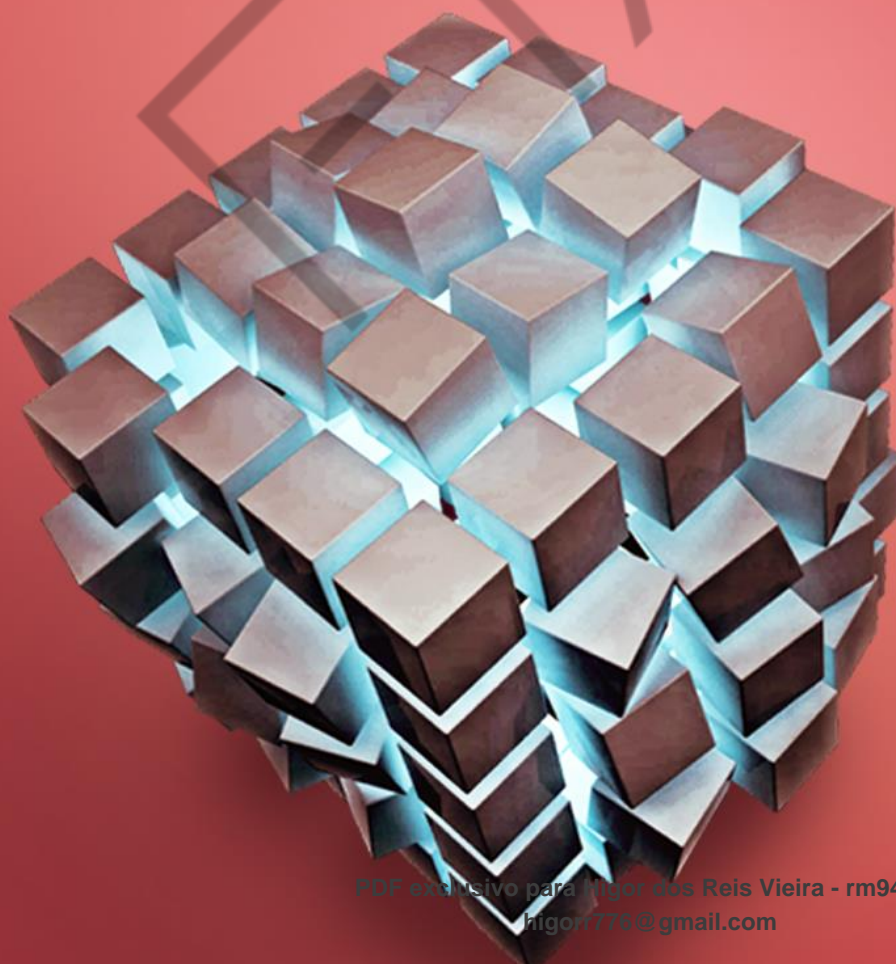


INTEGRATION

# *JUNTANDO* **TUDO!**

HENRIQUE RUIZ POYATOS NETO



01

## SUMÁRIO

1 JUNTANDO TUDO!.....	3
1.1 Capítulo 1: Juntando tudo!.....	4
1.2 Capítulo 2: Mais dinamismo na interface com o usuário .....	4
1.3 Capítulo 3: JSP, a interface do usuário em Java.....	4
1.4 Capítulo 4: Frameworks em Java! .....	5
1.5 Capítulo 5: A alternativa do Facebook.....	5
1.6 Capítulo 6: Esse estudo de caso pode te ajudar .....	6
1.7 Capítulo 7: Grand finale.....	6

## 1 JUNTANDO TUDO!

Seja bem-vindo à última fase do Fintech! Estamos na reta final de nosso projeto e aprenderemos aqui os últimos conceitos essenciais que lhe permitirão desenvolver um sistema web completo.

Esta fase é chamada de *Integration*, ou “Integração”. Você verá de que maneira todas as tecnologias aprendidas até aqui, como a linguagem de programação Java, se integram com o banco de dados e de que forma ambos se integram com a interface web e, por consequência, com o usuário de nosso sistema.

Mais ao final da fase, aprenderemos o conceito de MVC (*Model-View-Controller*), que é uma maneira muito adequada de organizar nosso código-fonte, facilitando a sua manutenção e sustentação. O conceito é apresentado e veremos, na prática, como essas três camadas realmente funcionam.

O grande desafio desta fase será essa grande integração do Fintech: a camada de modelo de negócio, conhecida como *Model*, ou simplesmente o “M” da sigla MVC, foi construída na fase 5 e aprimorada na fase 6, por meio das classes do tipo *entity* e outras, como o *design pattern* DAO, que será responsável pela intermediação com o banco de dados. Nossa camada de interface com o usuário, conhecida como visão ou *View* – o “V” da sigla MVC –, foi iniciada na fase 4 e será retomada nesta última fase, com códigos de programação em HTML/JSP (*Java Server Pages*), EL (*Expression Language*) e JSTL (*Java Server Pages Standard Tag Library*) que permitirão a integração com a camada de intermediação conhecida como controle, controladora ou *Controller* – o “C” que completa a sigla MVC –, que será realizada com as *Servlets*, **vale ressaltar que, apesar de parecer algo muito antigo, os componentes Servlets e JSP em uma aplicação Java Web são a base para os componentes utilizados por outros frameworks MVC como JSF, Struts e Spring MVC, por isso a importância desse conteúdo no início do aprendizado de uma aplicação web.** Façamos agora um breve comentário a respeito de cada capítulo desta importante fase.

## **1.1 Capítulo 1: Juntando tudo!**

Você está bem aqui! Este capítulo lhe dá as boas-vindas a esta importante fase, esclarece o que ela contém e quais são os objetivos acadêmicos e de projeto.

## **1.2 Capítulo 2: Mais dinamismo na interface com o usuário**

O capítulo 2 marca a retomada da camada de visão que foi iniciada na fase 4. Embora sejam muito úteis, as linguagens HTML, CSS e JavaScript, mesmo que sejam acompanhadas do framework Bootstrap, deixam as interfaces muito estáticas, ou seja, não permitem a atualização de informações em tempo de execução, exigindo a intervenção de um desenvolvedor para tal.

Você aprenderá, nesse capítulo, como funciona uma Servlet, um componente que provê a integração entre a linguagem de programação Java e as tecnologias web que foram apresentadas na fase 4. As Servlets e todas as tecnologias associadas a elas, apresentadas nos capítulos posteriores, precisam ser armazenadas em um Java Web Container, e aqui recuperamos o Apache Tomcat que foi instalado e integrado ao Eclipse na primeira fase.

## **1.3 Capítulo 3: JSP, a interface do usuário em Java**

Novamente, o título do capítulo entrega seu objetivo: embora a Servlet promova uma integração muito interessante entre a linguagem de programação Java e as tecnologias web, ela não se apresenta de maneira tão prática quanto em linguagens ASP (Active Server Pages, da Microsoft) ou na linguagem PHP, as quais permitem intercalar suas estruturas de programação diretamente no HTML, apenas nos pontos em que se busca o dinamismo. A tecnologia Java Server Pages (JSP) é a evolução que traz essas facilidades. Mas não se engane: o Java Web Container (em nosso caso, o Tomcat) transformará o JSP em uma Servlet e o compilará, ou seja, todas as regras apresentadas no capítulo anterior são aplicáveis por aqui.

## 1.4 Capítulo 4: Frameworks em Java!

Apesar da velocidade de desenvolvimento que se atinge usando a tecnologia JSP, alguns procedimentos podem, no entanto, exigir ainda muitas linhas de código para cumprirem seu objetivo. Assim como o framework jQuery economiza muitas linhas de código JavaScript, aumentando a produtividade do desenvolvedor web, a *Expression Language* (EL) e as *Java Server Page Standard Tag Libraries* (JSTL), ou simplesmente *TagLibs*, desempenham esse papel no JSP: com elas, você fará muito mais, escrevendo menos, e você saberá mais sobre elas por aqui. O capítulo traz, ainda, o conceito de MVC e um primeiro exemplo de código que passa por estas três camadas: *Model-View-Controller* (MVC).

E existe a possibilidade de ficar mais veloz?

Sim, existe, e você terá contato com isso no futuro, um exemplo disso são as aplicações web desenvolvidas com o framework Spring MVC, que disponibiliza componentes, configurações e anotações que reduzem muito nossa codificação. Apenas como curiosidade, o framework Spring MVC utiliza JSP, Servlets, JSTL e EL como a base dos seus componentes.

## 1.5 Capítulo 5: A alternativa do Facebook

Se pensarmos em tudo o que foi abordado desde a primeira fase do projeto até a última, é impressionante a quantidade de conteúdos e tecnologias que precisam ser combinados para fazer um sistema web funcionar. É natural que surjam dúvidas de como integramos tudo isso.

O capítulo 5 foi concebido com isso em mente: integrar tudo o que foi realizado do início ao fim. Faremos aqui um estudo de caso usando duas entidades, desde a criação de suas estruturas no banco de dados Oracle, a criação da interface com o usuário, o projeto no Eclipse até a construção das três camadas, *Model-View-Controller*. As quatro operações em um CRUD (sigla para **C**reate, ou a inserção da informação, **R**etrieve, ou a recuperação/consulta da informação, **U**ppdate, ou atualização da informação, e o **D**elele, ou remoção da informação) serão abordadas

uma a uma, em forma de passo a passo, utilizando Java, Servlets, JSP, EL, JSTL, HTML, CSS, JavaScript, Bootstrap, ou seja, tudo o que aprendemos até aqui. Temos certeza de que esse estudo de caso vai ajudá-lo.

## 1.6 Capítulo 6: Esse estudo de caso pode te ajudar

Este capítulo aborda a biblioteca ReactJS para a construção de novos layouts. É mais um conhecimento para você adicionar à sua lista de habilidades. O ReactJS é uma biblioteca JavaScript muito utilizada no Facebook, Instagram, Netflix, Airbnb, entre outras.

## 1.7 Capítulo 7: Grand finale

Tudo precisa ter um fim, e este é o capítulo final do projeto Fintech. Faremos uma retrospectiva de tudo o que foi visto até aqui!

Não se preocupe, estamos aqui para ajudá-lo!

Surgiram dúvidas? Procure-nos imediatamente! Em nosso canal no Teams, nossos tutores e até seus colegas de turma podem ajudá-lo em sua jornada até o fim do projeto. Se quiser se aproximar ainda mais, teremos as nossas tradicionais *lives*, assim você pode fazer a sua pergunta ao vivo. Bons estudos nesta reta final! Vamos lá, falta pouco!