

Descrição

Público-Alvo

Premissas

Funcionalidades

Arquitetura

Protótipo de Interfaces do Usuário

Tela 1 – Visão Cliente - Cafézinho

Tela 2 – Visão Cliente – Mapa com escritórios

Tela 3 – Visão Cliente – Lista de Advogados

Tela 4 – Visão Cliente – Detalhes do Advogado

Tela 5 – Visão Advogado – Lista de “cafézinhos”

Tela 6 – Visão Advogado Chat

Tela 7 – Visão Cliente Tablet

Tela 8 – Widget Visão Cliente

Tela 9 – Widget Visão Advogado

Considerações Chave

Como seu app vai tratar a persistência de dados?

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

Descreva como você implementará o Google Play Services.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto

Tarefa 2: Implementar o Servidor Restful

Tarefa 3: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

Tarefa 4: Desenvolver o acesso ao servidor Restful

Tarefa 5: Implementar o Content Provider

Tarefa 6: Implementar os Loaders

Tarefa 7: Implementar Google Play Services

Tarefa 8: Implementar o chat firebase

Agradecimentos

Usuário do GitHub: mmiotto@gmail.com

Café Legal

Descrição

- Vamos tomar um cafezinho? – esse é um convite feito toda vez que queremos ter um bate papo informal com alguém, mas muitas vezes serve também para tirarmos uma dúvida sobre assuntos importantes, técnicos ou de negócios.

O aplicativo “Café Legal” pretende auxiliar as pessoas a entrarem em contato com advogados, facilitando o acesso aos profissionais através de um convite para um bate papo informal. Se o convite for aceito, neste momento, será possível tirar uma dúvida jurídica, pedir um conselho ou uma orientação através de um chat.

Para facilitar ainda mais esse contato será possível visualizar os advogados (escritórios) próximos segundo alguns critérios de seleção do usuário, bem como acessar uma lista de todos os advogados cadastrados no app.

Será apresentado ao usuário os detalhes do advogado selecionado (clicado) para contato, como email, telefone, endereço e etc.

O app para os especialistas terá as mesmas funcionalidades, mas, em vez de uma lista de advogados, será possível visualizar todos os convites ainda em aberto.

Público-Alvo

O público alvo são todas as pessoas que tenham alguma dúvida jurídica e não tenham acesso a um advogado de maneira fácil, sem necessariamente oficializar uma consulta.

Premissas

Para este documento passam a ser chamados “clientes” todos os usuários do aplicativo não especialistas e “advogados” todos os usuários especialistas, os próprios advogados.

Para fins de entendimento neste documento será tratado a partir deste momento o convite para os advogados como “cafézinho”.

Depois de pesquisas não foi possível encontrar um Web Service que fizesse a validação do número da OAB no momento do cadastro do advogado no App. Portanto, para fins deste projeto, está sendo considerada a validação manual. Essa validação será feita no site CNA (Cadastro de Advogados - <http://cna.oab.org.br/>). O advogado somente receberá notificações de clientes após esta confirmação. Esta validação será feita pelo administrador do servidor.

Foi adquirido o link cafelegal.net para acessar o servidor Restful.

As funcionalidades semelhantes em ambas as visões (Cliente e Advogado), quando acessadas por advogados este documento considera-os como clientes.

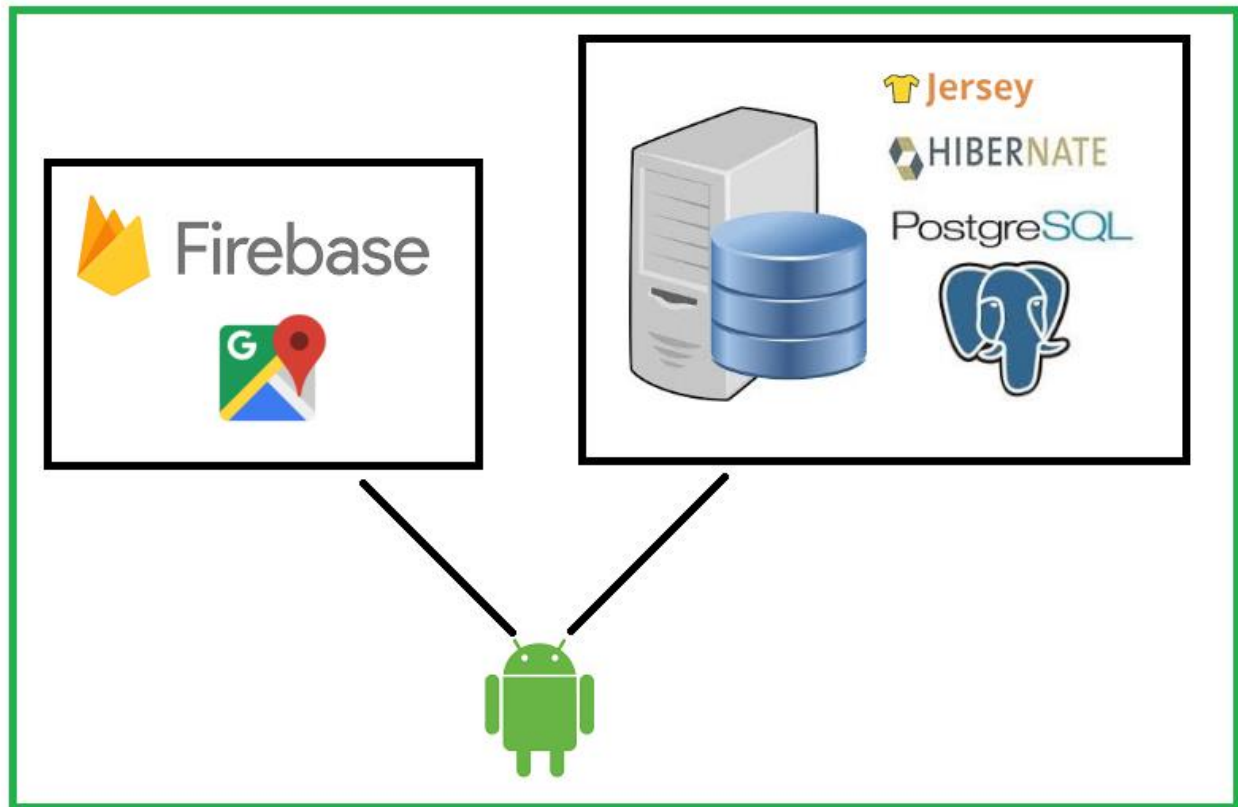
As funcionalidades de envio de convite e mapa com advogados próximos serve também para os advogados porque pode fortalecer a sua rede de contatos.

Funcionalidades

As principais funcionalidades são:

- Envia um convite para um “cafézinho” - se aceito estabelece um chat entre o cliente e o advogado.
- Marca através da geolocalização os escritórios de advogados mais próximos em um mapa, considerando os critérios de seleção do cliente.
- Lista todos os advogados que atendam aos critérios de pesquisa – somente na visão para os clientes.
- Lista todos os “cafézinhos” em aberto – somente na visão para os advogados.
- Apresenta os detalhes do advogado selecionado (clicado) no mapa ou na lista de advogados.

Arquitetura

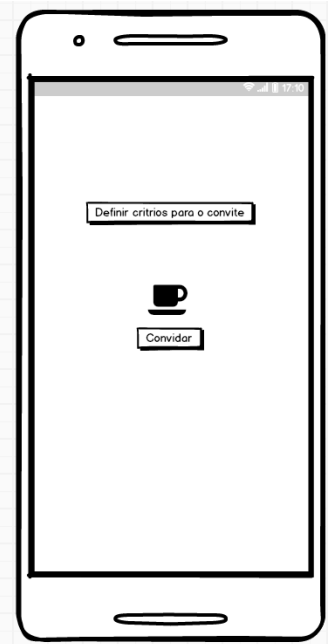


O app utilizará o Firebase para fazer a autenticação e será o repositório dos dados do chat. Serão utilizados também o Google Maps and Location do Google Play Services.

Um servidor Restful será usado para manter o cadastro dos clientes e advogados e as solicitações de “cafézinho”.

Protótipo de Interfaces do Usuário

Tela 1 - Visão Cliente - Cafézinho

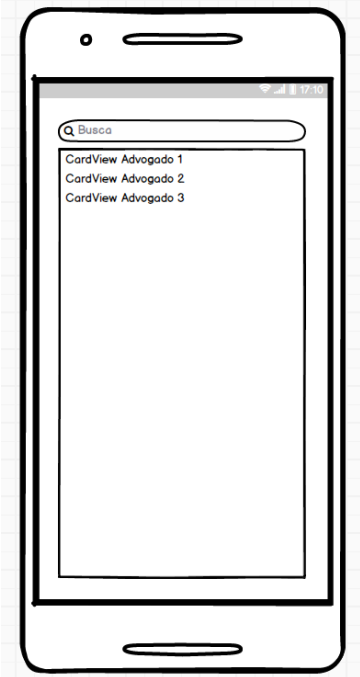
	<p>Nesta tela é possível fazer o convite para um cafézinho. Os critérios serão definidos em uma janela Dialog.</p> <p>Os critérios serão a especialidade do advogado e a localização.</p>
--	---

Tela 2 - Visão Cliente - Mapa com escritórios

	<p>Nesta tela, a partir do geoposicionamento do cliente serão mostrados todos os escritórios próximos, considerando sempre os últimos critérios definidos.</p>
---	--

b

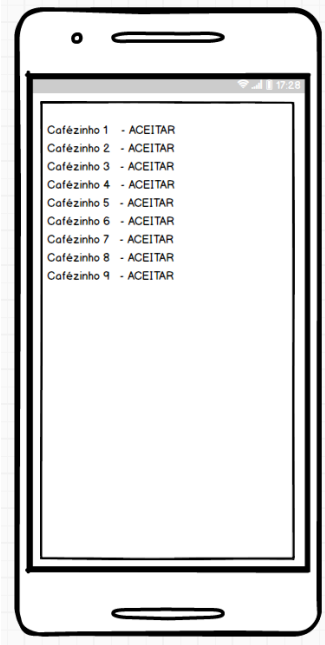
Tela 3 - Visão Cliente - Lista de Advogados

	<p>A lista de advogados será apresentada em CardView, onde cada card representará um cartão do advogado. Ao clicar em um cartão o cliente é direcionado à tela de detalhes do advogado.</p>
---	---

Tela 4 - Visão Cliente - Detalhes do Advogado

	<p>Nesta tela teremos todos os dados do advogado, facilitando o acesso ao mesmo.</p>
---	--

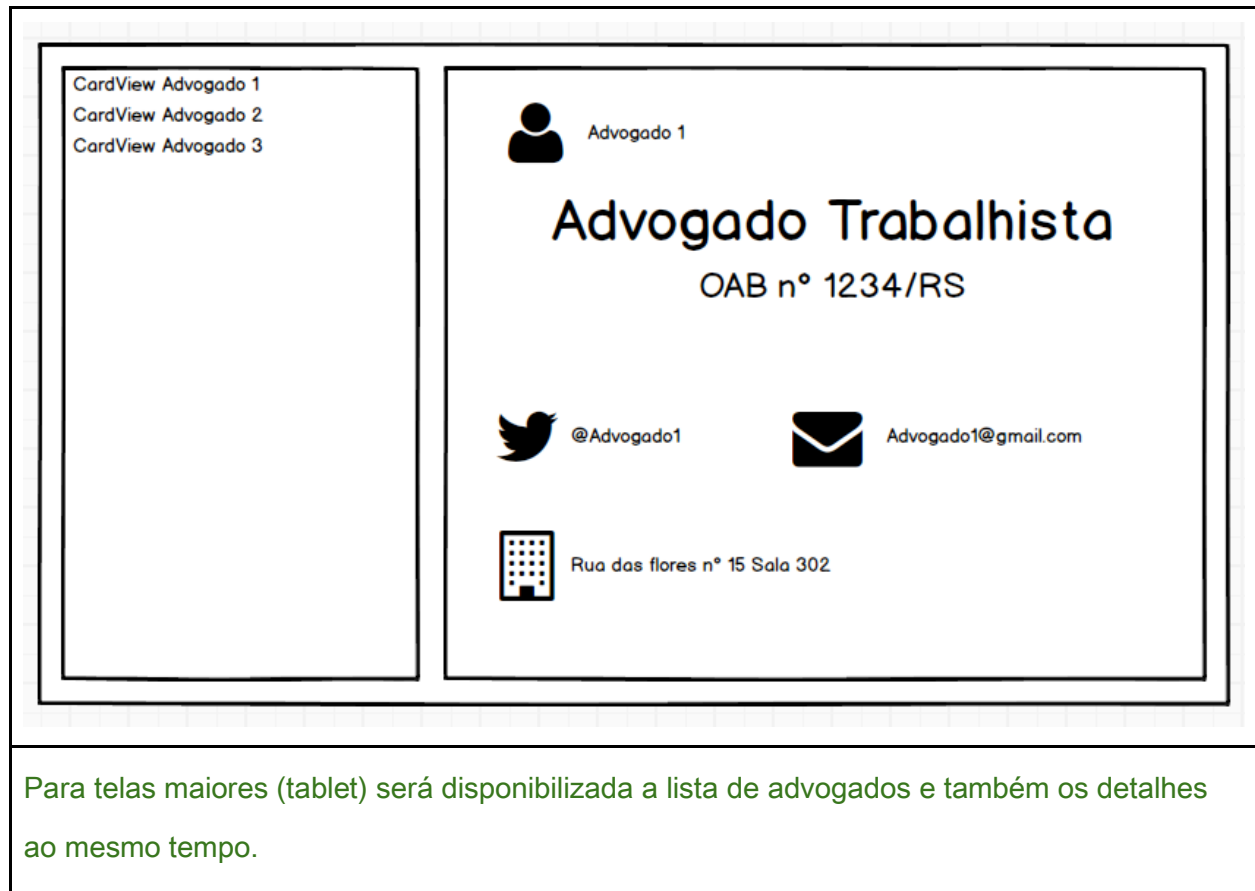
Tela 5 - Visão Advogado - Lista de “cafézinhos”

	<p>Nesta tela estarão disponíveis todos os convites ainda não aceitos em que o advogado preencha os critérios de seleção.</p>
---	---

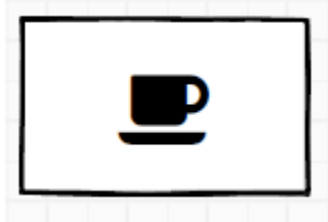
Tela 6 - Visão Advogado Chat

	<p>Esta é a tela do chat que será aberta após o advogado aceitar o convite.</p>
---	---

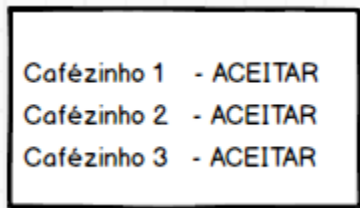
Tela 7 - Visão Cliente Tablet



Tela 8 - Widget Visão Cliente

	<p>Com este widget o Cliente poderá enviar os convites sem acessar o app. O convite utilizará o ultimo criterio de pesquisa para enviar o convite.</p>
---	--

Tela 9 - Widget Visão Advogado

	<p>Com este widget o Advogado poderá aceitar os convites em aberto sem acessar o app. Após aceitar o convite o chat será aberto.</p>
---	--

Considerações Chave

Como seu app vai tratar a persistência de dados?

- O app terá um Content Provider para persistir os dados:
 - Dos advogados (escritórios) próximos que correspondam ao critério de pesquisa – funcionando como uma memória cache.
 - Os critérios de seleção de advogados das últimas 5 pesquisas.
- A autenticação e conversas do chat serão persistidos no Firebase
- Os dados do cadastro do perfil e convites para “cafézinhos” serão persistidos no PostgreSQL no servidor Linux.

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

Ao clicar no botão para enviar o convite o usuário será avisado através de um snack bar que o convite foi enviado, o app pode ser fechado. A partir deste momento ele deve aguardar que o convite seja aceito. Ele então retornará ao app para o chat com o advogado através de uma notificação.

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

Serão utilizadas as seguintes bibliotecas:

- Picasso – para tratar as imagens armazenadas no servidor, esta biblioteca foi utilizada no projeto “Filmes Populares” e estou bem confortável em utilizá-la.
- Volley – para tratar as requisições para as Rest APIs, também foi utilizada no projeto Stock Hawck!
- Butter Knife – annotation – esta biblioteca será utilizada para facilitar o acesso aos itens da UI. Foi recomendada durante um code review e achei muito interessante.
- GSON - para serializar e deserializar objetos java em json.

Descreva como você implementará o Google Play Services.

Serão utilizados os seguintes Google Play Services:

- Firebase – será utilizado para autenticação e para persistir os dados dos chats entre clientes e advogados.
- Maps – será utilizado para fazer um pin dos advogados (escritórios) próximos ao cliente.
- GeoLocation – será utilizado para definir a localização para o pin no Maps.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto

- Configurar as bibliotecas escolhidas;
- Configurar o Google Play Services (chaves e permissões)
- Criar o servidor restful e estrutura de diretórios;
- Fazer a configuração do Hibernate;
- Fazer a configuração do PostgreSQL;
- Fazer o setup do link <https://cafelegal.net> para o servidor criado;
- Configurar o servidor Linux com SSL;

Tarefa 2: Implementar o Servidor Restful

- Escrever o código java para as APIs do servidor Restful.
- Escrever o código java para testar as APIs.
- Fazer a definição das tabelas para a persistência dos convites e cadastro dos clientes e advogados.

Tarefa 3: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

- Construir a UI da MainActivity
- Construir a UI para o fragment “Convite para o cafezinho”
- Construir a UI para o fragment “Mapa”
- Construir a UI para o fragment “Lista de advogados”
- Construir a UI para o fragment “Lista de cafézinhos”
- Construir a UI da DetailActivity

Tarefa 4: Desenvolver o acesso ao servidor Restful

- Desenvolver o acesso ao servidor Restful

Tarefa 5: Implementar o Content Provider

- Implementar o content Provider

Tarefa 6: Implementar os Loaders

- Implementar os Loaders

Tarefa 7: Implementar Google Play Services

- Implementar o acesso ao Google Play Services
- Implementar a autenticação com o Firebase

Tarefa 8: Implementar o chat firebase

- Implementar o chat, será usado o código utilizado nos estudos opcionais, Firebase in a Weekend.

Agradecimentos

Quero abrir um item de agradecimento à:

- Susana Barros (esposa) que fará os assets gráficos utilizando o Corel Draw.
- Fernando Miotto (irmão) – advogado OAB 22750/SC – que foi consultado inúmeras vezes sobre todas as regras de negócio e cadastro dos advogados.