

## Situação Problema– CheckList

Curso: **ADS – Análise e Desenvolvimento de Sistema**

Ano atualização: **2023/1**

Nome da Trilha: *Projeto Individual*

### Nome do projeto

Tema

Link para Trello: <após criá-lo, adicione o link aqui>

## CHECKLIST DE ATIVIDADES - Projeto Individual<sup>1</sup>

- ☐ Escolher proposta
- ☐ Descrever Justificativa para o projeto
- ☐ Criar Mapa Mental
- ☐ Descrever Justificativa para o projeto
- ☐ Criar Diagrama de Caso de Uso
- ☐ Criar Diagrama de Sequência
- ☐ Criar Diagrama de Classe
- ☐ Criar Diagrama de Atividades
- ☐ Criar Diagrama de Entidade e Relacionamento
- ☐ Criar a Estrutura de arquivos do Sistema;
- ☐ Descrever proposta de sistema
- ☐ Descrever Regras de negócio
- ☐ Descrever Requisitos Funcionais
- ☐ Desenvolver wireframes (recomendamos o uso do FIGMA)
- ☐ Descrever Requisitos Não Funcionais
- ☐ Gerar Dicionário de Dados

---

<sup>1</sup>A ordem acima pode ser alterada de caso para caso, assim a palavra que vale é a de quem lhe orientar!

**O sistema deve conter no mínimo:**

- ❑ *Página padrão;*
- ❑ *Página de Contato;*
- ❑ *Página de About (quem somos);*
- ❑ *Página de Login;*
- ❑ *Página de gerenciamento (Para administrar as informações mostradas na pagina padrão e também de usuários);*
  - ◆ *Consultar;*
  - ◆ *Adicionar;*
  - ◆ *Alterar; ou*
  - ◆ *Excluir.*

Cabe ressaltar que cada projeto terá suas especificidades as quais serão adaptativas na análise de software podendo ter itens complementares.

Ressaltamos que não se deve pensar em algo muito grande para que não demande muito tempo e recursos no desenvolvimento desse primeiro projeto.

## **CRIAR QUADRO NO TRELLO**

### **RESULTADO FINAL:**

link do quadro do Trello na página inicial do projeto

### **CONTEXTUALIZANDO:**

Primeiramente, você deverá fazer um fichamento sobre:

- O que são os métodos ágeis e o que pregam?
- Kanban
  - O que é?
  - Quando surgiu e por quê?
  - Imagens que exemplificam o funcionamento.
  - Empresas que utilizam este modelo.

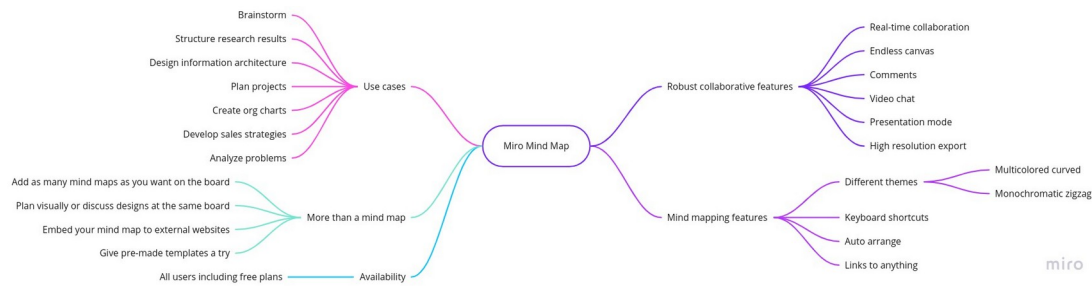
Depois de validado o fichamento, você deve buscar no AVA pelo documento chamado “Trello - Gestão de Atividades”, e seguir suas orientações.

Assim que finalizar o que é indicado em “Trello - Gestão de Atividades”, siga as seguintes orientações:

- Gere o link compartilhável da sua cópia deste template (“Compartilhar” > “Copiar link”). No Trello, em “Menu > Sobre este quadro”, cole o link. Assim, teremos acesso mais fácil a este arquivo a partir de seu Trello quando quisermos ver algo, já que seu quadro foi compartilhado conosco (Foi, né? Você fez o que pedimos no outro arquivo né? rs).
- Crie um cartão para cada atividade que está no Checklist (1ª página deste documento) de “Criar Mapa Mental” até “Gerar Dicionário de Dados” (as demais etapas não seguem a mesma linha destes, portanto devem ser criados apenas quando você chegar nelas).
- Crie o *checklist* nos cartões das atividades que irá desenvolver de hoje até o final dessa semana (se for quinta ou sexta, já faça da próxima semana também).
- Mova a primeira atividade, “Criar Mapa Mental”, para a lista “Fazendo”.
- Insira o link do seu Trello na primeira página deste documento, no local destinado (logo acima do checklist).
- Exclua essa seção deste documento e siga em frente :)

## MAPA MENTAL

### RESULTADO FINAL:



Exemplo do Mapa Mental

Site Recomendado: <https://miro.com/index/> (existem outros caso queira utilizar)

### CONTEXTUALIZANDO:

Primeiramente, você deverá fazer um fichamento sobre Mapa Mental, abordando:

- ✓ O que é Mapa Mental
- ✓ Como fazê-lo

Depois de validado o fichamento, você deve criar um Mapa Mental, abordando o que você precisa conhecer sobre a empresa para a qual está desenvolvendo seu projeto para que o cliente seja atendido da melhor maneira possível. Veja bem: como deve ter compreendido com sua pesquisa, um mapa mental pode auxiliar em várias situações diferentes, mas nessa seu foco deve ser o que você **PRECISA CONHECER** sobre o funcionamento atual da empresa. Pense como se você estivesse se preparando para uma entrevista com o cliente que lhe pediu o software (o dono da empresa) e precisasse definir o que você não pode sair da entrevista sem saber como algo acontece HOJE EM DIA na empresa. A principal intenção deste artefato será lhe auxiliar a ter compreensão dos pontos que deverão ser abordados na ferramenta seguinte, afinal, em um projeto de um software, todas as ferramentas são interligadas e interdependentes.

**Ação:** Realizar o mapa mental para execução do seu projeto.

## LEVANTAMENTO DA SITUAÇÃO-PROBLEMA

### RESULTADO FINAL:

Apresentar texto descritivo da situação-problema

### CONTEXTUALIZANDO:

Primeiramente, é importante destacar que situação-problema é bem diferente de problema. Os textos que apresentamos propostas de projeto são resumos de situações-problema, pois **descrevem o que acontece atualmente na empresa em um contexto global, abordando o funcionamento da empresa como um todo, não apenas os “problemas” que lá ocorrem**. Apenas assim, podemos propor algo ao cliente que realmente possa ajudá-lo, pois para entender as causas de um problema, precisamos entender o todo. Sem essa compreensão geral, não se pode chegar a soluções verdadeiramente efetivas (podemos descobrir até que um software não é a necessidade do cliente, mas vocês estão nesse curso por um motivo, então esse não será o caso).

Sabendo disso, seu papel é **detalhar o funcionamento da empresa escolhida na atualidade, ou seja, antes de seu novo software**, usando como base a situação que passamos, mas aprofundando os detalhes de como as coisas acontecem. Nesse momento normalmente fazemos anotações com base em entrevistas ao cliente ou usando informações coletadas de outras formas, mas não temos um cliente real, então vamos dar algumas dicas para ajudar:

- *Pesquise sobre empresas do ramo escolhido* para entender como funcionam;
- *Aproveite seus conhecimentos previamente adquiridos* (também conhecidos como experiência de vida) na área da empresa que escolheu, se houver;
- *Simule uma situação real*. Lembre-se que são propostas com empresas fictícias, sendo assim, você terá que tomar certas decisões sobre como a empresa funciona em relação às coisas que não estão definidas no documento base, então tente “visualizar” a empresa funcionando, como se você estivesse lá acompanhando o dia a dia;
- *Use o Mapa Mental* criado anteriormente, afinal de contas, lá você levantou justamente o que precisa saber sobre a empresa, então agora é hora de esclarecer todos os pontos levantados.

Seguindo essas dicas, você deve ser capaz de descrever o dia a dia da empresa selecionada. E para ajudar na organização do texto, indicamos uma abordagem em 3 etapas:

- **Introdução:** comece com um parágrafo apresentando a empresa (nome, o que faz, tempo de existência, o dono, funcionários, etc);
- **Situação-problema:** essa vai ser a parte realmente longa e detalhada desse artefato de projeto. Aborde em detalhes como ela funciona, procurando seguir uma ordem lógica dos acontecimentos e organizando parágrafos diferentes para cada coisa diferente que for explicar (como faria em uma redação);
- **Conclusão:** tenha um parágrafo de conclusão focando nos problemas que você notou dentro da situação problema analisada (lembre-se que situação-problema é diferente de problema), e aponte brevemente como um software poderia ajudar a resolvê-los.

Além da descrição textual, usaremos uma ferramenta chamada Desenho Rico. No Ava há um artigo sobre ela, então leia-o, tire suas dúvidas com os orientadores.

## DESCRIÇÃO DA PROPOSTA

### RESULTADO FINAL:

Apresentar texto descritivo explicando a proposta de solução

### CONTEXTUALIZANDO:

Após entender a empresa para a qual desenvolverá o software, poderá propor a ela uma solução que será útil nos aspectos de dificuldade encontrados. Assim, aqui você deverá explicar de maneira resumida, e preferencialmente mais textual, como o software funcionará. Pense nesse texto como uma introdução ao seu cliente do que você pretende fazer por ele, para que ele confirme se realmente está dentro do desejado e permita sua continuidade. Pontos importantes a se destacar são:

- Qual o foco de ação do software (ex: o foco de ação do Gmail é permitir o envio e recebimento de e-mails), relacionando com os problemas levantados na análise da situação-problema
- Os níveis de usuário do sistema (vai ser pro dono do negócio? pros funcionários? talvez seja para ambos, ou para funcionários de cargos diferentes...)
- O que poderá ser feito no software (apenas o principal, sem pensar em telas ou detalhes específicos, pois isso será feito em outro momento)
  - Se houver mais de um nível de usuário, ressaltar as diferenças entre eles

Tenha em mente que essa é uma etapa relativamente breve. *Não é necessário um texto gigantesco, apenas dar uma noção do funcionamento do sistema.* Mais adiante precisaremos ser bem detalhistas, todavia agora a intenção é apenas fazer algo que permita ao cliente nos dizer se estamos no caminho certo...

## REGRAS DE NEGÓCIO

### RESULTADO FINAL:

Apresentar regras de negócio definidas, enumeradas e organizadas.

### CONTEXTUALIZANDO:

Primeiramente, você deverá fazer um fichamento sobre Regras de negócio, abordando:

- ✓ Definição
- ✓ Aplicação
- ✓ Importância/impacto da definição de regras de negócio no contexto da análise de sistemas

Se desejar, no Ava há um arquivo que pode lhe ajudar a entender o assunto!

Depois de validado o fichamento, você irá descrever usando a lista abaixo todas as Regras de negócio pertinentes para o desenvolvimento de seu software. Procure organizá-las de alguma maneira lógica, como por afinidade de assunto ou por sequência de acontecimento dentro da empresa. Além disso, tente descrevê-las como mostra a página 44 do arquivo do Ava que citamos acima, ou seja, de maneira mais sucinta e dividida.

- RN1: Digite uma regra de negócio aqui.
- RN2: Digite uma regra de negócio aqui.
- RN3: Digite uma regra de negócio aqui.
- RN4: Digite uma regra de negócio aqui.

RN5: Dê um *enter* no final de um dos marcadores acima. Olha só, foi criado um novo marcador! Legal, né? Assim, caso tenha mais regras, é só ir digitando, e a cada *enter* um novo marcador é criado!

RN6:

## REQUISITOS FUNCIONAIS

### RESULTADO FINAL:

Apresentar requisitos funcionais descritos dentro do padrão orientado

### CONTEXTUALIZANDO:

Primeiramente, você deverá fazer um fichamento abordando:

- ✓ Definição de requisito funcional
- ✓ Definição de requisito não funcional
- ✓ Definição de requisito suplementar ou normativo
- ✓ Técnicas de levantamento de requisitos: tipos
- ✓ Importância/impacto da definição dos requisitos de um sistema

DICA: Entre os vários disponíveis na biblioteca, dois livros que podem ajudar muito na compreensão e definição dos requisitos são:

- ✓ *Princípios de análise e projeto de sistemas com UML*, de Eduardo Bezerra (na seção 2.1.1, iniciada na página 22, são abordados os 3 tipos de requisitos acima) e
- ✓ *Garantia da qualidade de software*, de Alexandre Bartié (na página 110, no tópico 12.1, se abordam categorias de testes, porém a tabela da página 111 é uma ótima ajuda para pensar em requisitos não funcionais, e na 113 em diante se abordam 12 categorias de testes que também podem ser vistas como categorias de requisitos não funcionais, dando ainda mais ideias para seu projeto).

Além deles, o arquivo do Ava que já indicamos para as regras de negócio também tem conteúdos que podem ajudar.

Depois de validado o fichamento, você deverá definir e descrever os Requisitos funcionais de seu projeto, seguindo o formato abaixo:

**Nome do requisito funcional:** descrição do requisito.

**Dados necessários:** dado 1, dado 2, dado 3.

**Usuários:** todos os níveis de usuário.

Explicando o padrão:

- **R.F. XX - Nome do requisito funcional:** meio óbvio, mas é o nome da função que o software terá. Sugerimos, por padronização, que tenha o prefixo R.F. (requisito funcional) seguida da numeração, para melhor identificação do requisito, acrescido do formato “Substantivo + onde será feita a ação”, como:
  - “R.F. 01 - Registro de Funcionários”,
  - “R.F. 15 - Gerenciamento de consultas” ou
  - “R.F. 04 - Débito em conta corrente”.

- Deixe para definir as numerações ao final, tendo em vista que mudanças podem acontecer e não é prático sempre ficar reajustando os números.
- **Descrição do requisito:** local para descrever a função deste requisito. Sempre se preocupe em esclarecer dois pontos: *o que o requisito faz e o motivo de sua existência* (especialmente se a ação executada nesse requisito não for algo que já acontece naturalmente na empresa - como um Registro de funcionários, que talvez não exista hoje mas para o software é necessário para viabilizar uma Autenticação de usuários -, ou se for algo que faz sentido apenas por ser um software - como a própria Autenticação).
- **Dados necessários:** aqui devem ser colocados os nomes dos dados que serão usados para que esse requisito atenda o que precisa fazer. Nas entradas e processos, em geral, são os dados que serão salvos (seja algo digitado pelo usuário ou captado do sistema, como a hora atual). Já nas saídas, são os dados que serão exibidos em tela (sejam eles vindos diretamente do banco, ou criados por um cálculo ou busca na sessão do usuário).
- **Usuários:** informar os usuários que terão acesso ao requisito, haja vista que você pensou neles quando fez a descrição da proposta.

A seguir, um exemplo de requisito funcional (que, por sinal, pode usar em seu projeto, adaptando se necessário):

**R.F. 01 - Autenticação de usuário:** tem como propósito autenticar o acesso ao sistema, verificando se o usuário pode acessá-lo e, caso possa, o direcionando para a página principal de seu perfil de acesso.

**Dados necessários:** login, senha, nível de permissão.

**Usuários:** todos os níveis de usuário.

Além disso, as funcionalidades devem ser organizadas, usando os títulos escritos em verde mais abaixo neste documento, em:

- ***Entradas:*** São as funcionalidades que alimentarão o software com as informações essenciais para seu uso. Exemplos possíveis são: “Registro de usuário” (para permitir depois seu acesso ao software) ou “Registro de paciente” (que seria útil caso nosso software fosse para uma clínica, evitando registrar várias vezes os mesmos dados da pessoa a cada consulta e viabilizando até um histórico de seus atendimentos).
- ***Processos:*** Em geral, englobam toda ação que executa cálculos, processamentos de tomada de decisão ou transforma dados em novos dados. Como exemplo, podemos citar a “Autenticação de usuário”, que usará os dados de “Registro de usuário” em sua execução, ou um “Agendamento de consulta”, que usará dados do “Registro de paciente” e talvez do “Registro de funcionário” em sua execução.
- ***Saídas:*** São os relatórios, gráficos, impressões, enfim, onde forem usados os dados do software para gerar informações pertinentes ao negócio, mas sem intenção de alterá-los, apenas permitindo sua visualização e filtragem. Exemplificando, um “Relatório de consultas por paciente”, um “Relatório de vendas” e um “Log de usuários autenticados” podem ser consideradas saídas, pois usam informações de entradas e processos de modo a mostrar informações relevantes ao negócio. Lembre-se que, diferentemente das entradas e processos, aqui os dados necessários devem ser os que a tela exibirá.

## Entradas

**R.F. XX - Nome do requisito funcional:** descrição do requisito. **Dados necessários:** dado 1, dado 2, dado 3. **Usuários:** usuário 1, usuário 2, usuário 3.



## Processos

**R.F. XX - Nome do requisito funcional:** descrição do requisito. Dados necessários: dado 1, dado 2, dado 3. Usuários: usuário 1, usuário 2, usuário 3.

## Saídas

**R.F. XX - Nome do requisito funcional:** descrição do requisito. Dados necessários: dado 1, dado 2, dado 3. Usuários: usuário 1, usuário 2, usuário 3.

## REQUISITOS NÃO FUNCIONAIS

### RESULTADO FINAL:

Apresentar requisitos não funcionais de seu projeto

### CONTEXTUALIZANDO:

Com base no que foi compreendido no fichamento previamente realizado, você deverá definir os requisitos não funcionais de seu sistema.

Algumas coisas serão mais difíceis de se medir, principalmente sendo seu primeiro projeto, mas outras serão mais fáceis. Siga seu coração, pense sobre o que foi fichado, e aplique como conseguir. Mais uma vez, dê uma olhada naqueles livros que sugerimos no fichamento da etapa anterior, isso vai ajudar bastante!

Você deve seguir o mesmo padrão para definir os requisitos não funcionais, com a diferença de que deverá adotar o padrão R.N.F. XX. Porém, não existem usuários a serem definidos e dados necessários. Veja a seguir um exemplo:

**R.N.F. 01 - Navegadores homologados:** o sistema deverá ser homologado para os navegadores Google Chrome e Mozilla Firefox.

## DIAGRAMA DE CASOS DE USO

### RESULTADO FINAL:

Apresentar Diagrama de Caso de Uso relacionado ao seu projeto.

### CONTEXTUALIZANDO:

Primeiramente, você deverá fazer um fichamento sobre Casos de Uso, abordando:

- ✓ Definição
- ✓ Descrição de Casos de Uso
  - O que é?
  - Alguns exemplos que permitam entender como desenvolver
- ✓ Diagrama de Casos de Uso
  - O que é?
  - Como representá-lo (as formas/desenhos/ícones principais e seus significados)
  - Diferenças entre *include* e *extend* (neste tópico, geralmente surgem mais dúvidas e até explicações confusas ou erradas, então solicitamos que busquem o conteúdo em dois livros disponíveis na nossa biblioteca: “Aprenda UML por meio de estudos de caso” de W. M. Góes, e “UML: a bíblia” de T. Pender)

Depois de validado o fichamento, você deverá criar o Diagrama de Casos de Uso de seu sistema, com todos os atores, casos de uso (no nosso caso são os seus Requisitos funcionais, pois não iremos fazer as Descrições de Casos de Uso) e seus relacionamentos. Sugerimos o uso do **lucidchart** (<https://www.lucidchart.com>), no entanto existe varias outras opções on-line e ou local para que você possa criar os Diagramas de UML.

## **DIAGRAMA DE ATIVIDADES**

### **RESULTADO FINAL:**

Apresentar Diagrama de Atividades relacionado ao seu projeto.

### **CONTEXTUALIZANDO:**

Primeiramente, você deverá fazer um fichamento sobre os Diagramas de Atividades, abordando:

- ✓ Definição
- ✓ Descrição de Atividade
  - O que é?
  - Alguns exemplos que permitam entender como desenvolver
- ✓ Diagrama de Atividade
  - O que é?
  - Como representá-lo (as formas/desenhos/ícones principais e seus significados);
  - Componentes Básicos de um Diagrama de atividades;
  - Símbolos de diagrama de atividade;
  - Exemplos de diagrama de atividade;

Depois de validado o fichamento, você deverá criar o Diagrama de Atividades de seu sistema, com todos os componentes e símbolos pertinentes. Sugerimos o uso do **lucidchart** (<https://www.lucidchart.com>), no entanto existe varias outras opções on-line e ou local para que você possa criar os Diagramas de UML.

## **DIAGRAMA DE SEQUÊNCIA**

### **RESULTADO FINAL:**

Apresentar Diagrama de Sequência relacionado ao seu projeto.

### **CONTEXTUALIZANDO:**

Primeiramente, você deverá fazer um fichamento sobre os Diagramas de Sequência, abordando:

- ✓ Definição
- ✓ Descrição de Sequência
  - O que é?
  - Alguns exemplos que permitam entender como desenvolver
- ✓ Diagrama de Sequência
  - O que é?
  - Como representá-lo (as formas/desenhos/ícones principais e seus significados);
  - Casos e componentes básicos;
  - Exemplos de diagrama de Sequência;

Depois de validado o fichamento, você deverá criar o Diagrama de Sequência de seu sistema, com todos os componentes e símbolos pertinentes. Sugerimos o uso do **lucidchart**

(<https://www.lucidchart.com>), no entanto existe varias outras opções on-line e ou local para que você possa criar os Diagramas de UML.

## DIAGRAMA DE CLASSE

### RESULTADO FINAL:

Apresentar Diagrama de Classe relacionado ao seu projeto.

### CONTEXTUALIZANDO:

Primeiramente, você deverá fazer um fichamento sobre os Diagramas de Classe, abordando:

- ✓ Definição
- ✓ Descrição de Classe
  - O que é?
  - Alguns exemplos que permitam entender como desenvolver
- ✓ Diagrama de Classe
  - O que é?
  - Como representá-lo (as formas/desenhos/ícones principais e seus significados);
  - Componentes básicos;
  - Exemplos de diagrama de Sequência;

Depois de validado o fichamento, você deverá criar o Diagrama de Classe de seu sistema, com todos os componentes e símbolos pertinentes. Sugerimos o uso do **lucidchart** (<https://www.lucidchart.com>), no entanto existe varias outras opções on-line e ou local para que você possa criar os Diagramas de UML.

## DIAGRAMA ENTIDADE RELACIONAMENTO

### RESULTADO FINAL:

Apresentar Diagrama de Entidade e Relacionamentos relacionado ao seu projeto.

### CONTEXTUALIZANDO:

Crie separadamente um cartão no Trello para cada OT, assim poderemos controlar sua evolução. Depois de finalizada a trilha, você deverá fazer um fichamento sobre:

- ✓ Normalização de banco de dados
  - O que é?
  - Vantagens e desvantagens
- ✓ Primeira Forma Normal (1FN)
  - O que ela determina?
  - Exemplo (mostrando o antes e depois)
- ✓ Segunda Forma Normal (2FN)
  - O que ela determina?
  - Exemplo (mostrando o antes e depois)
- ✓ Terceira Forma Normal (3FN)
  - O que ela determina
  - Exemplo (mostrando o antes e depois)

Sugerimos bastante atenção nesse fichamento das 3 formas normais, pois há vários exemplos errados e/ou complexos demais na internet e até em livros, então faça um fichamento **mais cauteloso**, focado em livros e com leitura de várias referências, para ter certeza que não se baseou numa fonte ruim. Validado o fichamento, aí sim você poderá desenvolver o DER de sua aplicação,

através do software **MySQL Workbench**, usando como base os dados dos Requisitos funcionais (ou os dados dos wireframes, caso tenhamos assim orientado), e criando as tabelas, campos (incluindo tipos, tamanhos, etc) e relacionamentos conforme aprendeu na trilha e na pesquisa.

Depois de validar seu DER, gere o SQL do seu banco (automaticamente, não manualmente...) e crie o banco para testar este SQL (não perca tempo inserindo dados nele ainda, teste apenas se o que foi gerado automaticamente funciona). Se tiver algum problema nessa etapa, há dois principais motivos (se encontrar outro, nos procure):

- Você alterou alguma chave primária e não refez os relacionamentos dessa tabela com as outras (nesse caso, refaça os relacionamentos da tabela que não foi criada com sucesso, e recrie o banco do zero após a alteração);
- O Workbench gerou uma palavra extra - “VISIBLE” - em um relacionamento (para resolver, apenas apague todas as ocorrências desta palavra no arquivo SQL).

## DICIONÁRIO DE DADOS

### RESULTADO FINAL:

Apresentar arquivo do seu Dicionário de Dados.

### CONTEXTUALIZANDO:

Com o DER criado e validado, você deverá fazer um fichamento sobre:

- ✓ Dicionário de Dados

O que é

Utilização

Como comentar seu banco no Workbench (comentários do banco em si, e das tabelas e campos que o compõe)

E, após o fichamento, deverá comentar seu banco de dados para gerar o Dicionário de Dados. Dê atenção maior para casos onde informações extra são relevantes, como um campo de nível de permissão, onde é interessante comentar qual será o valor salvo para cada nível de permissão diferente, ou um campo de status, onde cada valor armazenado representará um status diferente.

Feito isso, é só gerar o Dicionário de Dados com o plugin indicado por nós lá no Ava.

## DESENVOLVER WIREFRAMES

### RESULTADO FINAL:

Projeto do FIGMA Wireframes contendo todos os wireframes do projeto (armazenado no mesmo local deste documento)

### CONTEXTUALIZANDO:

No Ava, estão alguns arquivos sobre desenvolvimento do layout de uma aplicação. Faça a leitura de todos, e leia também este artigo, para podermos realizar uma discussão.

Após a discussão, você irá usar esses conhecimentos para criar os wireframes do que visualiza ser um layout interessante para sua aplicação. Para isso, sugerimos o uso do FIGMA ferramenta online gratuita para até três (3) projetos, no entanto pode utilizar outras ferramentas disponíveis tanto online quanto local. Fica a seu critério a utilização da ferramenta.

Desejamos ver, no mínimo, os seguintes wireframes de seu projeto:

- Tela de autenticação;
- Tela inicial de cada perfil: por exemplo, se tiver os perfis de administrador e cliente, serão 2 telas. Nessas telas, desejamos ver o padrão de cabeçalho, de menu (se for do estilo dropdown, exiba-o aberto) e rodapé - esses 3 devem ser iguais em todas as páginas daquele perfil - e também o que será carregado na parte “central” assim que o usuário for autenticado;
- Telas do padrão visual para os Requisitos de Entrada ([CRUDs](#)): para não ter que fazer todos os wireframes desses Requisitos, sugerimos que represente apenas o mais complexo (com mais dados) ou então represente de um modo genérico como será o padrão deles. Dessa maneira já conseguiremos compreender como será feita cada uma das 4 operações de cada CRUD, afinal todos seguirão este padrão. Caso haja modais ou telas diferentes para alguma das operações, também as represente;
- Todos os Requisitos Funcionais de Processo e Saída (todas as telas/modais envolvidas): como cada RF desses deve ter suas especificidades, é importante representá-las com clareza.

Considere também fazer paginação nos registros e consultas (não fizemos nas OTs, mas é básico em um software), e se pensar em outras coisas, contanto que não sejam funcionalidades novas, podemos considerar, mas ao mesmo tempo, cuidado para não dizer que fará coisas que nem sabe ainda, mas “já viu antes, então dá pra fazer”, sem nem pesquisar como se faz! Se quiser implementar algo que não viu nas OTs, primeiro pesquise como fazer, e então pense o seguinte:

- ✓ Realmente é algo necessário para meu PI ou posso ver isso depois, no Consolidado?
- ✓ Levará muito tempo para ser feito?
- ✓ Precisaré fazer “na mão” ou há algum plugin/biblioteca que faz?
- ✓ É de uma dificuldade aceitável?

Aí sim, poderemos discutir sobre a viabilidade de acrescentar um novo conhecimento no projeto.

Assim que terminar, antes de validar, use as telas criadas, principalmente nos Processos e Saídas, para simular o fluxo de execução de cada funcionalidade. Isso vai te ajudar a perceber se não falta nenhuma tela/botão.

Feito isso, aí sim, procure seus orientadores para a validação.

Ah, é MUITO IMPORTANTE que defina a ideia final de TODAS as telas, pois isso não deve ser alterado no futuro (na hora de programar), exceto em caso de algum erro. O QUE VOCÊ PROJETA E FOR APROVADO, VOCÊ IRÁ REALIZAR!

**P.S.:** Capriche na divisão das atividades do checklist desse cartão no Trello, tem bastante coisa pra fazer...

## **CRIAR ESTRUTURA DE ARQUIVOS DE SISTEMA**

### **RESULTADO FINAL:**

A organização por diretórios é o modo como o Sistema organiza logicamente os diversos arquivos contidos em um dispositivo físico de armazenamento.

### **CONTEXTUALIZANDO:**

Diretórios

O diretório contém entradas associadas aos arquivos onde são armazenadas informações como localização física, nome, organização e demais atributos.

Ao abrir um arquivo, o Sistema Operacional procura a sua entrada na estrutura de diretórios em uma tabela mantida na memória principal, contendo todos os arquivos. É necessário fechar o arquivo ao término de seu uso.

### **Nível Único:**

- Organização mais simples de uma estrutura de diretórios.
- Existe apenas um único diretório contendo todos os arquivos do disco.
- O nível único é bastante limitado, não permitindo que usuários criem arquivos com mesmo nome.

### **Master File Directory (MFD):**

- Existe um nível de diretório adicional para controlar os diretórios individuais dos usuários.
- indexado pelo nome do usuário e, nele, cada entrada aponta para o diretório (UFD) pessoal.

### **Estrutura de diretórios em árvore:**

- Existe o diretório MFD que é a raiz, os galhos são os UFD e os arquivos são as folhas.
- Cada subdiretório abaixo do MDF pode conter arquivos e novos subdiretórios e assim por diante.
- Quando se referencia a um arquivo, é necessário especificar seu nome, bem como o diretório onde ele se encontra, referência chamada PATH.
- Mais organizada e adotada pela maioria dos Sistemas Operacionais.
- Na maioria dos sistemas, diretórios também são tratados como arquivos, com identificação de atributos, proteção identificação do criador e data da criação.

## **CRIAR INTERFACE DE LOGIN**

### **RESULTADO FINAL:**

Front-End da interface visual da tela de login finalizado

### **CONTEXTUALIZANDO:**

Neste momento, desenvolva o frontend da tela de autenticação conforme seu wireframe.

Por ser sua primeira tela, você criará seus primeiros arquivos HTML, CSS e JS (esse, por enquanto, apenas para validar o formulário), e talvez ainda precise de alguma biblioteca (talvez jQuery e jQueryUI ou Bootstrap) ou imagem. Assim, organize seus arquivos de uma maneira coerente, pensando em todo o restante do projeto que está por vir e todos os outros arquivos que utilizará. Se essa tela for a inicial de seu website, lembre-se que ela deve ser a index.html e estar na raiz.

## **CRIAR INTERFACE DE LAYOUT GERAL DE CADA PERFIL**

### **RESULTADO FINAL:**

Frontend das interfaces com o layout de cada perfil finalizado.

### **CONTEXTUALIZANDO:**

Neste momento, você desenvolverá as telas para os wireframes de tela inicial de cada perfil.

Essas telas deverão conter o cabeçalho, menu e rodapé conforme seu wireframe, e uma parte da tela deve ser “reservada” para possibilitar que sejam carregados dinamicamente os conteúdos, algo que faremos mais à frente.

Pense bem na organização de pastas, separando cada perfil (isso vai lhe ajudar no futuro), porém, levando em consideração o que é igual entre eles, como por exemplo, se todos tiverem configurações CSS em comum, você poderá colocar tudo em apenas um arquivo CSS.

## **DESENVOLVER A PROGRAMAÇÃO DO PRIMEIRO CRUD DO PROJETO**

### **RESULTADO FINAL:**

Front-End e Back-End do primeiro CRUD funcionais e finalizados

### **CONTEXTUALIZANDO:**

Primeiro, defina o CRUD a ser realizado. Sugerimos que seja o Requisito Funcional de entrada que tenha menos campos e que seja um sem chaves estrangeiras pois, como será seu primeiro, terá várias coisas pra fazer além de simplesmente programá-lo (como criar a classe de conexão ao banco, por exemplo).

Depois, pense quais os arquivos necessários para isso, e onde os criará, tanto do frontend quanto do backend (que, por sinal, será iniciado agora, ou seja, pense também na organização de seus pacotes). Lembre-se de levar em conta que o arquivo HTML dele deve ser carregado dentro da área destinada para isso no layout geral (então já é uma boa hora de fazer a função de carregamento dinâmico das páginas), que o CSS desse CRUD será usado em todos os CRUDs (então use nomes mais genéricos para suas classes visando reaproveitamento), e que será a primeira ação backend de seu projeto (então o web.xml e o Jersey farão sua estreia em seu projeto).

A seguir, crie um cartão no Trello com o mesmo nome do Requisito Funcional escolhido, e o checklist dividindo as ações necessárias para sua finalização.

Por fim, prepare o SQL que criou a partir de seu DER para execução e, seguindo também seus wireframes, faça com que esse CRUD funcione conforme planejado.

## **FINALIZAR E APLICAR TRILHA DE SEGURANÇA**

### **RESULTADO FINAL:**

Conhecimentos adquiridos na trilha aplicados no projeto.

### **CONTEXTUALIZANDO:**

Sua primeira tarefa nesse momento é realizar o(s) CRUD(s) relacionados a registro dos usuários do software (de todos os níveis), e se houver outro CRUD que precise ser feito antes, por dependência direta entre eles, faça também. Solicitamos apenas que converse sobre isso com seu orientador, indicando quais CRUDs você identificou que devem ser feitos antes dos relativos ao cadastro de usuários.

Assim que finalizar, estão disponibilizadas lá no Ava as OTs da trilha de Segurança. Sua missão é fazer a trilha, validando conosco cada OT, e aplicá-la em seu projeto.

Claro, crie os cartões no Trello necessários para registrar suas atividades, tanto dos CRUDs quanto de cada OT da própria trilha.

## **DESENVOLVER PROGRAMAÇÃO DOS R.F. DE ENTRADA**

### **RESULTADO FINAL:**

frontend e backend dos R.F. de entrada finalizados

### **CONTEXTUALIZANDO:**

Agora, siga o fluxo, finalizando todas as entradas, uma por uma, das mais simples às mais complexas (ou seja, deixe os que envolvem relacionamentos entre tabelas para o final). A cada uma,

faça primeiro o frontend e depois o backend, indicando uma por uma sua finalização para que possamos validar.

Apesar do layout de todos os CRUDs ser o mesmo, dependendo de seu projeto a partir daqui você já terá algumas novidades, como manipular duas tabelas do banco de dados pelo mesmo formulário ou fazer o preenchimento de um campo afetar o carregamento de outro (como por exemplo, numa situação onde escolher um estado carregará em outro campo apenas as cidades daquele estado). Quando chegar em um caso destes, com algo novo, nos procure para que lhe orientemos sobre a melhor maneira de fazer este registro.

No Trello, crie um card para cada R.F., ordenando-os no *backlog* na sequência em que pensa em fazê-los. Ao definir os prazos, pense bem em tudo o que deve ser feito, e se houverem novidades na programação, considere o tempo de aprendizado.

## **DESENVOLVER PROGRAMAÇÃO DOS R.F. DE PROCESSO**

### **RESULTADO FINAL:**

frontend e backend dos R.F. de processo finalizados

### **CONTEXTUALIZANDO:**

Continue a programação finalizando todos os processos, um por um, primeiro frontend e depois backend, indicando um por um sua finalização para que possamos validar.

Nessa fase, já é mais provável que o layout não seja mais o mesmo. Além disso, é aqui que fará sua função de recuperar senha (que utilizará envio de e-mail), então novos conhecimentos serão necessários! Então, pense bem no que fazer e em como fazer, lembre-se de seus wireframes e conte conosco nas dúvidas que surgirem!

No Trello, como na etapa anterior, crie um card para cada R.F., ordenando-os no *backlog* na sequência em que pensa em fazê-los.

## **DESENVOLVER PROGRAMAÇÃO DOS R.F. DE SAÍDA**

### **RESULTADO FINAL:**

Frontend e Backend dos R.F. de saída finalizados

### **CONTEXTUALIZANDO:**

Para fechar, é hora de programar as saídas, da mesma maneira que antes.

Aqui já é provável que possa talvez reaproveitar algumas configurações anteriores, como dos formulários (para filtrar as consultas) ou das tabelas (para exibir os dados), mas isso depende de seus wireframes. Talvez até você tenha pensado em gráficos e, se for o caso, deve pesquisar um plugin que atenda suas necessidades. Além disso, conforme definido no projeto, você irá programar uma de suas saídas para possibilitar impressão via PDF ou exportação via CSV, então verifique na internet um plugin para isso, compreenda-o e use-o.

No Trello, siga as orientações das duas etapas anteriores.



## **ATUALIZAR DOCUMENTAÇÃO DO SOFTWARE APÓS SEU DESENVOLVIMENTO**

### **RESULTADO FINAL:**

PI finalizado!!!

### **CONTEXTUALIZANDO:**

Depois de finalizar a programação de seu PI, vamos fazer uma super hiper ultra mega power validação! Depois dela, pode ser que peçamos para fazer alguns ajustes no projeto, sejam eles anotados por você e negociados conosco durante o andamento do projeto, ou realmente verificados e solicitados por nós.

Crie esse cartão no Trello e mais um cartão chamado “Ajustes finais de programação”, e no momento da validação, alimentaremos juntos os checklists de ambos com os ajustes necessários.

*Uma vez que todas as solicitações sejam atendidas, podemos seguir para novos aprendizados!!*