



Pós-graduação em Desenvolvimento Mobile
APIs e Web Services
(Prof. Luiz Vitor Ramos Silva)
Tarefa 1

Nome: Ismália Dulce Gonçalves Santiago

Proponha detalhadamente as APIs necessárias e descreva as abordagens e motivadores das decisões para o caso abaixo:

Um novo produto está sendo criado pela empresa Ideia Original. É um aplicativo mobile e um site para conectar pessoas que buscam e prestam serviços do lar, como, faxina (limpeza) e marido de aluguel (serviços pequenos de manutenção como troca de chuveiro queimado, troca de torneira vazando etc.)

1 - A aplicação deve ter uma área para que seja possível fazer o CRUD (Create, Read, Update, Delete) de categorias e serviços disponíveis na plataforma.

Para operações CRUD básicas, como as de categorias e serviços disponíveis na plataforma, a API REST é a melhor opção.

2 - O prestador de serviço deve conseguir se cadastrar. E por medida de segurança, o cadastro da pessoa deve ser verificado, validando se os dados inseridos são reais e se a pessoa possui alguma pendência com a justiça.

O cadastro de prestador pode ser feito usando API REST, já que ela é a melhor opção para operações CRUD.

Validar se os dados inseridos pelo prestador são reais pode ser feito, por exemplo, com a API Datavalid (<https://www.loja.serpro.gov.br/datavalid/>), fornecida pelo SERPRO, que checa a validade dos dados junto às bases oficiais do governo (RFB e SENATRAN).

Já a verificação se a pessoa possui alguma pendência com a justiça pode ser feita, por exemplo, pela API governamental Certidão de Antecedentes Criminais (<https://www.gov.br/conecta/catalogo/apis/antecedentes-criminais>), fornecida pelo Ministério da Justiça e Segurança Pública, que retorna a informação de “Nada Consta” de antecedentes criminais transitados em julgado, consultando a base de dados do Sistema Nacional de Informações Criminais (SINIC), gerenciada pela Polícia Federal.

Podemos fazer um aplicativo apenas para o prestador de serviço.

3 - O prestador após ter o seu cadastro validado poderá cadastrar em quais categorias e serviços existentes no aplicativo ele se enquadra, junto com o tempo necessário e o valor de execução do serviço.

Todas essas operações podem ser feitas utilizando API REST, já que se encaixam em operações CRUD.

4 - O usuário cliente deve conseguir se cadastrar na aplicação. E por medida de segurança, o cadastro da pessoa deve ser verificado, validando se os dados inseridos são reais e se a pessoa possui alguma pendência com a justiça.

Aqui utilizamos a mesma estratégia já descrita no item 2:

- O cadastro do cliente pode ser feito usando API REST;
- A validação dos dados inseridos pode ser feita pela API Datavalid do SERPRO;
- A verificação se a pessoa possui alguma pendência com a justiça pode ser feita pela API Certidão de Antecedentes Criminais do Ministério da Justiça e Segurança Pública.

Podemos fazer um aplicativo apenas para o usuário cliente.

5 - O usuário cliente poderá cadastrar, editar e apagar um meio de pagamento pelo aplicativo. A fim de segurança o aplicativo não terá dinheiro físico como opção, apenas cartão de crédito ou saldo no próprio aplicativo (depositado por meio de pix). O aplicativo deve ter uma função de transferir o dinheiro de saldo nele por um pix, ao qual foi informado no cadastro.

A informação do saldo no próprio aplicativo pode utilizar API REST e ficar armazenada no banco de dados.

APIs como o Pagar.me (<https://docs.pagar.me/>) funcionam como um gateway de pagamentos, permitindo que o usuário cadastre, edite ou apague um meio de pagamento, e fornecendo condições para realizar pagamentos com cartão de crédito ou com Pix no aplicativo.

6 - O usuário cliente poderá fazer a pesquisa do prestador pelo filtro desejado, desde todos os prestadores da região, até filtros mais específicos como prestadores disponíveis no dia e horário marcados específicos para determinados serviços e com nota de avaliação boa ou serviço imediato ao qual o prestador que pegar, irá atender imediatamente.

É possível construir ordenações, filtros e paginações em uma API REST para atender aos propósitos desse requisito, sendo possível inclusive construir filtros complexos.

7 - Quando o usuário cliente marcar um serviço e o prestador aceitar o mesmo, o valor do serviço deve ser cobrado, ficando bloqueado no aplicativo e liberado apenas quando o serviço for marcado como concluído ou estornado quando cancelado.

Podemos utilizar WebHook para dispararmos, por exemplo, uma push notification quando um evento importante ocorrer (nesse caso, quando o serviço for marcado como concluído, ou estornado quando cancelado). Enquanto o status do WebHook for “pending”, bloqueamos cobranças via gateway de pagamento no aplicativo.

8 - A aplicação deve possuir um mapa, para facilitar a busca de endereços e dos prestadores de serviço por região. Quando o prestador estiver indo executar o serviço dentro do aplicativo mobile, a rota para o endereço do usuário deve ser traçada. O usuário poderá acompanhar em tempo real o deslocamento do prestador.

A API Google Maps (<https://developers.google.com/maps/documentation?hl=pt-br>) fornece todos os serviços citados nesse requisito, mas ainda é necessário estudar a documentação para saber como aplicar cada sub-API do Maps na situação correta.

9 - A aplicação deve ter como funcionalidade um chat em tempo real entre o usuário e o prestador de serviço.

Pode-se desenvolver um chat em tempo real entre o usuário e o prestador usando WebSocket. Todavia, podemos também utilizar uma API pronta como o Real Time Chat API (<https://developer.zendesk.com/api-reference/live-chat/real-time-chat-api/introduction/>), da Zendesk.

10 - O prestador deve ter a opção de concluir o serviço no aplicativo após executar o mesmo, ao clicar em concluir serviço, será lhe pedido um código de confirmação que será enviado ao usuário que pediu o serviço, o prestador deve perguntar verbalmente qual o código, e ao inseri-lo e confirmar o fechamento do serviço, o dinheiro do serviço ficará disponível em sua conta. Após a conclusão do serviço o usuário poderá avaliar o prestador de serviço.

Mais uma vez citando a API Pagar.me, ela pode ser usada para confirmações de cobrança usando um código. Podemos configurar para que esse código seja mostrado apenas no aplicativo do usuário do serviço e fazer com que ele o informe de forma verbal ao prestador. Também podemos programar para o gateway de pagamentos só liberar o dinheiro quando o aplicativo informar o fechamento do serviço após as devidas confirmações.