--- Relatório da Aplicação-Web Bancária ---

* Localização das funções/conceitos requisitados

1. Função Lambda de Alta Ordem:

* Nome do arquivo – *Alterações.py*
* Nome da função – *apagar\_usuario()*

1. Função Lambda utilizando Currying:

* Nome do arquivo – *funcoes.py*
* Nome da função – *currying(dentro de saldo)*

1. Utilização de List Comprehension:

* Nome do arquivo – *list\_compr.py*
* Nome da função – *obter\_dados\_usuarios()*

1. Utilização de um Dicionário:

* Nome do arquivo – *senha\_re.py*
* Nome da função – padrões{}

1. Utilização de um MAP ou FILTER ou REDUCE:

* Nome do arquivo – *app.py*
* Nome da função – senhas\_baixas() – dentro de autenticar()

1. Utilização de ao menos um MONAD:

* Nome do arquivo – *funcoes.py*
* Nome da função – maybe\_bind , dentro de verificar\_user()

1. Função Lambda Recursiva:

* Nome do arquivo --
* Nome da função –
* Requisitos Funcionais

1. Sistema deverá ter uma tela para usuário fazer cadastro.
2. Sistema deverá ter uma tela para usuário fazer login.
3. Sistema deverá criptografar a senha do usuário para armazenar no banco de dados.
4. Sistema deverá possuir uma tela para usuário redefinir sua senha.
5. Sistema deverá ter uma FAQ como uma forma de suporte a usuários que possam estar tendo dificuldade com o sistema.
6. Sistema deverá ter uma tela de chat, para que usuários possam interagir e trocar informações. Enquanto não se encontram.
7. Sistema deverá permitir usuário fazer saques e depósitos em sua conta.
8. Sistema deverá ter uma tela de histórico do usuário que mostre os eventos que aquela pessoa já participou.
9. Sistema produzirá feedback visível para usuário nos momentos de login e cadastro.
10. Sistema produzirá feedback visível para usuário quando fizer saque e depósito.
11. Sistema deve ter um algoritmo de filtro e busca que faça a combinação de pessoas compatíveis.

* Requisitos Não-Funcionais

1. Definição de boas cores para o sistema (melhor contraste).
2. Boa responsividade no login e cadastro.
3. Sistema deve ser seguro e dar um suporte adequado aos usuários.
4. Deve ser um serviço que possa ser personalizado que busque uma compatibilidade entre usuários e os companheiros.
5. O serviço deve armazenar diversos dados adquiridos pelo usuário pra que ele possa analisar posteriormente (ex: quantidade de eventos, nota e etc...).