**DESIGN THINKING – SPRINT02**

**Grupo 3 – API 3º semestre em Banco de Dados**

**Faculdade de Tecnologia de São José dos Campos – FATEC Jessen Vidal.**

# **Empatia**

* Preciso garantir o máximo possível de integridade nos documentos.
* Necessidade de atualização manual do codelist (Excel).
* Mecanismo de comparação de documentos não é amigável.
* Problema de integridade, visto que as alterações são manuais.
* Acesso à documentos é restrita à equipe de publicações.
* Pastas são organizadas conforme mockup.
* Não gosta muito do jeito que é feito atualmente.
* Codelist é tão importante quanto a LEP.
* Revisões são organizadas em pastas.
* Revisões são processos grandes.
* Só é possível abrir uma revisão de cada vez.
* Toda revisão meche com a LEP e tem chances de mecher com a codelist.
* Os arquivos das codelists possuem várias abas.
* A LEP precisa ser feita por traço.
* "Mini LEP" para cada bloco.
* A LEP registra página por página de cada bloco.

# **Definir**

* Ver a codelist e todos os traços pertencentes.
* Ler todas as abas do arquivo de codelist.
* Editar codelist (adicionar/remover/editar blocos/traços).
* \*\*Criar um codelist pelo sistema.

# **Ideias**

* Editar uma codelist utilizando as tabelas geradas pelo back.
* Utilizar um Map para ler cada aba do arquivo e tratar os traços e blocos daquela aba usando a corrente.
* Inserir a aba desejada dentro do algoritmo que lê o excel
* Ver todos os traços de uma codelist buscando pelo nome e partnumber do documento
* Ver todos os traços de uma codelist e editar a localização dos blocos através de checklist
* Clica no traço e levar para uma página de edição dos traços
* Inserir um novo codelist no sistema através de inserção de documento e traço

# **Protótipo**

Tela de Visualização e edição do codelist.

