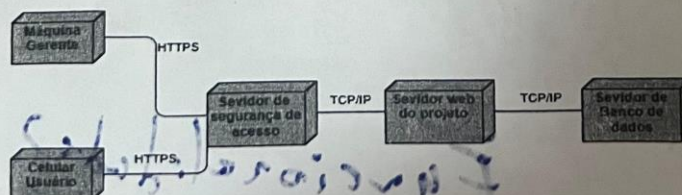
#	Nome	Telefone	E-mail	Ação
1	John	(31) 999-8877	jj@gmail.com	<input type="button" value="Excluir"/>
2	Mary	(31) 998-8767	mary@cad.com	<input type="button" value="Excluir"/>
3	Doreen	(31) 987-4433	doreen@edu.gov	<input type="button" value="Excluir"/>

- 1) Quais são os atributos e métodos da classe: **Usuário**? Qual o tipo de cada atributo? Quais os parâmetros e retornos de cada método?

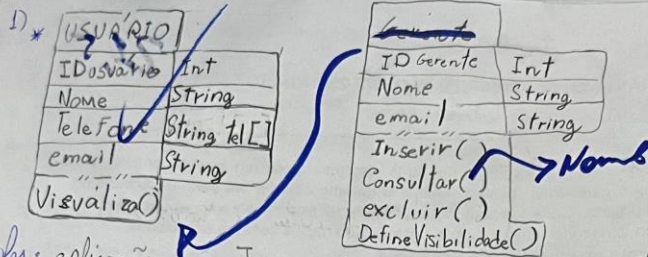
Sendo os diagramas abaixo:



- 2) Faça o protótipo referente à **associação de perfil de acesso** pelo gerente para as funcionalidades do projeto. Os atributos das classes de **Perfil** e de **Funcionalidade** você pode acrescentar o que achar necessário na(s) sua(s) tela(s).
- 3) Faça o diagrama de robustez para o protótipo que você fizer.
- Sabendo-se que temos o seguinte diagrama de implantação/execução:

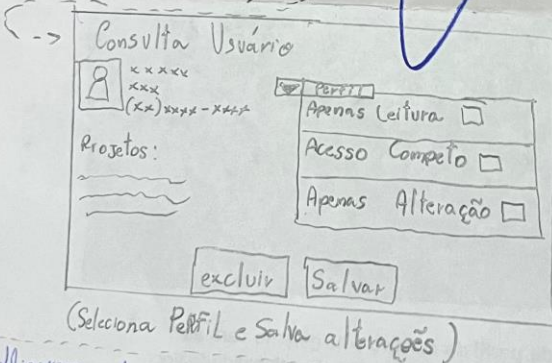
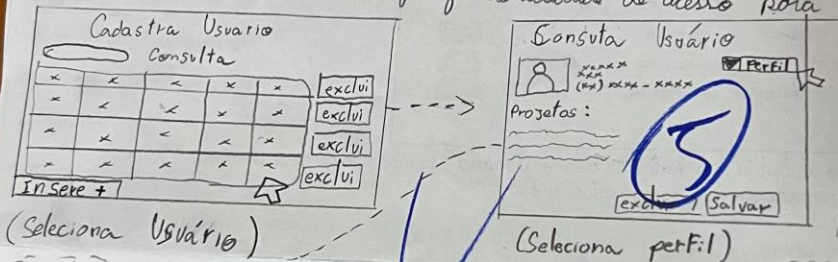


- 4) Faça o diagrama de componentes para os componentes contidos no servidor de segurança de acesso e no servidor web.

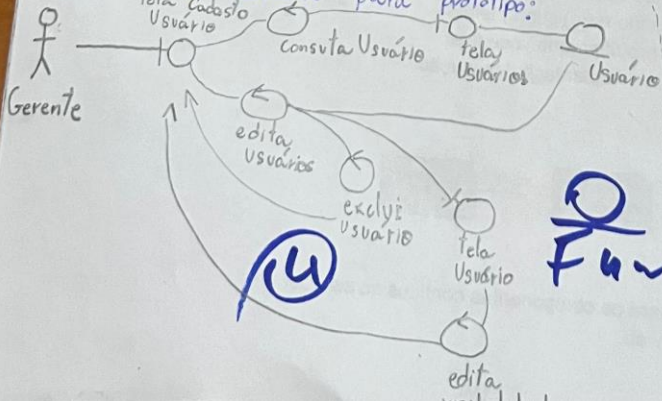


Obs: aplicação em Java, "telefone" é um vetor de Strings para que o ID usuário seja a posição do telefone do mesmo no vetor.

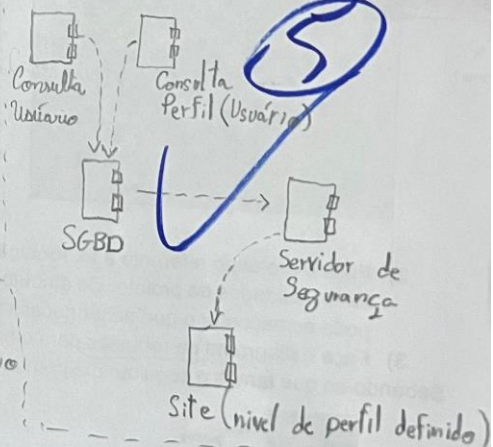
2) Protótipo de tela: Gerente define funcionalidade de acesso para Usuário Cadastrado



3) Diagrama de Robustez para protótipo:



4) Diagrama de Componentes na hora de acessar Servidor!



Funcionalidade?



5) Seja os seguintes elementos da UML (Processo unificado) na definição de interfaces com usuários de um projeto de software:

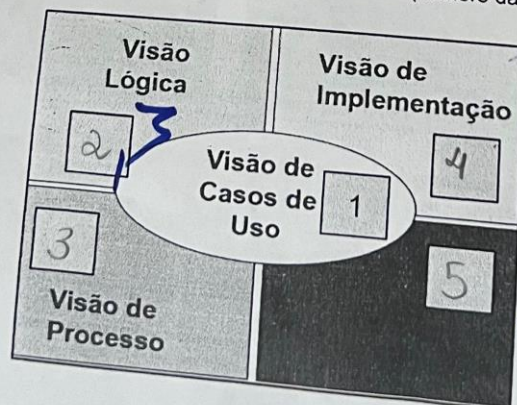
1	2	3	4	5
Arquitetura Física <i>Lógica</i>	Arquitetura Lógica	Arquitetura Física <i>Lógica</i>	Arquitetura Física	Arquitetura Lógica <i>Física</i>
1) Criamos os casos de uso	2) Para cada interface com os usuários identificada no diagrama de robustez	3) Fazemos projeto detalhado com classes e pacotes (neste caso implementou em java)	4) A partir dos pacotes desenhamos os componentes	5) E finalmente identificamos os servidores de implementação dos componentes

5.1) A ordem dos diagramas está correta? Caso contrário apresente a melhor ordem.

5.2) A especificação da arquitetura (definida na 2ª linha) para cada etapa está correta? Caso contrário ajuste para melhor identificar cada coluna.

5.3) Na sua opinião, quais dessas colunas são essenciais e quais são dispensáveis em um processo rápido de desenvolvimento de software.

5.4) Numere no quadro abaixo, de acordo com a tabela acima (número da 1ª linha)



4,5

Visão de Instalação/distribuição

5.1) A ordem dos diagramas está coerente, uma vez que os primeiros ajudam a entender e indentificar melhor as relações e interações entre classes e objetos, e os diagramas futuros mostram componentes relacionados entre si, planejamento de servidores para os mesmos e propostas de implementação.

5.2) Existem diagramas com nomenclatura de arquiteturas equívocas.  
(Mudanças serão feitas na própria imagem). marcadas com 'x'.

5.3) Na minha humilde opinião, os mais necessários são (1, 3, 5), diagrama de caso de uso sempre essencial para criar uma base ao projeto e facilita implementações futuras (1), diagrama com classes e pacotes se posicionam na parte de projeto detalhado, cuja é outra etapa fundamental para desenvolver um software (3), e por fim servidores de implementação que irão de fato organizar e separar os componentes para que o sistema tenha melhor desempenho e mais coerência (5).

Agora em minha opinião, o mais "dispensável" é o Diagrama de Robustez (2), pois o mesmo ajuda mais na parte de prototipagem e design já que nele é mostrado telas e suas sequências de ações.

5.4) Na Imagem.

2, 3, 5

OK