

## Desafio MQLab

### Introdução:

Esse documento tem o objetivo de descrever toda a proposta de solução para o problema apresentado. Ele está dividido nas seguintes sessões:

1. Especificação Funcional: Descreve todas os requisitos funcionais e não funcionais do sistema, assim como dúvidas do projeto, que devem ser respondidas pela Mastermaq.
2. Solução Arquitetural: Descreve a proposta de solução a nível técnico.
3. Backlog: Apresenta todos os itens de backlog priorizados para uma primeira entrega.
4. Trabalhos Futuros: Descreve melhorias que poderiam ser realizadas com mais tempo.

Para facilitar referências e descrições, o projeto será nomeado **FINFAST**.

### Especificação Funcional:

#### Atores:

- Empreendedor: aquele que acessa o sistema a fim de realizar um lançamento de movimentação financeira.
- Contador: aquele que recebe, mensalmente, o relatório da movimentação financeira do Empreendedor atendido pelo Contador.

#### Requisitos funcionais:

Código	Descrição
RF1	O Empreendedor deve ser capaz de inserir, editar e remover movimentações financeiras.
RF2	O Empreendedor deve ser capaz de ver a lista de todas as movimentações financeiras do mês vigente.
RF3	O Empreendedor deve ser capaz de enviar o relatório de movimentações financeiras ao Contador.

#### Requisitos não funcionais:

Código	Descrição
RNF1	O software deve ser desenvolvido para plataforma web.

RNF2	O software deve ser desenvolvido para plataforma mobile android.
RNF3	O software deve ser desenvolvido para plataforma mobile ios.
RNF4	O software deve ter layout responsivo.
RNF5	Deve ser possível cadastrar movimentação financeira de maneira alternativa à digitação.

### Dúvidas endereçadas a Mastermaq:

1. Quais informações devem conter cada movimentação financeira?
2. Onde está localizada a informação de qual Contador atende a um Empreendedor?
3. Qual o padrão de email para os Contadores?
4. Qual a lógica de fechamento de mês? É definido um dia exato no mês para todos Empreendedores ou é definido de caso a caso?
5. Os Empreendedores estão cadastrados em algum banco de dados da Mastermaq?
6. Quais são os documentos em que os Empreendedores se baseiam para lançar suas movimentações financeiras?

### Solução Arquitetural:

O arquivo *Anexo1\_Arquitetura\_Completa.jpg* contempla a imagem em tamanho real da arquitetura proposta.

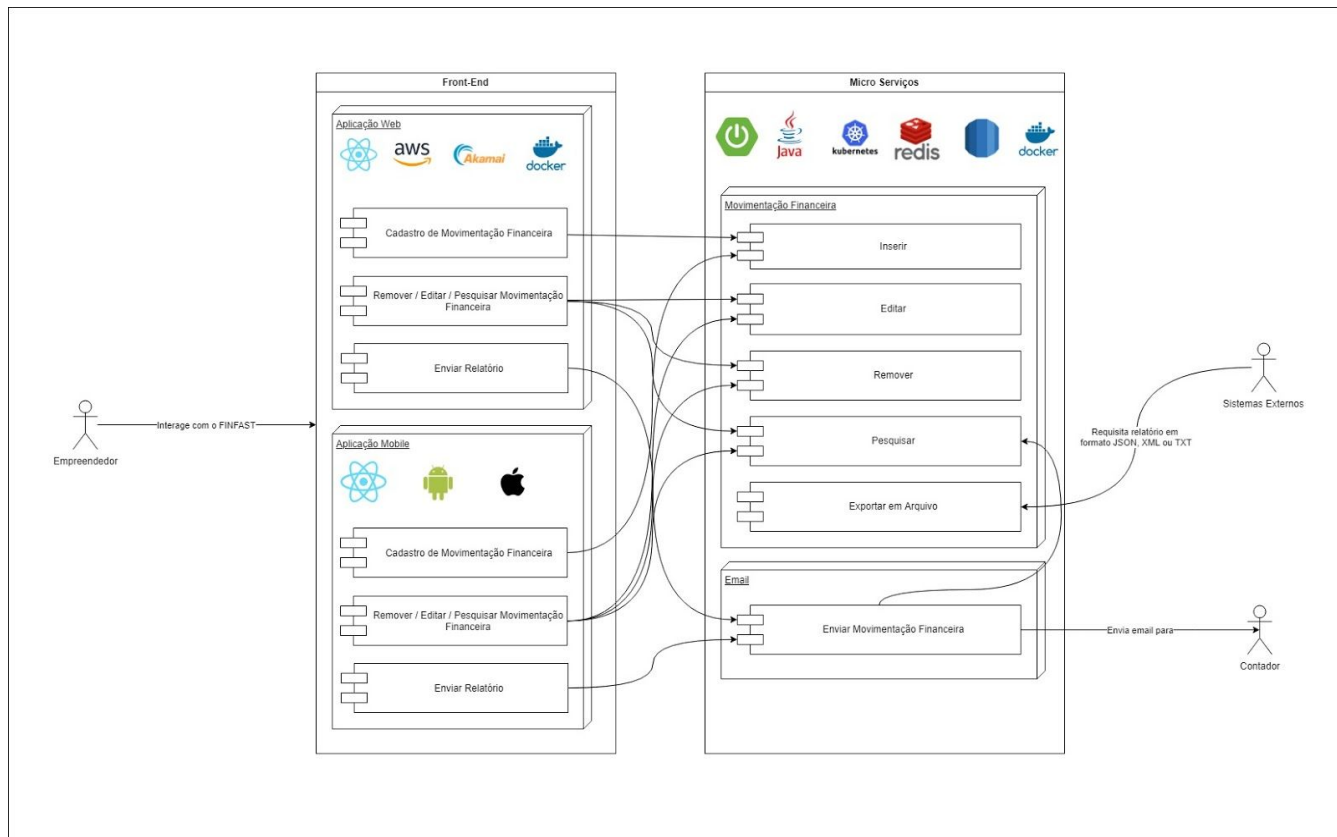


Imagem reduzida do anexo.

## Front-End:

- Framework ReactJS e React Native: Utilizando a mesma tecnologia para desenvolver tanto a aplicação web quanto a mobile, é possível aproveitar o know-how da equipe para um desenvolvimento mais eficiente.
- A aplicação ficará hospedada em uma máquina EC2 na aws e, visando escalabilidade global da aplicação, utilizaremos CDN Akamai para cache.
  - Como a comunicação é totalmente feita client-side, o CDN poderia ser substituído por qualquer fornecedor de cache, como Dispatcher por exemplo, caso não seja um desejo escalar a aplicação para que ela seja global.
  - A configuração da máquina pode ser *t2.small*, visto que toda a lógica é processada *client-side*.
- Todas as ações do usuário que resultam em manipular ou pesquisar dados em banco será feita via micro serviços;
  - Essa abordagem faz com que, indiferente da maneira que o usuário vai interagir com o sistema, o código que contém a manipulação do dado é sempre o mesmo.
  - Foi proposto uma aplicação mobile, porém, caso seja desejado implementar apenas uma aplicação web, a lógica dos micro serviços se mantém a mesma.
- Haverá um *dockerfile* preparado para iniciar um container *Docker* com a aplicação web.
  - Gerando uma imagem *Docker* a cada fechamento de versão a entrega em produção é facilmente automatizada.

## Micro Serviços:

- O framework *Spring Boot* foi escolhido pois facilita a criação de micro serviços integrados com o banco de dados relacional *Amazon RDS*, juntamente com integração a biblioteca de cache Redis.
  - O *End-Point* “Exportar em Arquivo” deve ter cache configurado em relação ao mês vigente dos relatórios.
  - O *End-Point* “Pesquisar” deve ter cache perene.
  - Os *End-Points* “Inserir”, “Editar” e “Remover” devem limpar o cache do *End-Point* “Pesquisar”.
- *Kubernetes* foi escolhido por ter escalabilidade totalmente configurável.
  - Ao passar do tempo, o número de acessos vai nos dizer quantos clusters são necessários para cada Pod.
- Pela característica de ter dados alterados a todo momento, não foi decidido utilizar uma abordagem *NoSQL*, a velocidade de disponibilidade dos dados pesquisados vai ser garantida via cache no Redis.
  - Caso a abordagem não seja o suficiente, é possível criar uma estrutura no *CloudSearch (AWS)* contendo os dados de movimentação financeira, que deve ser atualizada com um período fixado (Exemplo: diariamente).
- Assim como na aplicação web, haverá automatização de entrega em produção via container Docker.

## Propostas para cobrir RNF5:

É necessário uma resposta para a pergunta 6 endereçada a Mastermaq, porém é possível propor algumas soluções para cenários específicos:

1. Boletos bancários: Na versão dos aplicativos, é possível preencher automaticamente alguns campos do cadastro de movimentação financeira com a leitura do código de barras.
2. Extrato bancário: Como extratos são geralmente padronizados, é possível alimentar o sistema com uma imagem e retirar informações dela, a fim de automatizar o lançamento. A maneira mais confiável de extrair essa informação é utilizando a API [Google Vision](#). A API retorna todas informações da imagem em formato JSON, que pode ser utilizada em qualquer aplicação.
  - a. O usuário, na aplicação mobile, poderia iniciar esse preenchimento tirando uma foto do extrato, porém a confiabilidade da imagem é baixa. Sendo assim não é recomendado liberar funcionalidade de leitura por foto da câmera.
3. Machine learning: É possível facilitar o trabalho de lançamento observando e aprendendo alguns comportamentos do usuário:
  - a. O software deve identificar lançamentos corriqueiros, que devem virar alertas para o usuário apenas confirmar se ele vai continuar acontecendo naquele mês ou nos próximos.
  - b. O software pode ser alimentado com o conhecimento de que existe correlação em alguns tipos de lançamento e, a partir disso, criar alertas para o usuário que um lançamento deve ser feito em detrimento de outro.
  - c. Os termos que forem inseridos pelos usuários podem ir para uma mesma base de conhecimento, a fim de exibir funcionalidade de auto-completar nos cadastros.

## Backlog:

Abaixo está descrito alguns dos itens de backlog para implementação da solução proposta. Os itens estão ordenados, de cima para baixo, por prioridade.

Código	Descrição
1	<b>Micro Serviço:</b> criar Movimentação Financeira, com <i>End-Points</i> : <ul style="list-style-type: none"><li>• Inserir</li><li>• Editar</li><li>• Remover</li><li>• Pesquisar</li><li>• Exportar em Arquivo</li></ul>
2	<b>Micro Serviço:</b> criar Email, com <i>End-Point</i> : <ul style="list-style-type: none"><li>• Enviar Movimentação Financeira</li></ul>
3	<b>Front-End Web:</b> Criar tela de cadastro de movimentação financeira.

4	<b>Front-End Web:</b> Criar tela de pesquisa de movimentação financeira.
5	<b>Front-End Web:</b> Criar funcionalidade de edição de movimentação financeira.
6	<b>Front-End Web:</b> Criar funcionalidade de remoção de movimentação financeira.
7	<b>Front-End Web:</b> Criar funcionalidade de envio de relatório.

## Trabalhos Futuros:

A autenticação do usuário Empreendedor não foi contemplada no desenho da solução devido a tempo. Em resumo, deveria ser feito um sistema de autenticação SSO (Single Sign On), que poderia se basear em Cognito para autenticação e permissão de usuário, com federação para diversos logins sociais (Google, facebook, twitter...).

Criação de backlog completo com discriminação de área de atuação (DevOps, Desenvolvimento Front-End, Desenvolvimento Back-End, e desenho de telas).