linha horizontal

**Nicolas Engler**

BiblioTech

**Biblioteca**

Link para Trello: <https://trello.com/b/tUpkcG27/pi-nicolas-engler>

# CHECKLIST DE ATIVIDADES - Projeto Individual[[1]](#footnote-0)

* Criar Mapa Mental
* Descrever situação-problema
* Criar Desenho Rico
* Descrever proposta de sistema
* Descrever Regras de negócio
* Descrever Requisitos Funcionais
* Descrever Requisitos Não Funcionais
* Criar Diagrama de Casos de Uso
* Criar Diagrama Entidade Relacionamento
* Gerar Dicionário de Dados
* Desenvolver wireframes/mockups
* Criar workspace
* Criar interface de Login
* Criar interface de Layout geral de cada perfil
* Desenvolver a programação do primeiro CRUD do projeto
* Finalizar e aplicar trilha de Segurança
* Desenvolver programação dos Requisitos Funcionais de entrada
* Desenvolver programação dos Requisitos Funcionais de processo
* Desenvolver programação dos Requisitos Funcionais de saída
* Atualizar documentação do software após seu desenvolvimento

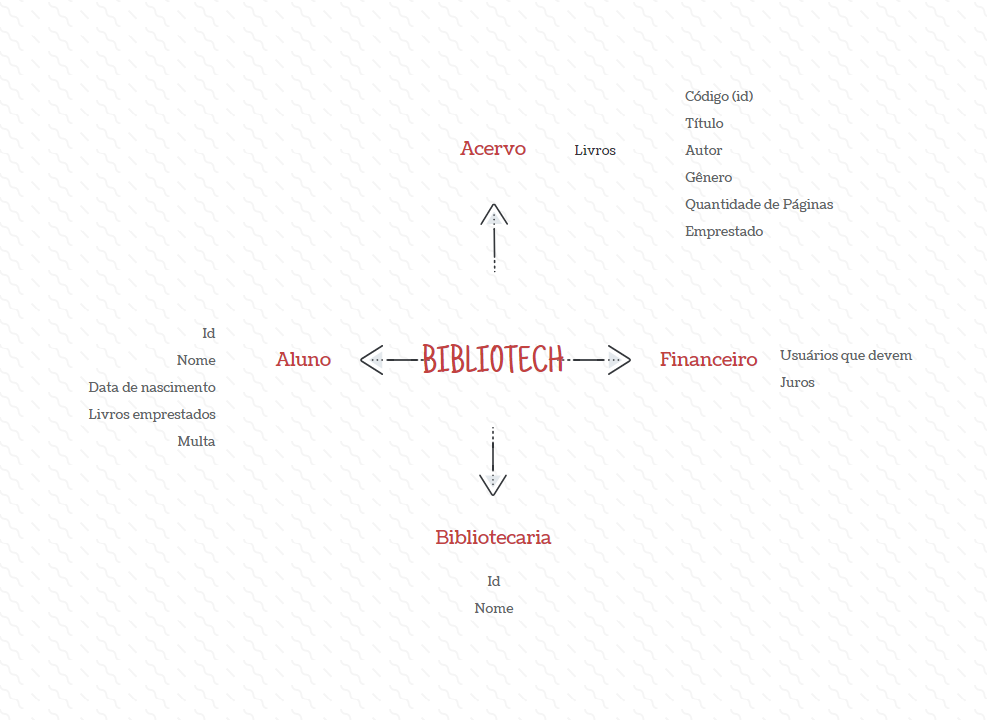
# MAPA MENTAL

RESULTADO FINAL: imagem do Mapa Mental

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. A principal intenção deste artefato será lhe auxiliar a ter compreensão dos pontos que deverão ser abordados na ferramenta seguinte.

# 

# LEVANTAMENTO DA SITUAÇÃO-PROBLEMA

RESULTADO FINAL: texto descritivo da situação-problema e imagem mostrando o Desenho Rico

CONTEXTUALIZANDO:

Primeiramente, é importante destacar que situação-problema é bem diferente de problema. Os textos que apresentamos nas 03 propostas de projeto (padaria, ONG e ordem de serviço) 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 sem conhecer as causas 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*, 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 você precisará *pesquisar sobre empresas do ramo escolhido* para entender como funcionam, e com base nisso e em seus conhecimentos previamente adquiridos (também conhecidos como experiência de vida) descrever em detalhes como a empresa funciona. 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 e *simular uma situação real* (por exemplo, no caso da padaria, dizemos que Leandro contratou mais funcionários, mas saber quantos e o que fazem pode ser relevante para o software). Resumindo, aqui você deve entender e descrever o dia-a-dia da empresa selecionada. Ah, e não se esqueça do Mapa Mental realizado 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.

Para ajudar na organização do texto, sugerimos a seguinte abordagem: comece com um parágrafo falando sobre a empresa (o que faz, o dono, funcionários...), depois fale sobre como ela funciona (focando na situação-problema, essa vai ser a parte realmente longa e detalhada, pode até organizar em mais parágrafos se precisar) e por fim, 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).

Além da descrição textual, usaremos uma ferramenta chamada Desenho Rico. No Classroom há um artigo sobre ela, então leia-o, tire suas dúvidas com os orientadores e então produza o Desenho Rico de sua situação-problema. Sugerimos que o faça em um software como Google Desenhos, Google Apresentações ou Microsoft Powerpoint, pela facilidade de inserção e organização de formas e imagens.

# SITUAÇÃO-PROBLEMA

horizontal line

A Sala Arco Íris é uma ONG que ajuda crianças cuja renda é inferior à média, tendo como principal objetivo proporcionar a elas uma educação básica de qualidade.

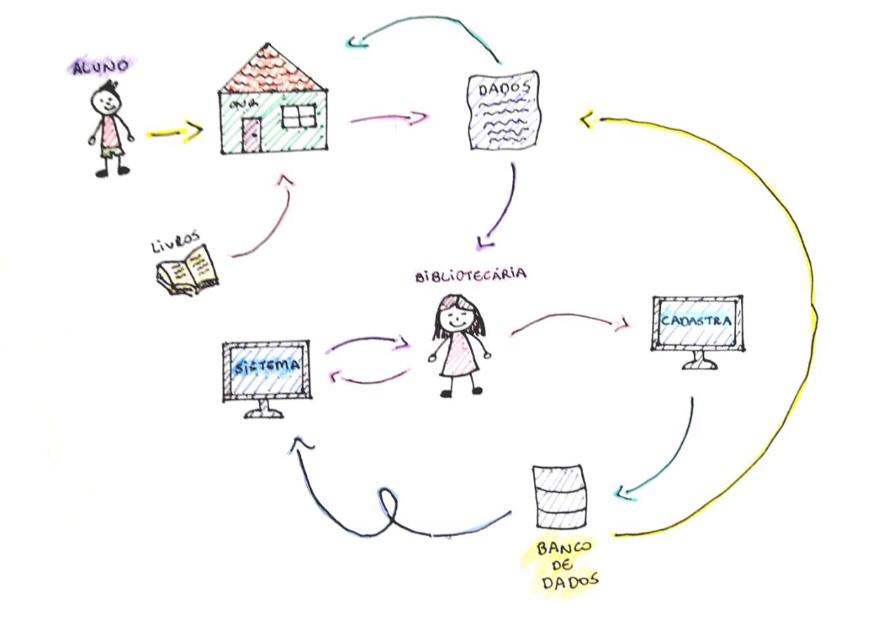
Sob esse ponto de vista, o instituto recebeu recentemente uma doação de 1000 livros, para incentivo do letramento das crianças. De maneira que, possibilita a montagem da biblioteca e sua funcionalidade. Ainda nessa perspectiva, devido a falta de verba, não é possível adquirir um leitor de impressões digitais, ou ainda, fazer carteirinhas para todos os estudantes. Sob tal conjuntura, foi decidido que será através de um software.

O sistema por meio de um usuário do tipo bibliotecário, será capaz registrar alunos, mostrando seus dados, os livros emprestados, as obras disponíveis na biblioteca, e ainda, gerar multa caso haja atrasos na entrega dos livros.

Portanto, o software desenvolvido terá como principal objetivo facilitar a organização e funcionalidade do projeto da ONG, gerando assim, melhoria e qualidade para todos.

# DESENHO RICO

horizontal line



# DESCRIÇÃO DA PROPOSTA

RESULTADO FINAL: 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, 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

# DESCRIÇÃO DA PROPOSTA

horizontal line

O principal foco desse software será trazer a facilidade para o usuário realizar as atividades necessárias para gerenciar uma biblioteca. Nele existirá apenas um tipo de usuário (bibliotecário), o mesmo será capaz de exercer as seguintes tarefas: Cadastrar alunos, cadastrar livros e ver as informações dos mesmos, realizar empréstimos, realizar devoluções

# REGRAS DE NEGÓCIO

RESULTADO FINAL: 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 Classroom há um arquivo que podem 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.

1. Um aluno pode pegar no máximo 2 livros por vez.
2. A validade do empréstimo é de 15 dias.
3. Atrasos só poderão ser justificados com atestado.

# 

# REQUISITOS FUNCIONAIS

RESULTADO FINAL: 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

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).

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, mudanças podem acontecer e não é prático sempre ficar reajustando os números.
* **Descrição do requisito:** local para descrever a função deste requisito.
* 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 abaixo, em:

* *Entradas*: São as funcionalidades que alimentarão o software com as informações essenciais para seu uso. Um exemplo é um possível “Registro de usuário”.
* *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 uma “Abertura de ordem de serviço”, que usará “Registro de clientes” e talvez “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. Exemplificando, um “Relatório de ordens de serviço” 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.

Por fim, não se esqueça que, conforme indicado no documento onde você escolheu o tema de seu PI, seu projeto deve ter: autenticação; um formulário mestre-detalhe nos processos (provavelmente no empréstimo/venda/orçamento de seu projeto); pelo menos um relatório que gere um arquivo CSV ou PDF; e função de esquecimento de senha (por e-mail). Assim, não se esqueça de indicar isso nos requisitos!

## 

## Entradas

**R.F. 01 - Cadastrar aluno**: responsável por cadastrar os dados de um aluno no sistema. Dados necessários: nome, sobrenome , data de nascimento, email, senha, gênero, escola, cpf. Usuários: bibliotecário, admin.

**R.F. 02 - Cadastrar bibliotecário**: responsável por cadastrar os dados de um bibliotecário no sistema. Dados necessários: nome, sobrenome , data de nascimento, email, senha, gênero, escola, cpf. Usuários: admin.

**R.F. 03 - Cadastrar livro**: responsável por cadastrar os dados de um livro no sistema. Dados necessários: título, autor, gênero, número de páginas. Usuários: bibliotecário, admin.

**R.F. 04 - Cadastrar gênero literário**: responsável por cadastrar um novo gênero literário no sistema. Dados necessários: gênero. Usuários: bibliotecário, admin.

**R.F. 05 - Cadastrar autor**: responsável por cadastrar um novo autor no sistema. Dados necessários: nome. Usuários: bibliotecário, admin.

**R.F. 06 - Cadastrar escola**: responsável por cadastrar uma nova escola no sistema. Dados necessários: nome. Usuários: bibliotecário, admin.

## Processos

**R.F. 07 - Autenticação de usuário**: responsável por autenticar o acesso do usuário no sistema. Dados necessários: email, senha. Usuários: bibliotecário, admin, aluno.

**R.F. 08 - Empréstimo de livro**: responsável por realizar o empréstimo de um livro. Dados necessários: aluno, livro, data. Usuários: bibliotecário, admin.

**R.F. 09 - Editar aluno**: responsável por editar os dados de um aluno no sistema. Dados necessários: id, nome, sobrenome , data de nascimento, email, senha, gênero, escola, cpf. Usuários: aluno, bibliotecário, admin.

**R.F. 10 - Editar bibliotecário**: responsável por editar os dados de um bibliotecário no sistema. Dados necessários: id, nome, sobrenome , data de nascimento, email, senha, gênero, escola, cpf. Usuários: admin.

**R.F. 11 - Editar livro**: responsável por editar os dados de um livro no sistema. Dados necessários: id, título, autor, gênero, número de páginas. Usuários: bibliotecário, admin.

## Saídas

**R.F. 12 - Log alunos**: responsável por mostrar uma lista de alunos registrados no sistema. Dados necessários: alunos. Usuários: bibliotecário, admin.

**R.F. 13 - Log livros**: responsável por mostrar uma lista de livros registrados no sistema. Dados necessários: livros. Usuários: bibliotecário, admin, aluno.

**R.F. 14 - Log bibliotecários**: responsável por mostrar uma lista de alunos registrados no sistema. Dados necessários: bibliotecários. Usuários: bibliotecário, admin.

**R.F. 15 - Log empréstimos**: responsável por mostrar uma lista de empréstimos registrados no sistema. Dados necessários: empréstimos. Usuários: bibliotecário, admin, aluno.

**R.F. 16 - Relatório de livros mais emprestados**: responsável por mostrar uma lista de livros mais emprestados no sistema. Dados necessários: relatório. Usuários: admin.

**R.F. 17 - Alunos que mais emprestam livros**: responsável por mostrar uma lista de alunos que mais emprestam livros no sistema. Dados necessários: relatório. Usuários: admin.

# REQUISITOS NÃO FUNCIONAIS

RESULTADO FINAL: 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.

**R.N.F. 02 - Linguagem implementada**: o sistema deverá ser implementado em Java como backend e usar a biblioteca Vue no frontend.

**R.N.F. 03 - Resolução mínima homologada**: o sistema deverá ser homologado para a resolução mínima de 1024x768.

**R.N.F. 04 - Profissional autorizado**: o sistema deverá ser gerenciado apenas pelos administradores.

**R.N.F. 05 - Utilização**: o sistema poderá ser utilizado por alunos, bibliotecários e administradores.

# 

# DIAGRAMA DE CASOS DE USO

RESULTADO FINAL: imagem mostrando o DCU

CONTEXTUALIZANDO:

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

* Definição
* Descrição de Casos de Uso
  + O que é
  + Como desenvolver
* Diagrama de Casos de Uso
  + O que é
  + Como representá-lo (as formas principais e seus significados)
  + Diferenças entre *include* e *extend*

Depois de validado o fichamento, você deverá criar o Diagrama de Casos de Uso de seu sistema, com todos os atores, casos de uso (que 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 Draw IO, uma extensão gratuita do Google Drive, para fazer o diagrama, pois ela possui uma aba com vários desenhos usados nos diagramas da UML.

# 

# DIAGRAMA ENTIDADE RELACIONAMENTO

RESULTADO FINAL: imagem mostrando o DER

CONTEXTUALIZANDO:

Primeiramente, você deverá concluir a trilha de Banco de Dados (você já fez a OT1 durante a trilha WEB, lembra?). Para isso, siga a partir da OT2 até o final da trilha, validando cada OT com o orientador assim que a finalizar. Sugerimos que crie separadamente um cartão no Trello para cada OT, assim poderemos controlar sua evolução.

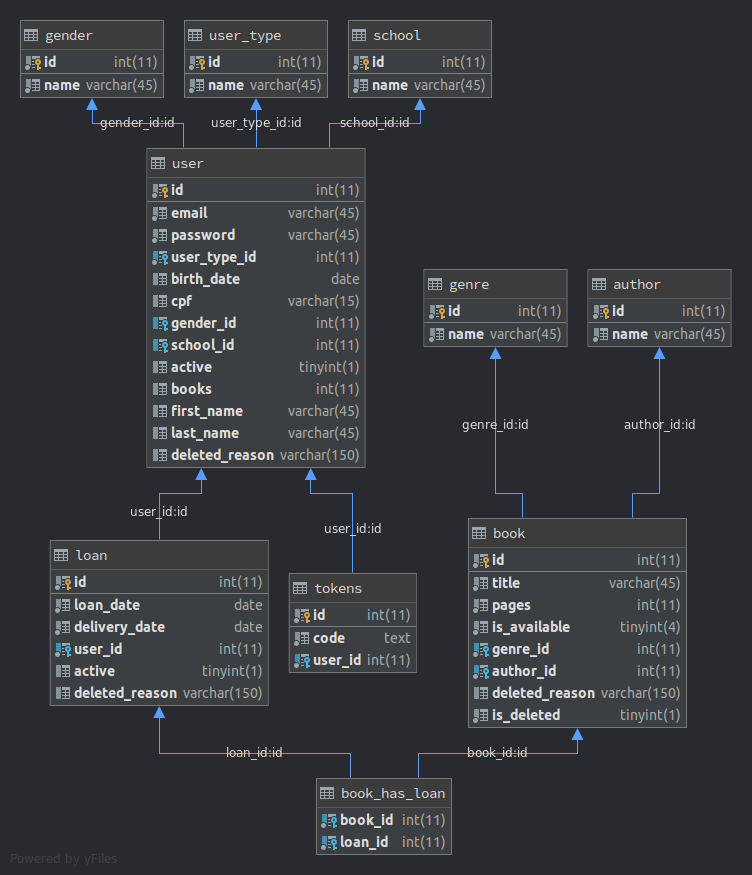
Depois de finalizada a trilha, você deverá fazer um fichamento sobre:

* Normalização de banco de dados
  + O que é
  + Vantagens e desvantagens
* Primeira Forma Normal (1FN)
  + O que ela determina
  + Exemplo
* Segunda Forma Normal (2FN)
  + O que ela determina
  + Exemplo
* Terceira Forma Normal (3FN)
  + O que ela determina
  + Exemplo

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 **de verdade**, com leitura de várias referências, não só o primeiro link de uma pesquisa no Google. 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). 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 em um relacionamento - “VISIBLE” (para resolver, apenas apague todas as ocorrências desta palavra).



# DICIONÁRIO DE DADOS

RESULTADO FINAL: arquivo do seu Dicionário de Dados (armazenado no mesmo local deste documento)

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 Classroom.

# DESENVOLVER WIREFRAMES/MOCKUPS

RESULTADO FINAL: projeto do Balsamiq Mockups contendo todos os wireframes do projeto (armazenado no mesmo local deste documento)

CONTEXTUALIZANDO:

No Classroom, estão alguns arquivos sobre desenvolvimento do layout de uma aplicação. Faça a leitura de todos, e pesquise também o que são wireframes e sua importância, 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, use a aplicação “Balsamiq Mockups 3” disponível na área de trabalho. Com ela, você terá vários recursos para desenhar suas telas.

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 cada página do perfil - e também o que será carregado na parte “central” assim que o usuário for autenticado;
* Telas do padrão visual para os CRUDs: represente o padrão de como serão todos os Requisitos Funcionais de Entrada, criando um wireframe para cada tela/modal. Sugerimos que use o mais complexo deles, pela riqueza de detalhes;
* Todos os Requisitos Funcionais de Processo e Saída (todas as telas/modais envolvidas).

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?
* Levará muito tempo para ser feito?
* Precisarei fazer “na mão” ou há algum plugin/biblioteca que faz?
* É de uma dificuldade aceitável?

Aí sim, você poderá considerar 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.

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 PROJETAR, VOCÊ IRÁ REALIZAR!

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

# CRIAR WORKSPACE

RESULTADO FINAL: workspace criada com projeto e servidor (armazenada no mesmo local deste documento ou em repositório online)

CONTEXTUALIZANDO:

Neste momento, você irá criar a workspace de seu PI. Verifique a OT inicial da trilha de frontend, pois ela tem as orientações necessárias. Esse passo é muito importante, pois se esquecer de certas coisas, como a codificação de caracteres, você terá problemas durante TODA a fase de programação. importe também todas as bibliotecas usadas nas trilhas de programação web, afinal, você também usará aqui o Jersey, o Connector/J...

Assim como antes, a workspace também deve ter um Server, onde escolherá o Tomcat disponível em sua máquina.

Para os backups diários, como já praticou bastante o Git, solicitamos o uso do Github. Temos uma organização para gerenciar todos os nossos projetos por lá, então crie uma conta com seu e-mail de estudante e nos solicite que lhe cadastremos em nossa organização. Nós mesmos criaremos o repositório para você, então só será necessário fazer os *commits* com as atualizações de seu projeto.

# CRIAR INTERFACE DE LOGIN

RESULTADO FINAL: frontend da interface visual da tela de login finalizado

CONTEXTUALIZANDO:

Neste momento, desenvolva a 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 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 da WebContent.

# 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 dos wireframes de tela inicial de cada perfil, porém sem a parte que será carregada dinamicamente (ela será programada mais a frente).

Esses arquivos deverão conter cabeçalho, menu e rodapé conforme seu wireframe, e uma parte da tela deve ser “reservada” para possibilitar que sejam carregados dinamicamente os conteúdos 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 as mesmas configurações CSS, você poderá usar apenas um arquivo CSS.

# DESENVOLVER A PROGRAMAÇÃO DO PRIMEIRO CRUD DO PROJETO

RESULTADO FINAL: frontend e backend 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 seja sem chaves estrangeiras pois, como será seu primeiro, terá várias coisas pra fazer além de simplesmente programá-lo (como 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.

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 pra rodar e, seguindo 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 algum 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 devem ser feitos antes dos relativos ao cadastro de usuários.

Assim que finalizar, estão disponibilizadas lá no Classroom 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 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, dos mais simples aos mais complexos (ou seja, deixe os que envolvem relacionamentos entre tabelas para o final) primeiro frontend e depois 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) e o formulário mestre-detalhe, 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, precisamos 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, é hora do Projeto Consolidado!

1. A ordem acima pode ser alterada de caso para caso, assim a palavra que vale é a de quem lhe orientar! [↑](#footnote-ref-0)