



Roteiro de Aula Prática

**Programação web para
Back-End**

Disciplina: Programação web para back-end

ROTEIRO DE AULA PRÁTICA 1

Unidade:

Aula (White Label)/Seção (KLS):

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software de ambiente de desenvolvimento para codificação de páginas web em JSF (Java) com Eclipse IDE.

Descrição do software:

Eclipse é uma IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e Kotlin. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software.

ATIVIDADE PRÁTICA 1

Atividade proposta:

Desenvolver um pequeno projeto de um currículo web utilizando HTML5 e CSS3. Note que ainda não deverá ser utilizado JSF. Nessa primeira atividade, pretende-se familiarizar com o ambiente de desenvolvimento, utilizando linguagem básica de programação web.

Objetivos:

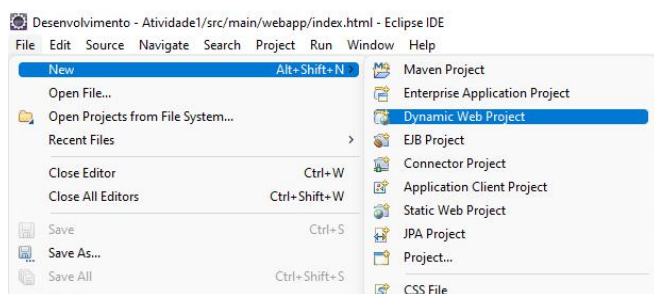
- Criar familiaridade com ambiente de desenvolvimento Eclipse com servidor WildFly.
- Construir estrutura de um projeto web no ambiente de desenvolvimento java, utilizando as linguagens HTML e CSS, para web, com Eclipse;
- Desenvolver um um pequeno projeto de criação de um currículo web.
- Compreender o processo de configurações necessárias para funcionamento do ambiente de desenvolvimento Eclipse.

Procedimentos para a realização da atividade:

Para cumprir com o proposto dessa aula prática, é necessário ter previamente instalado o Eclipse e o Temurin. Tendo isso garantido, crie uma forma de organização de pastas para receber os arquivos de seu projeto.

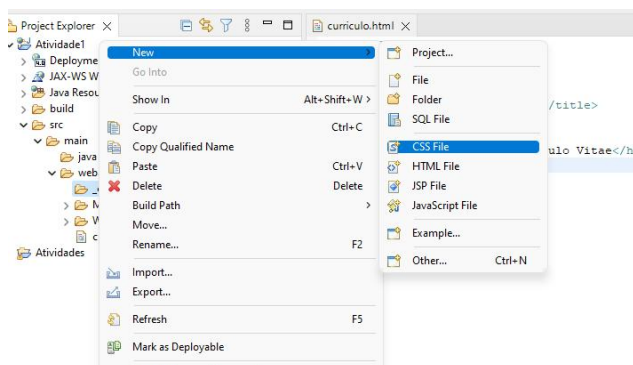
1 - Criar dentro da pasta eclipse-workspace, definida durante o processo de instalação da ferramenta, uma subpasta chamada "Atividade1", para o armazenamento dos arquivos desse

projeto. Crie um novo projeto indo ao menu File > New > Dynamic Web Project.



2 - Depois do projeto criado, clique com o botão direito em cima da pasta webapp (que fica em src > main), escolha a opção New e HTML File, para criar um arquivo do tipo HTML. Siga os passos sugeridos pela ferramenta e nomeie o arquivo como index.html.

3 - Crie uma nova pasta clicando com o botão direito em cima de webapp e coloque o nome de _css. Dentro dela, crie um arquivo de estilo, clicando com o botão direito sobre ela (_css), New > CSS File.



4 - Construa o código em HTML para criação de um mini projeto de currículo web com seus dados pessoais. Para isso, previamente, é necessário reunir todas as informações necessárias para isso. Escolaridade, Dados pessoais, Cursos, Experiências profissionais, e etc. Além disso, escolha uma bela foto para compor o seu currículo web. Note que toda a parte de estilo de seu projeto deverá ser implementada com CSS. Coisas como tipo de fonte, cores, tamanho e etc, tudo deve ser implementado em CSS. Não esqueça que será necessário ligar seu arquivo e estilo ao seu arquivo HTML.

5 - Para que a etapa anterior possa ser completada com êxito, você precisará utilizar o elemento <link>, do HTML, para linkar o arquivo de estilo ao seu projeto. Para isso, basta inserir a linha a seguir, entre os elementos <head> </head> do seu projeto, assim:

```
<head>
    <meta charset="ISO-8859-1">
    <title>Atividade1</title>
    <link rel="stylesheet" href="_css/style.css">
</head>
```

6 - Dentro do arquivo index.html, você deverá:

- Conferir se a estrutura do HTML foi criada adequadamente no momento da criação do arquivo;
- Utilizando elementos simples como elementos de parágrafo, de hierarquias (<h1> a

<h6>), e outros de seu conhecimento, compor o seu projeto de modo que ele contenha suas informações essenciais para um currículo.

- c) Organizar seus dados em sessões, como: Dados pessoais, Experiências profissionais, Formação, Trabalhos voluntários, e outros.
- d) Utilize hiperlinks para colocar na sessão de Dados pessoais, links para redes sociais como LinkedIn e Instagram.
- e) Insira uma foto e ajuste-a para que ela fique do lado direito da tela, ou seja, o lado oposto aos dados contidos na sessão "Dados pessoais".

7 - O arquivo de estilo, deve se chamar style.css para padronização. No entanto, é possível nomeá-lo de outras formas.

8 - Dentro do arquivo de estilo, certifique-se de iniciar o arquivo com a regra correta para especificar a codificação de caracteres a ser utilizada na folha de estilo.

9 - Organize os seletores por ordem, de forma organizada e devidamente indentado. Tente organiza-lo de acordo com a estrutura do seu HTML. Por exemplo, se em sua estrutura HTML você tem a utilização de um elemento <h1>, em seguida de um elemento <h4> e por fim, um elemento <p>, organize os seletores nessa mesma ordem. Pode não parecer, mas faz toda diferença quando se trabalhar em um projeto mais robusto. Adicione as regras de estilos de seu interesse em seu currículo web.

Checklist:

- 1 - Download das ferramentas e preparação do ambiente de desenvolvimento com Eclipse.
- 2 - Criação de um projeto e organização dos arquivos e pastas criados.
- 3 - Criação de arquivo HTML e construção do projeto utilizando elementos simples para composição de um currículo web;
- 4 - Criação de uma pasta para armazenar o arquivo de estilo e criação do arquivo de estilo para escrita das regras de estilização.
- 5 - Configuração do servidor Wildfly para apresentação do resultado do projeto pelo navegador de internet, utilizando o referido servidor.

Resultado: Aluno, você deverá entregar:

Construção de um projeto de currículo web com estrutura utilizando linguagem de marcação HTML5 e linguagem de estilo CSS3, a partir da correta configuração do ambiente de desenvolvimento Eclipse, com Temurin e o servidor de aplicações WildFly para apresentação do projeto desenvolvido em um browser. Todo o projeto deverá contar com estrutura de pastas sugerida na criação de projeto do Eclipse e, adicionalmente, com uma pasta _css que abrigará o arquivo de estilos, um arquivo de HTML e uma imagem.

Referências:

Alura - Novo projeto web usando eclipse. Disponível em: <<https://www.alura.com.br/apostila-java-web/novo-projeto-web-usando-eclipse>>. Acesso em: 18, agosto de 2022.

ROTEIRO DE AULA PRÁTICA 2

Unidade: Digite aqui

Aula (White Label)/Seção (KLS): Digite aqui

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software de ambiente de desenvolvimento para codificação de páginas web em JSF (Java) com Eclipse IDE.

Descrição do software:

Eclipse é uma IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e Kotlin. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software.

ATIVIDADE PRÁTICA 2

Atividade proposta:

Criar um projeto utilizando JavaServer Faces, onde deverá ser construído um formulário de contato, semelhante aos formulários utilizados na grande maioria das empresas que contam com sistema e-commerce. Empresas como: Americanas, Magazine Luiza e outros, contam com um modelo de formulário semelhante ao proposto nessa atividade.

Objetivos:

Compreender o processo de criação de projeto web para utilização do framework JavaServer Faces;

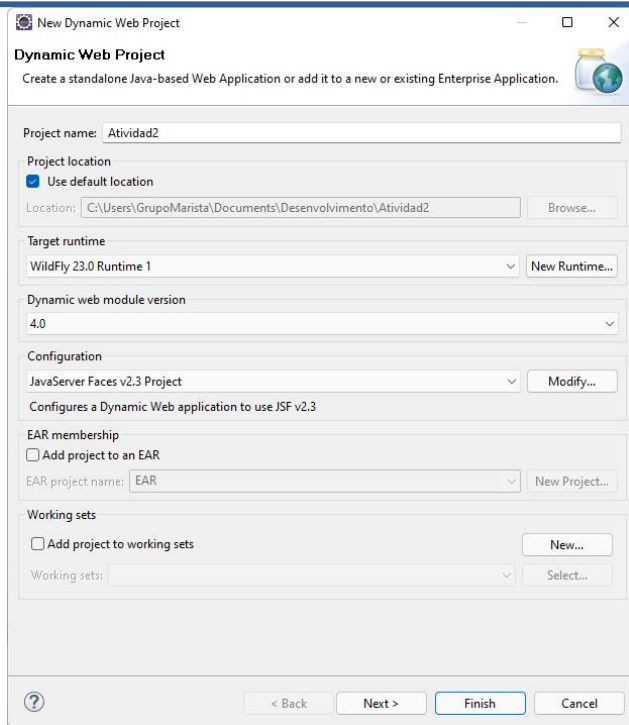
Realizar configurações básicas do JSF para o projeto em questão;

Construir formulário de cadastro utilizando JSF.

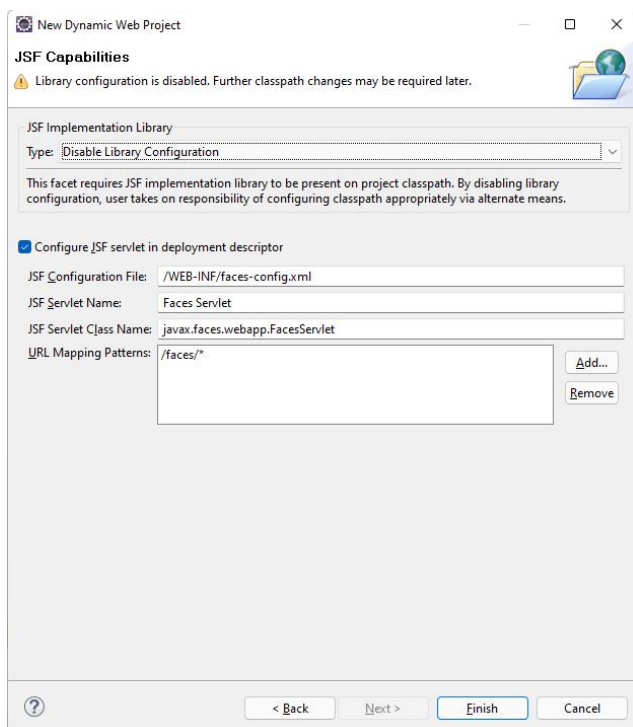
Procedimentos para a realização da atividade:

Será necessário a construção de um projeto novo. Sugere-se utilizar o nome Atividade2 para efeito de organização considerando esse roteiro de atividades. O processo de criação do referido projeto é semelhante ao da atividade1, contudo agora, será necessário configurar o JSF para utilizá-lo nesse projeto.

O JSF utilizado neste projeto, foi o de versão 2.3. Veja:

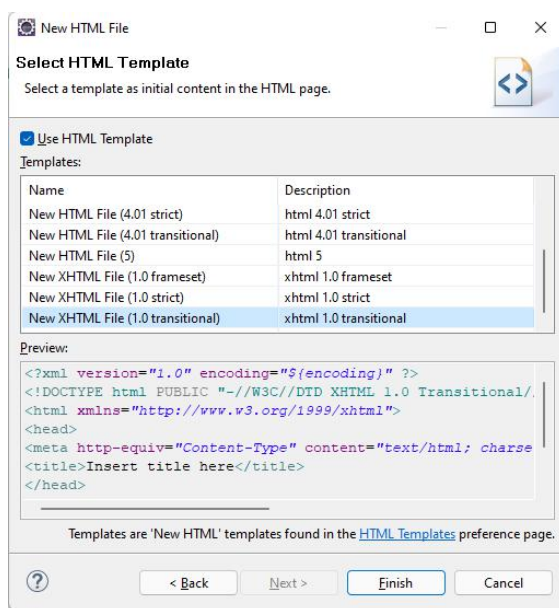
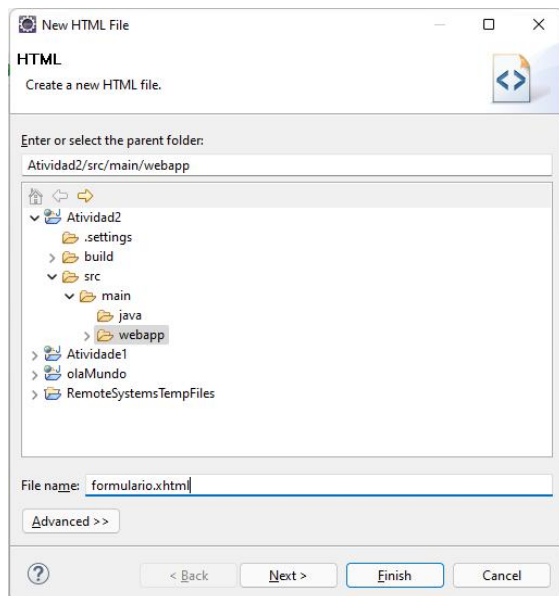


1 - Em seguida, deve-se realizar a configuração do campo de implementação de biblioteca, o “JSF Implementation Library” e deixá-lo desativado, como na imagem a seguir:



2 - Será necessário instalar a biblioteca do JSF no projeto. Para isso, deve-se acessar o endereço: <https://maven.java.net/content/repositories/releases/org/glassfish/javax.faces/2.3.0/> e produzir a versão javax.faces-2.3.0.jar, a escolhida para esse projeto. Baixe e coloque o arquivo na pasta lib (para isso, basta copiar e colar o arquivo na pasta).

3 - Duas configurações novas devem ser feitas no momento de construir uma página HTML. Agora, iremos criar um arquivo de nome formulario.xhtml e na tela seguinte, o template a ser usado é “New XHTML file (1.0 transitional)”, veja nas imagens:



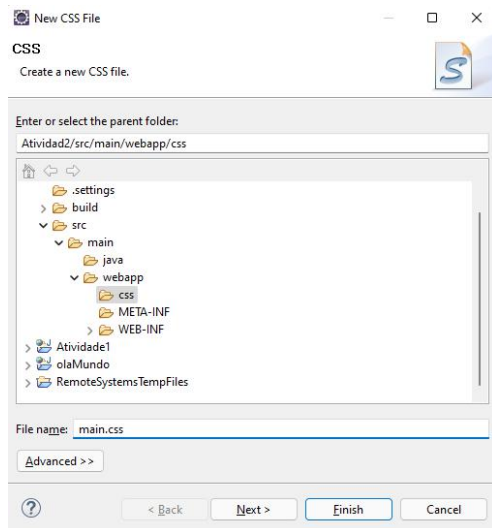
4 - Agora será necessário fazer uma pequena alteração na estrutura de código criada no arquivo. Note, na linha 3, que será necessário fazer uma inclusão, adicionando o seguinte código:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
```

5 - Será necessário criar o código utilizando essa “variável” h utilizada no escopo do projeto. Defina, pelo menos, 4 campos em seu formulário para informações como: Nome, Sobrenome, Fone e e-mail.

6 - Crie o arquivo de CSS, dentro de uma pasta com o mesmo nome só que escrito tudo em minúsculo, _css. O nome do arquivo, dessa vez, pode se main.css. Em seguida, faça as

configurações de estilo para tamanho de campos input, que nesse caso se chamam `<h:inputText>` e adicione a propriedade `styleClass` aos elementos no arquivo `.XHTML`, esta, será a responsável por aplicar a estilização desejada. Veja:



7 - Cuidado para realizar a indentação do código corretamente. Basta utilizar a indentação criada pelo próprio editor.

8 - Adicione o projeto ao servidor WildFly clicando com o botão direito em cima do nome do projeto, "Run As" > "Run on Server", escolha o WildFly na lista de servidor localhost que aparecerá na tela e pressione o botão Finish.

9 - Para visualizar o resultado de seu projeto, basta clicar com botão direito em cima do nome do arquivo XHTML, nesse caso, `formulario.xhtml`, "Run As" > "Run on Server", escolha o WildFly na lista de servidor localhost que aparecerá na tela e pressione o botão Finish. Note que ao abrir o navegador padrão em seu computador, o caminho do servidor apresentará o nome faces, como isso: `<localhost:8080/Atividade2/faces/formulario.xhtml>`. Isso é necessário para que a biblioteca JSF seja carregada no seu projeto.

Checklist:

- 1 - Ambiente de desenvolvimento Eclipse, JDK e WildFly devidamente configurado e funcionando.
- 2 - Criação de um projeto e organização dos arquivos e pastas criados.
- 3 - Download e configuração da biblioteca `javax.faces-2.3.0.jar` no projeto, para utilização do framework JavaServer Faces.
- 3 - Criação de arquivo XHTML e construção do projeto para criação de um formulário de consulta utilizando JavaServer Faces.
- 4 - Criação de uma pasta css e do arquivo `main.css` para codificação da folha de estilo utilizada no projeto.
- 5 - Utilização dos principais elementos javaServer Faces para construção de um formulário simples de contato.

Resultados da aula prática: Aluno, você deverá entregar:

Criação, estrutura, organização e codificação de um projeto simples de construção de um formulário de contato, utilizando o framework JavaServer Faces de forma integrada ao CSS para estilização do referido formulário. Além disso, será necessário levantar o projeto utilizando o chamamento do framework mencionado na URL a ser digitada no navegador, incluindo o padrão de mapeamento, o *URL Mapping Patterns* que deve ser configurado no momento de criação do projeto.

Referências:

Tutorial das Ferramentas JSF - Construir um Aplicativo JSF 2.0. Disponível em: <<https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=started-jsf-tools-tutorial-jsf-20>>. Acesso em: 18, agosto de 2022.

Java Server Faces Technology. Disponível em: <<https://www.oracle.com/java/technologies/javaserverfaces.html>>. Acesso em: 18, agosto de 2022.

ROTEIRO DE AULA PRÁTICA 3

Unidade: [Digite aqui](#)

Aula (White Label)/Seção (KLS): [Digite aqui](#)

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software de ambiente de desenvolvimento para codificação de páginas web em JSF (Java) com Eclipse IDE.

Descrição do software:

Eclipse é uma IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e Kotlin. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software.

ATIVIDADE PRÁTICA 3

Atividade proposta:

Implementar um pequeno projeto para criação de um formulário utilizando JavaServer Faces e ManagedBean para cadastro de nome de usuário e, utilizando ArrayList, imprimir na tela do navegador o nome cadastrado. O projeto deve utilizar linguagem CSS para estilização dos campos.

Objetivos:

Compreender e utilizar JavaServer Faces, ManagedBeans e ArrayList para inserção de nome de usuários numa lista de nomes e exibir no navegador, no momento de sua inserção, o nome inserido.

Procedimentos para a realização da atividade:

Para realização dessa atividade, será necessário fazer uso do ManagedBeans. Nesse momento, utilizaremos a bean @Named e @RequestScoped. É preciso frisar que a primeira, substituiu a @ManagedBean que foi descontinuada.

1 - Crie um novo projeto, semelhante à construção realizada nas atividades anteriores. Vá em File > New > Dynamic Web Project.

2 - Dê o nome de CadastroUsuarios ao projeto, marque o campo "Target Runtime" com Wildfly 23.0 e, em seguida, no campo "Configuration", sete para a opção JavaServer Faces v2.3 Project.

3 - Avance até a tela que mostra o campo "JSF Implementation Library" pois deixaremos esse campo marcado com a opção "Disable Library Configuration". Isso, porque copiaremos a

biblioteca `java.faces-2.3.0.jar` para a pasta `src/webapp/WEB-INF/lib`, como nos projetos anteriores.

4 - Agora, vamos criar a pasta `css` dentro de `src/webapp/` e, dentro da pasta `css`, criaremos o arquivo `main.css`.

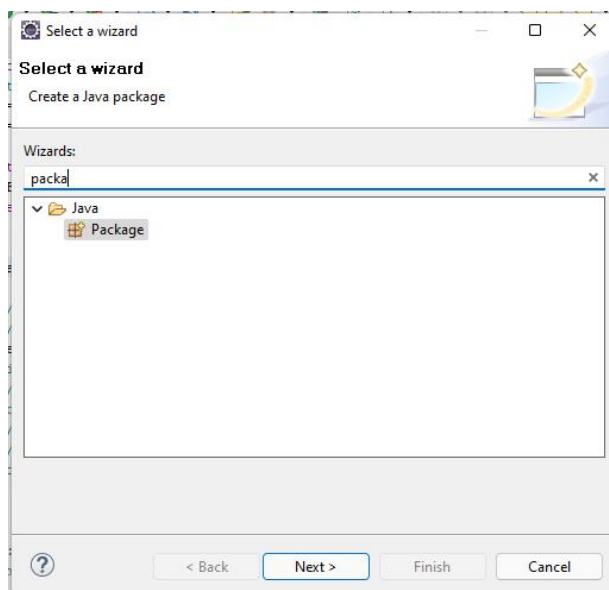
5 - Em seguida, ainda na pasta `src/webapp/` vamos criar o arquivo XHTML com nome `cadastro.xhtml`, clicando com o botão direito sobre a mencionada pasta, New, HTMLFile. Na tela que surge, você deverá substituir o nome do arquivo para `cadastro.xhtml` e, em seguida, clicar no botão Next. Na lista de templates que aparece, escolha a opção *New XHTML File (1.0 transitional)*. Não esqueça de alterar a linha do elemento `<html>` do arquivo para:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
```

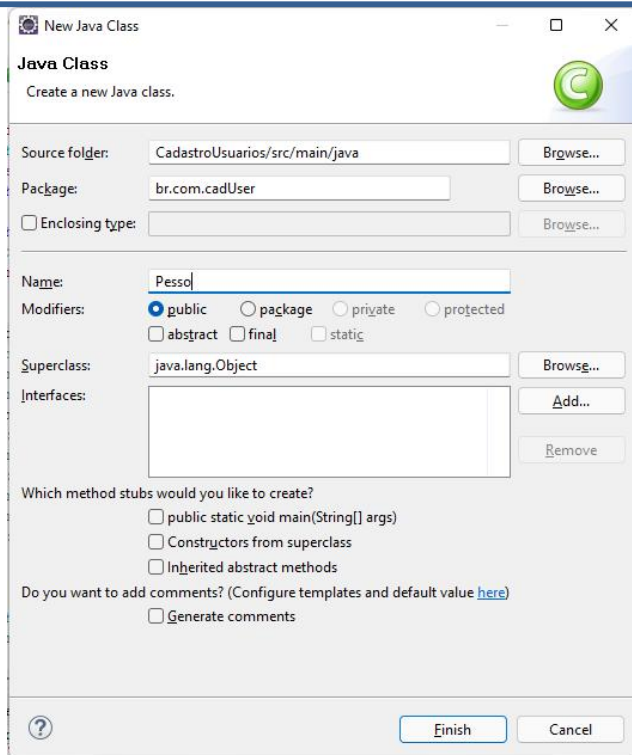
6 - No arquivo `cadastro.xhtml`, você deverá construir um código, que deverá contar com apenas um campo para inserção e nome.

7 - Agora é a hora de construir estilo de página para o seu arquivo. Vá para o arquivo `main.css` e, utilizando as propriedades de seu interesse, formate o seu campo utilizando recursos de cores, tamanhos, fontes e etc.

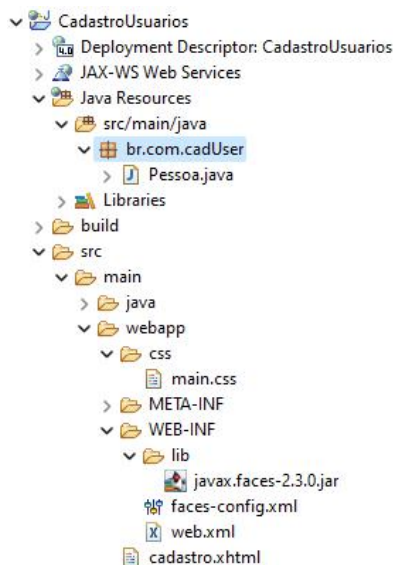
8 - Crie agora um arquivo de classe java em seu projeto. Mas antes, é necessário criar um pacote. Para isso, vá para JavaResources, `src/main/java`, clique com o botão direito sobre essa opção Other. Na janela que abrirá, pesquise no campo Wizards pelo termo `package`, como na imagem a seguir:



9 - Na tela seguinte, você poderá definir um nome. Então, utilize `br.com.cadUser`. Clique em Finish e, em cima do pacote criado, clique com o botão direito do mouse, New, Class dê o nome Pessoa no campo que aparecerá. Não altere nenhuma outra configuração. Veja:



10 - A raiz de diretório deverá ficar da seguinte forma:



11 - Finalmente, crie o código da classe, um atributo nome do tipo privado e um atributo privado do tipo ArrayList, para armazenar os nomes digitados. Não esqueça de criar os métodos getters e setters de ambos os atributos.

Checklist:

- 1 - Ambiente de desenvolvimento Eclipse, JDK, pacote e WildFly devidamente configurado e funcionando.
- 2 - Criação de um projeto e organização dos arquivos e pastas criados.
- 3 - Download e configuração da biblioteca javax.faces-2.3.0.jar no projeto, para utilização do

framework JavaServer Faces.

3 - Criação de arquivo XHTML e construção do projeto para criação de um formulário com um campo para preenchimento de nome.

4 - Criação de um pacote.

5 - Criação de uma pasta css e do arquivo main.css para codificação da folha de estilo utilizada no projeto.

6 - Criação de um arquivo de classe java para compor o back-end do projeto.

7 - Utilização dos principais elementos javaServer Faces para construção de um formulário simples de contato.

8 - Utilizar notações bean @Named e @RequestScoped para funcionamento do projeto.

Resultados da aula prática: Aluno, você deverá entregar:

Criação, estrutura, organização e codificação de um miniprojeto de página web com formulário contando com um campo para preenchimento de nome. O referido formulário deverá exibir o nome digitado logo abaixo, utilizando ArrayList que deverá ser implementado com framework JavaServer faces de forma integrada ao CSS para estilização do referido formulário. Além disso, será necessário empregar notações de ManagedBean no projeto e configurar o chamamento do framework mencionado na URL a ser digitada no navegador, incluindo o padrão de mapeamento, o *URL Mapping Patterns* que deve ser configurado no momento de criação do projeto.

Referências:

Criando e configurando um projeto web - JSF 2, Primefaces 3 e CDI. Disponível em: <<https://www.devmedia.com.br/criando-e-configurando-um-projeto-web-jsf-2-primefaces-3-e-cdi/25251#:~:text=Para%20configurar%2C%20clique%20com%20o,em%20OK%2C%20confirm e%20figura%208.>>. Acesso em: 18, agosto de 2022.

ROTEIRO DE AULA PRÁTICA 4

Unidade: [Digite aqui](#)

Aula (White Label)/Seção (KLS): [Digite aqui](#)

SOFTWARE

☒ Software / ☐ Acesso on-line

☐ Pago / ☒ Não Pago

Infraestrutura:

Computador;

Software de ambiente de desenvolvimento para codificação de páginas web em JSF (Java) com Eclipse IDE.

Descrição do software:

Eclipse é uma IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e Kotlin. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software.

ATIVIDADE PRÁTICA 4

Atividade proposta:

Configurar a integração da ferramenta Eclipse com o sistema de controle de versão git juntamente com a plataforma de hospedagem de código fonte e arquivos, github.

Objetivos:

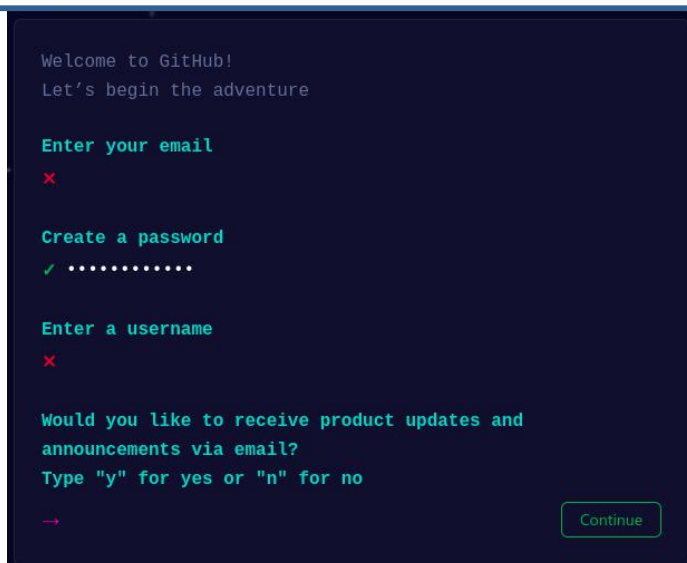
- Compreender os detalhes necessários para configuração e integração da ferramenta eclipse com sistema de controle de versão Git e a plataforma de hospedagem de código-fonte, github.
- Realizar as etapas de criação de repositório local e remoto e realizar registros de commits de alteração desses repositórios.
- Configurar repositório git em projetos já criados na IDE Eclipse.

Procedimentos para a realização da atividade:

Para realização dessa atividade, será necessário ter configurado o ambiente de desenvolvimento Eclipse com projetos já criados. Além disso, é importante ter conta na plataforma github criada.

1 - Vamos criar a conta na plataforma github. Inicialmente, acesse a plataforma em: <https://github.com/> e, em seguida, clique no botão Sign up que aparecerá na tela.

2 - Na sequência será necessário preencher alguns campos como:



Welcome to GitHub!
Let's begin the adventure

Enter your email
✖

Create a password
✔
.....

Enter a username
✖

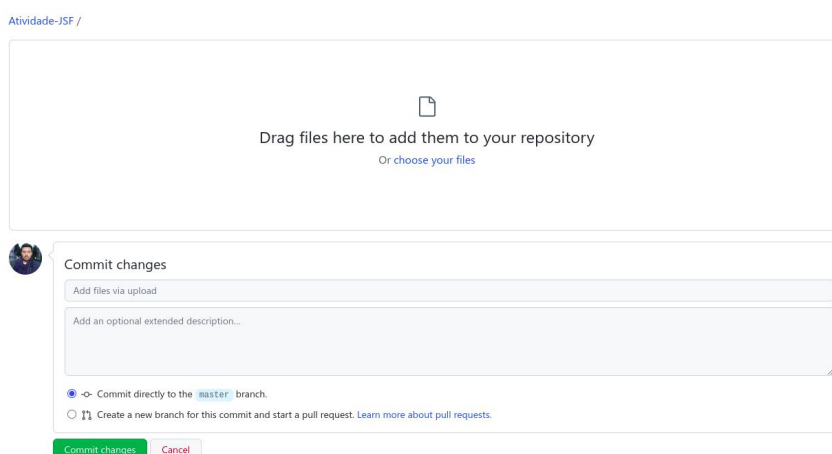
Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no

→ Continue

email, senha, nome de usuário, que não pode conter espaços e/ou caracteres especiais como “ç”, e na última pergunta, você pode teclar a letra “n” para não receber mensagens da plataforma. Basta seguir as etapas seguintes. Você deve inserir um e-mail válido pois, possivelmente, você receberá uma confirmação por e-mail, em forma de código ou link, para ativar sua conta.

3 - Agora, vamos abrir o projeto no Eclipse que pretendemos controlar com git. Em seguida, acesse sua conta no github e crie um novo repositório. Vá no botão New e crie um repositório com o nome Atividade-JSF. No campo Description coloque algo como “Projeto de aplicação Java com Eclipse.” Marque a opção “Add a README file” e clique sobre o botão “Create repository”.

4 - Faça o upload da pasta do seu projeto para o repositório criado. Sugere-se arrastar a pasta para cima do espaço dedicado ao upload na página do git. Veja:



Atividade-JSF /

Drag files here to add them to your repository
Or choose your files

Commit changes

Add files via upload

Add an optional extended description...

☒ Commit directly to the master branch.
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

5 - Depois de soltar a pasta do projeto sobre a área, aparecerá todos os arquivos que foram carregados e será possível realizar o primeiro commit, na área Commit changes. O commit, em linhas gerais, deve ser escrito para registrar quais alterações foram realizadas no repositório. Neste caso, acabamos de inserir arquivos de código do projeto ao repositório.

6 - Depois de escrever o texto do commit, basta clicar no botão Commit changes.

7 - Agora podemos nos desconectar da conta no github pois o processo de configuração, agora,

acontece no eclipse. Abra a ferramenta.

8 - Remova o projeto completamente de sua IDE, e, em seguida, faça a configuração dos repositório do git. Para isso, vá em Window > Show View > Other > Git > Git Repositories.

9 - Na janela que abrirá, vamos escolher a opção Clone a Git repository. Será necessário copiar a URL do repositório criado. Veja um exemplo: <<https://github.com/leonardossrocha/Atividade-JSF>>.

10 - Depois que o repositório for clonado, será necessário clicar com o botão direito do mouse sobre o nome do repositório e clicar na opção Import Projects.

11 - Para finalizar, vamos fazer uma pequena alteração no repositório, vamos inserir um comentário em um dos arquivos do projeto. Sugere-se a inclusão de um comentário no arquivo cadastro.xhtml, logo depois do elemento <h:body>, dessa maneira:

```
<h:body>
<!-- Formulário de cadastro -->
    <h1>Cadastro de usuário</h1>
    <h:form id="cadastro">
```

12 - Clique com o botão direito do mouse sobre o projeto é escolha a opção Team > Commit. Paracerá uma tela e nela, haverá uma área escrita Commit Message. Nela, você insere a mensagem de registro da alteração realizada.

13 - Por fim, é necessário escolher o botão Commit and Push, para commitar e já enviar as alterações para o repositório remoto. Caso você não tenha internet, é possível realizar só o commit localmente, escolhendo a opção Commit. Será solicitado as credenciais de acesso á conta, basta informar e clicar em Log In.

Checklist:

- 1 - Criar a conta na plataforma github.
- 2 - Iniciar processo de configuração do Git na IDE eclipse.
- 3 - Criar um repositório remoto e carregar os arquivos do projeto para ele.
- 4 - Clonar e importar o repositório na IDE Eclipse.
- 5 - Realizar alterações no código executar um commit com push para o repositório remoto.

Resultados da aula prática: Aluno, você deverá entregar:

Aplicação com repositório remoto devidamente criado. Código da aplicação commitada, clone para máquina local feito e primeiro commit e push direto da IDE Eclipse para a plataforma github.

Referências:

Git no Eclipse. Disponível em: <<https://git-scm.com/book/pt-pt/v2/Appendix-A%3A-Git-em-Outros-Ambientes-Git-no-Eclipse>>. Acesso em: 18, agosto de 2022.