



Atividade

Frontend + Backend

Grupo:
Matheus Lemes Tassara
Vitória Barbosa da Silva

São Paulo, 18 de novembro de 2022

Etapa 1:

Implementar 01 (uma) ou 02 (duas) telas estáticas usando ReactJS e NodeJS. A ideia inicial era que o protótipo atendesse os cenários web e app ao mesmo tempo. Em caso de não aderência do protótipo ao uso do ReactJS, a aplicação pode ser desenvolvida livremente ou com base em outro protótipo.

As atividades foram baseadas no Aplicativo TOP, um autosserviço para compra de bilhetes de embarque de trens do metrô.

No Frontend, desenvolvido em REACT, foi possível criar uma tela dashboard onde o usuário consegue ter visibilidade da quantidade de bilhetes disponíveis para uso, bilhetes já utilizados, e o saldo referente ao valor R\$ da quantidade de bilhetes disponíveis.

<https://github.com/vitoriabarbosas/pos/tree/main/Frontend%20%2B%20Backend/aplicativo-recarga-top>

Além disto, é possível realizar novas recargas que refletem na disponibilidade de bilhetes disponíveis para uso. Assim como, informar a quantidade de bilhetes que fora utilizado. Isto nos gera um extrato onde pode ser visualizado todas as transações.

Quantidade	Valor	Tipo
4	16	Disponível
1	4	Utilizado
1	4	Utilizado
1	4	Disponível
1	4	Utilizado

Exemplo simulando uma compra de 2 bilhetes no valor de R\$ 8,00

Quantidade

2

Valor

8

☒ Disponíveis

☐ Utilizado

NOVA RECARGA

Ao clicar em “Nova Recarga”, o sistema é atualizado com a informação referente a quantidade de bilhetes disponíveis e saldo monetário.

Bilhetes disponíveis

7

Bilhetes utilizados

3

Saldo

R\$ 28.00

Quantidade

Valor

☒ Disponíveis

☐ Utilizado

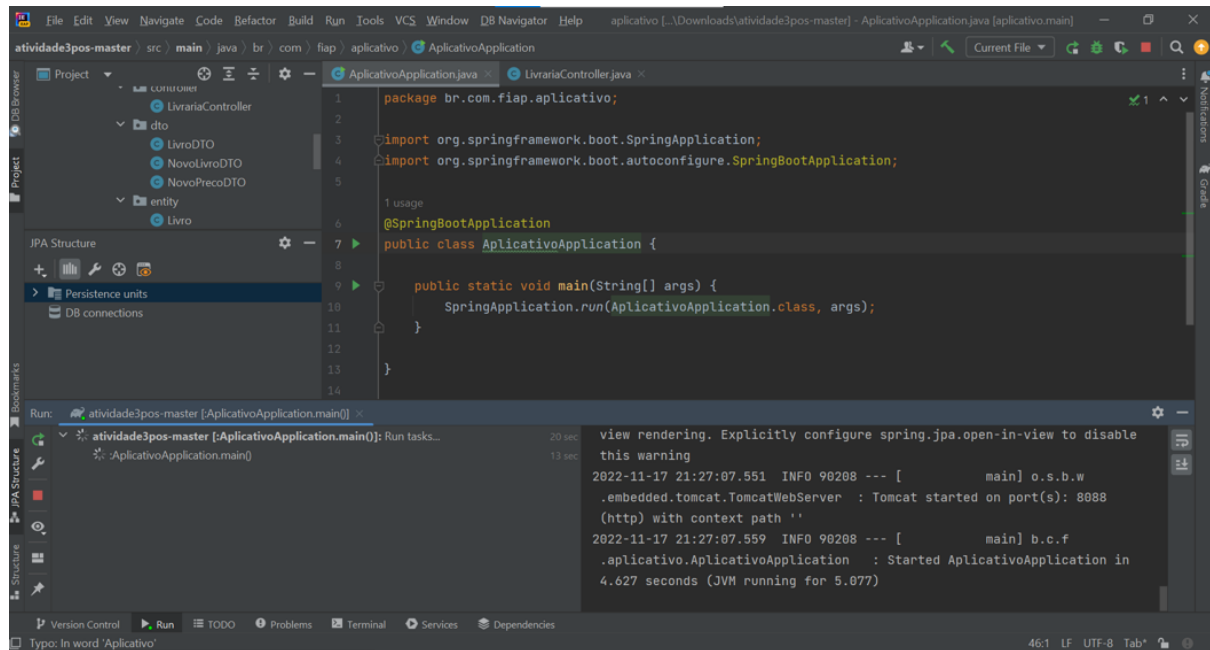
NOVA RECARGA

Quantidade	Valor	Tipo	
4	16	+	
1	4	-	
1	4	-	
1	4	+	
1	4	-	
2	8	+	

Etapa 2:

A segunda etapa consiste em criar 02 (dois) serviços de consulta ou cadastro. Na entrega deverá ser especificado o banco de dados escolhido e também devem ser enviados os scripts para geração das estruturas.

Atividade 2 foi implementada utilizando os conteúdo digital da plataforma FIAP ON com banco de dados jdbc:h2. Aqui, temos uma aplicação de biblioteca. https://github.com/mathtass/atv_pos



Aplicativo rodando onde URL a partir do id, retorna o livro.



Swagger funcionando:

← → ↻ localhost:8080/swagger-ui/#/livraria-controller/getLivrosUsingGET

GET /livros getLivros

Try it out

Parameters

Name	Description
titulo * required string (query)	titulo

titulo - titulo

Responses

Code	Description	Links
200	OK	No links

Media type
/

Controls Accept header.

Etapa 3:

A terceira etapa consiste em adaptar a aplicação React, desenvolvida na etapa 1, para consumir os web services desenvolvidos em Spring. O projeto deve ser versionado e, em seguida, modificado para consumir os serviços recém-implementados usando os protocolos REST ou SOAP.

Esta atividade foi realizada em django como backend e frontend, e banco de dados mysql.

<https://github.com/mathtass/atividade3>

Tela Inicial:



Sempre que clica no título aplicativo de recarga volta para a tela inicial

Após preencher o nome do usuário, ele verifica no banco de dados se já existe o usuário, se houver ele carrega os dados como saldo e número de bilhetes. Caso não haja, ele cria com o nome do usuário e traz seu saldo e número de bilhetes que vai ser 0.

Caso o usuário possua bilhetes, aparece o botão Usar Bilhete:



Caso não possua, aparece recarga:

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000'. The page has a dark brown header with the text 'APLICATIVO DE RECARGA' in white. Below the header, there is a white rectangular box containing the text 'Insira Nome Completo:' followed by a text input field with 'Vitoria Barbosa' and a button labeled 'Enviar'. Below this box, the text reads 'Bem vindo Vitoria Barbosa, Seu saldo é de R\$0 reais e você possui 0 bilhetes.' At the bottom of the page, there is a blue button labeled 'Recarga'.

Caso clique em recarga, chama outra url “/recarga” , onde é possível colocar quanto quer adicionar ao saldo:

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/recarga'. The page has a dark brown header with the text 'APLICATIVO DE RECARGA' in white. Below the header, the text reads 'Insira os dados e realize a recarga:'. This is followed by a white rectangular box containing the text 'Insira Nome Completo:' with an empty text input field, 'Insira Valor da Recarga:' with an empty text input field, and a button labeled 'Realizar Recarga'. Below this box, there is a blue button labeled 'Comprar Bilhete'.

Nessa tela, o usuário vai aumentar seu saldo, indicando quantos reais deseja adicionar. Ao realizar a recarga, ele acrescenta o valor ao saldo, e ao clicar em comprar bilhete, ele será redirecionado para a tela de comparar bilhetes.

Na tela de comprar os bilhetes, ao inserir o nome e quantidade de bilhetes, caso o usuario tenha o saldo (isso é visto através de uma consulta ao banco de dados) aparece “Compra Realizada” e é descontado 3 reais do saldo dele, caso ele não tenha, a compra não é realizada tendo em vista o saldo insuficiente.

← → 127.0.0.1:8000/comprar_bilhete

APLICATIVO DE RECARGA

Insira os dados e realize a compra:

Insira Nome Completo:
Matheus Lemes

Insira a Quantidade de Bilhetes:
1

Comprar Bilhetes

COMPRA REALIZADA

← → 127.0.0.1:8000/comprar_bilhete

APLICATIVO DE RECARGA

Insira os dados e realize a compra:

Insira Nome Completo:
Matheus Lemes

Insira a Quantidade de Bilhetes:
8

Comprar Bilhetes

SALDO INSUFICIENTE

Caso o usuário tenha saldo, na tela inicial aparece o botão de usar bilhete, caso clique, ele usará o bilhete e descontará do número de bilhetes dele.



A última tela é para mostrar o nosso banco de dados, ela mostra todos os clientes cadastrados e seu saldo respectivo.

APLICATIVO DE RECARGA		
NOME CLIENTES	NÚMERO DE BILHETES	SALDO EM REAIS
Gabriela Souza	0	0
Matheus Lemes	38	4
Vitoria Barbosa	1	3
Vitoria Barbosa da Silva	0	10

Referências:

[1] Material disponibilizado pela plataforma FIAP ON - Front-end + Backend