

Faculdade de Ensino Superior do Paraná - FAESP

Sistema de transporte

Curitiba

2019

Faculdade de Ensino Superior do Paraná - FAESP

Projeto Integrador Java

Docente: Eduardo Cristiano Kondras Miranda

Discentes:

Bruno de Sousa de Almeida

Rafael Rezende dos Santos

6º Período

Sistemas de Informação

Curitiba

2019

Sumário

Introdução.....	4
Projeto Java.....	5
Atividade Proposta.....	5
Problemática.....	5
Solução.....	6
Linguagem e Método de Desenvolvimento.....	6
Anexos.....	8
Conclusão.....	10
Referências.....	10

Introdução

Trabalho proposto para a disciplina de Projeto Integrador Java, com o intuito desenvolver um sistema de API's (do inglês: "Application Programming Interface"). As API's interligadas uma com as outras teriam o objetivo de emular o funcionamento de um processo para a venda de produtos, com base na empresa de transporte que será responsável pelo calculo do frete e entrega de acordo com as características do produto.

Projeto Java

Atividade proposta

Foi proposto a implementação de um sistema completo de baseado na venda de produtos, utilizando-se da implementação de API's.

As funções do sistema foram distribuídas em microserviços para as equipes envolvidas na disciplina, sendo as seguintes distribuições:

- *Calculadora de Frete;*
- *Transportadora;*

Nesse documento teremos descrito o desenvolvimento das funções que foram atribuídas a nós, sendo elas *Transportadora* e *Calculadora de frete*.

Problemática

Dentro do escopo da *Transportadora* seria preciso que a criação de um formulário para cadastro de pedidos, e apos registrados, os pedido precisavam ser armazenados. Junto a isso deveria criar uma tela para alteração do status do pedido, e deveria também criar um histórico de status relacionado às alterações de status realizadas. Além disso, seria necessária a criação de uma tela para a consulta dos pedidos.

No escopo da *Calculadora de Frete* seria necessário armazenar um conjunto de endereços para precificar o valor baseado na distância e no peso do produto a ser entregue.

Todas estas funcionalidades seriam desenvolvidas em API's com o intuito de que os outros microserviços do sistema pudessem trocar os dados que possuem uns com os outros.

Entendemos por microserviços a partição de uma funcionalidade que pode ser dividida em várias atividades menores, que, ao final, essas atividades unificadas resultam na funcionalidade completa.

O motivo pelo qual essa prática foi adotada, é pela facilitação do desenvolvimento das funcionalidades, pois, podendo dividi-la possibilita direcionar times específicos para cada microserviço, o que agiliza o processo e posteriormente simplifica a manutenção.

Solução

Para iniciar o planejamento de desenvolvimento das API's, foi realizada a criação de diagramas para ter uma visão geral de como seria o funcionamento do microsserviços do sistema que iríamos criar.

Os diagramas criados foram:

- Fluxograma
 - Descreve o funcionamento de um processo de uma forma mais simples, mostrando suas etapas do início ao fim, como é mostrado nos anexos *A* e *B*.
- Diagrama de classes
 - Descreve a funcionalidade dos componentes e seus atributos e também a relação dos componentes entre si, como mostrado nos anexos *C* e *D*.
- Diagrama de Entidade e Relacionamento
 - Descreve o comportamento das entidades e relacionamento das mesmas dentro de um banco de dados, como visto nos anexos *E* e *F*.

Linguagens e Métodos de Desenvolvimento

As ferramentas utilizadas para efetuar o desenvolvimento do projeto são ferramentas simples, porém, de grande ajuda para facilitar e aprimorar o desenvolvimento de uma API/Webservice.

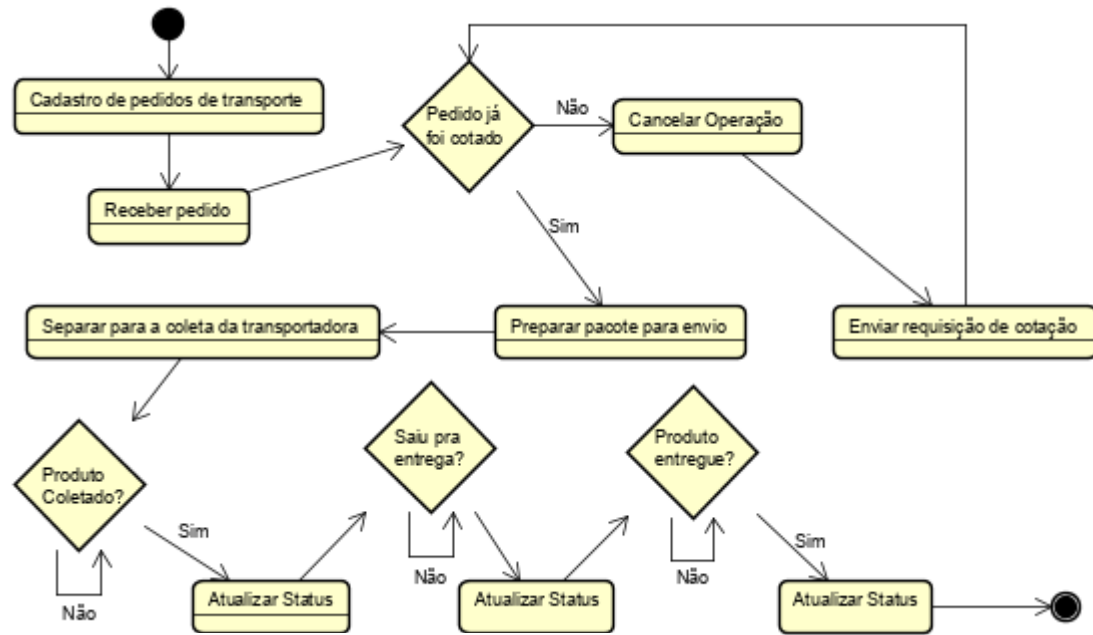
Utilizamos a linguagem Java como a principal para criação da API, pois o projeto trata-se de um projeto em Java. A IDE utilizada foi o Eclipse com um Projeto Maven, para versionar as dependências utilizadas, sendo elas: Devtools e MySQL Connection.

Usamos um framework chamado Spring para desenvolver o projeto, pois o mesmo além de facilitar a criação de uma API, já tem o JPA e algumas outras API's embutidas, fazendo com que a conexão com o banco de dados e a formatação para utilização dos métodos GET/POST/PUT/DELETE funcionem de forma mais simplificada.

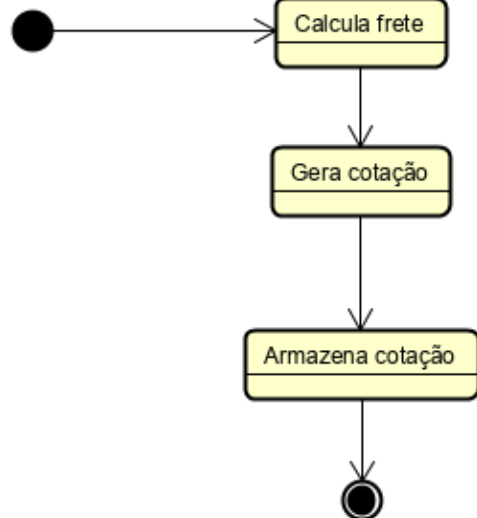
Para o desenvolvimento das conexões com as API's, usaremos JavaScript, pela simplicidade que a linguagem utiliza para trabalhar com objetos JSON e também pelo fato de funcionar muito bem com HTML e CSS que são usados para a criação das interfaces necessárias para as funcionalidades da transportadora.

Anexos

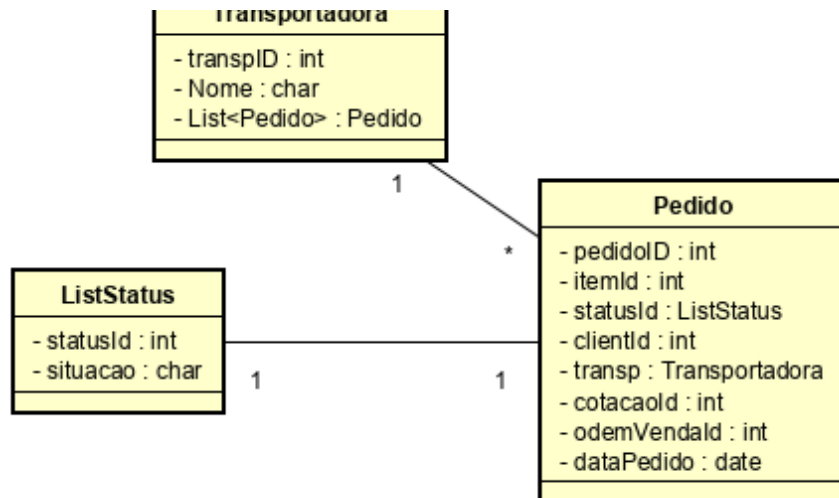
- Anexo A:



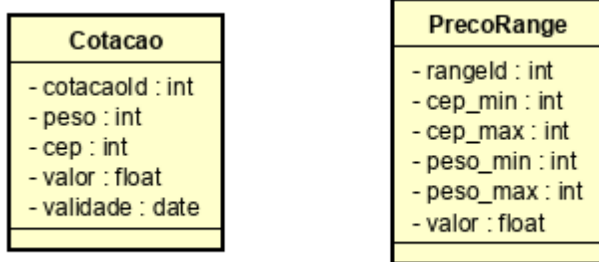
- Anexo B:



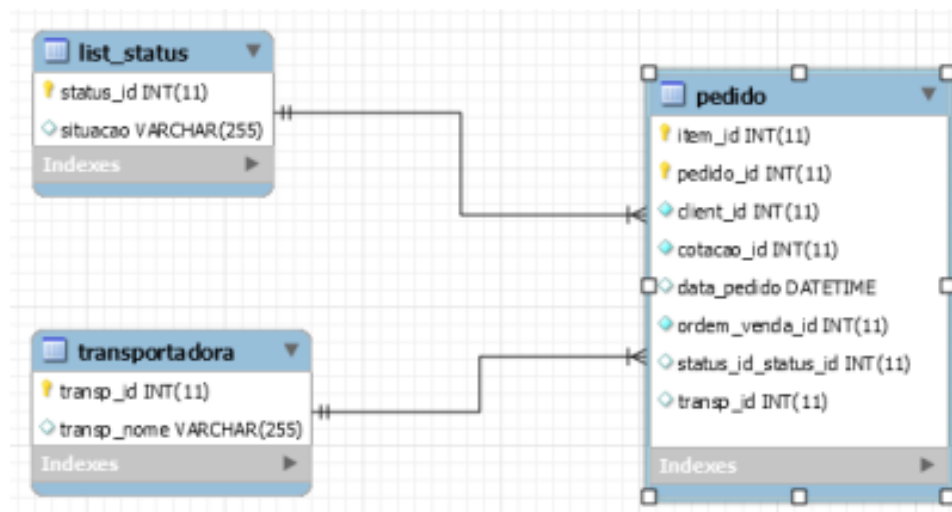
- Anexo C:



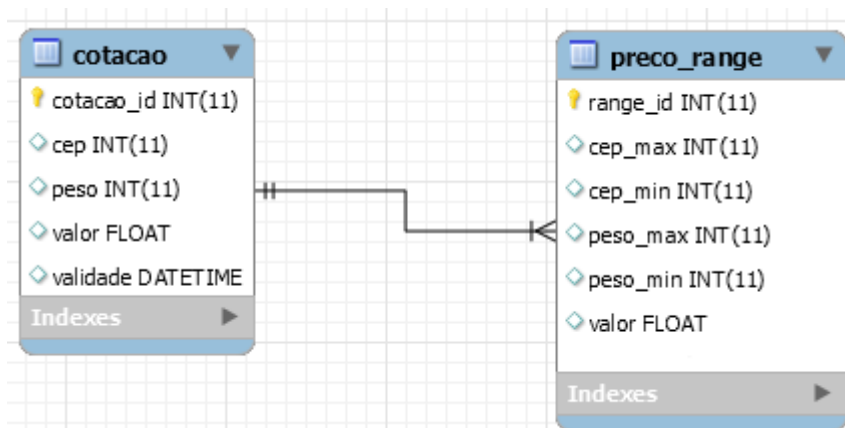
- Anexo D:



- Anexo E:



- Anexo F:



Conclusão

Através desse projeto, foi possível compreender e ter uma boa relação com a arquitetura de microsserviços e também uma noção de como funciona a junção dos microsserviços, através das API's conectadas aos webservices formando um todo.

No decorrer do processo, foi verificado o quão importante é a parte de diagramação de um projeto, pois após os diagramas estarem todos bem feitos, o desenvolvimento flui de forma mais descomplicada.

Por ultimo, observamos a facilidade que os microsserviços trazem para projetos maiores, pois cada parte passa a ser tratada de forma total e acaba sendo mais simples não somente de desenvolver, mas também de efetuar testes e a manutenção de modo geral.

Referências

Descrição e compreensão sobre microsserviços e API's:
<<https://vertigo.com.br>>. Acesso em março.

Compreensão e aprendizado sobre utilização do framework Spring:
<<https://www.baeldung.com>>. Acesso em março e abril.

Auxílio para correção de bugs durante o desenvolvimento do projeto:
<<https://www.guj.com.br>>. Acesso em março, abril e maio.
<<https://stackoverflow.com>>. Acesso em março, abril e maio.

Compreensão e aprendizado sobre a utilização de JSON e Javascript:
<<https://codepen.io>>. Acesso em abril e maio.
<<https://www.freecodecamp.org>>. Acesso em abril e maio.
<<https://shiffman.net>>. Acesso em abril e maio.
<<https://www.chillyfacts.com>>. Acesso em maio.
<<https://lengstorf.com>>. Acesso em maio.
<<https://egghead.io>>. Acesso em maio.