Funcionalidades:

Cadastro de conta:

- Mantido pelo administrador do sistema.

- Acrescentar imagem (logo)

- Falta serviço para o motorista se cadastrar através do aplicativo.

Cadastro de usuário:

- Mantido pelo administrador do sistema e pelo administrador da conta.

- Administrador do sistema informa a conta que o usuário pertence

- Administrador da conta só cadastra usuários vinculados a sua conta.

- Está cadastrando usuário com a senha, teria que mandar um link para o usuário confirmar a senha por exemplo.

- Falta serviço para alterar a senha e para recuperar a senha quando esquece.

- Falta opção para vincular e desvincular conta ao usuário.

- Falta serviço para o motorista se cadastrar através do aplicativo.

- Primeiro usuário cadastrado na conta vira administrador dela (implementado)

- Não retornar senha no serviço rest

Cadastro de UF:

- Será mantido pelo administrador do sistema. Sossegado.

Cadastro de município:

- Será mantido pelo administrador do sistema. Daria pra carregar um arquivo direto no banco de dados e ter o cadastro só pra ter mesmo. Sossegado.

Cadastro de estabelecimento:

- Mantido pelo administrador do sistema.

Cadastro de veículo:

- Falta definição dos atributos, inclusive proprietário (ou conta proprietária?).

- Falta definição de quem será responsável pelo cadastro, tendo poder de incluir ou alterar.

Demanda de transporte (cadastro):

- Será feito pelo tomador do serviço (empresas como Coamo, Seara, Cocamar, etc).

- Não existe forma de quantificar o produto em peso ou volume, talvez seria importante para validar a quantidade prevista para o veículo embarcar quando cria a viagem e para composição de saldos de embarques

- Quando a demanda será encerrada?

Demanda de transporte (painel de consulta):

- Quando a demanda deixará de aparecer dentro das "pendentes"? Apenas uma marcação "ocultar" pelo usuário?

Programação do veículo:

- Falta implementar serviço rest, gravação já está funcionando.

- Falta notificação. A criação da viagem tem que notificar o motorista através do aplicativo.

Viagem (aplicativo):

- Falta tudo

- Vai pedir usuário e senha, o motorista poderá criar a sua conta ou já se logar se possui conta

- O aplicativo irá buscar no servidor a viagem do motorista e dar opção para ele aceitar ou recusar, gerando um evento que será enviado para o servidor

- A partir daí o motorista terá funcionalidades para:

- informar que iniciou a viagem

- informar previsão de chegada em qualquer local programado

- registrar sua chegada num local

- registrar o término da operação no local

- registrar a saída do local

- encerrar a viagem

- Após encerrada a viagem, o aplicativo busca no servidor a próxima viagem.

- Se já tem próxima viagem sem a atual estar encerrada, dar um jeito de mostrar que tem uma nova viagem, mas não deixa iniciar, apenas aceitar ou recusar.

- Tudo off-line e transmite quando tem sinal

- Se a viagem no celular estiver ativa mas no servidor constata que ela foi cancelada, então o aplicativo deve buscar a nova viagem.

Outras pendências:

1 – [baixa] Como pegar exceção do servlet para devolver um json de objeto RespostaErro. Necessário nas situações em que a requisição REST foi mal formada e o servidor devolve uma exceção para o cliente, e o método da aplicação nem é executado;

2 –[alta] Como estruturar uma consulta que aceite diversos filtros e critérios de ordenação. Será necessário para o painel de demandas, onde o usuário poderá combinar vários filtros para encontrar a demanda desejada. Exemplos de filtro: município de coleta, uf de entrega, produto, status. Implementar um método para cada combinação possível acho que se torna inviável.

Idéia: objeto com os critérios de pesquisa para a camada cliente interagir com a camada de serviço e uma classe (talvez camada dao) que permitiria a camada de serviço ligar e desligar os critérios. Por exemplo, objeto.ativarFiltroPorMunicipio(String município).

<https://en.wikibooks.org/wiki/Java_Persistence/Criteria>

3 –[baixa] Avaliar: No cadastro de estabelecimento, podemos já colocar endereço e também latitude/longitude, e depois pensamos em mapear os locais onde o veículo estaciona para que o aplicativo possa reconhecer que o veículo chegou no local.