



WeLoveEat - Relatório

24/04/2019

Victoria Deorio
Campinas, São Paulo

Visão geral

De forma geral, o projeto que desenvolvi para o processo seletivo da Dextra, conta com todas as partes da realização de um projeto, desde a documentação até os testes unitários. E aqui está o porquê das minhas práticas de desenvolvimento

Objetivos

1. Deixar explícito as práticas que adotei e o porquê.
2. Deixar transparente meus conhecimentos e visões da realização de um projeto.

Especificações

O projeto foi dividido e estruturado em camadas, e entre elas as camadas: Controller, Services, Model e Dao.

- Camada Model

A camada Model é onde as classes de entidades ficam. Ou seja, meu projeto conta com VO's para ingredientes e pedidos que irão caracterizá-los para que eu possa trabalhar com eles de forma organizada.

- Camada Dao

A camada Dao geralmente contém as classes que mantêm uma conexão com o banco de dados, que fazem consultas, alterações, remoções e inserções no mesmo. Todos os dados que precisei utilizar no projeto (Nomes e valores de ingredientes, nomes de pedidos do cardápio e etc) ficaram alocados em memória, segundo a própria proposta do projeto. E, mesmo assim, achei válida a camada DAO, para poder organizar melhor na aplicação as funções de cada classe.

- Camada Service

A camada de service apresenta todas as regras de negócio do projeto, e é esta que realiza a lógica de cálculo de preço, verifica as promoções de um pedido e conversa com a camada DAO para resgatar dados, caso necessário.

- Camada Controller

A camada controller tem o papel de fazer a ponte entre o Front e o Backend. O front com JavaScript/JQuery faz requisições para a aplicação, requisições estas que serão tratadas no controller, chamando a camada service.

Spring Boot

Como um importante Framework do projeto, utilizei o SpringBoot em meu backend. Essa tecnologia foi escolhida de acordo com a recomendação e popularidade da mesma, diversos sites, fóruns e especialistas da área recomendam o seu uso. Ele é de grande ajuda na parte de configuração e integração do projeto.

Para utilizar o tomcat (Servidor web de Java) por exemplo, foi muito mais fácil e rápido com o Spring. Além disso, ele é um framework que não exige muito esforço para aprendizado, tem boa documentação e muito suporte em fóruns.

TomCat server

O apache tomcat foi escolhido pelo principal motivo de ter uma boa integração com o SpringBoot, não exigindo muito esforço de configuração.

Front-end

A parte do Front-end, página da web com que o usuário terá contato, foi desenvolvida com html, CSS, JS e JQuery puros. Entendo que existem diversos frameworks que poderiam tornar o desenvolvimento mais rápido, seguro e até mesmo com uma apresentação melhor do que as linguagens nativas.

O Angular, por exemplo, é uma tecnologia que poderia ter ajudado muito na realização dessa etapa. Contudo, nunca tive experiências com esse tipo de framework, apenas JSF e JSP. Me senti mais confortável em usar as linguagens nativas, as quais já utilizei muito para projetos acadêmicos ou até mesmo pessoais.

É uma projeção futura começar a entender e trabalhar com esse tipo de tecnologia.

Testes Unitários

De acordo com as exigências do projeto, este deve conter testes unitários de alguns casos específicos da aplicação. Assim, utilizei o suporte do próprio SpringBoot para testes.

O que é muito recomendado por alguns blogs e fóruns de desenvolvedores. Essa prática ajudou muito no mapeamento e resolução de bugs e falhas, que poderiam comprometer toda a aplicação

Marcos

I. Divisão de **Camadas**

A divisão do projeto em camadas e pacotes foi um marco, pois organizou a aplicação e a deixou bem estruturada, de modo que tenha facilitado todas as outras etapas do desenvolvimento.

II. Utilização do FrameWork **SpringBoot**

A utilização deste framework foi um marco para o projeto e pessoal. Sua utilização implicou em um resultado otimizado, organizado e ainda fácil de se implementar e usar. Com pouco domínio sobre o Spring, consegui estudar através de documentações e fóruns, e aprender a utilizá-lo, o que não favorece apenas o projeto mas também aos meus domínios sobre o framework.

III. Desenvolvimento e realização de **testes unitários**

A realização de um teste unitário foi algo novo, que eu nunca havia feito. O Spring ajudou bastante com este marco também, pois simplificou a maneira de se desenvolver e utilizar, fazendo com que mesmo com pouco domínio sobre a prática pudesse realizar os testes necessários.