

[Painel](#) / [Meus cursos](#) / [CETEJ5.2020 01.JAVA XIX](#) / [Semana 4 \(08 a 12/02\)](#) / [Questionário: componentes](#)

Iniciado em	domingo, 21 fev 2021, 16:42
Estado	Finalizada
Concluída em	domingo, 21 fev 2021, 17:05
Tempo empregado	23 minutos 13 segundos
Notas	4,80/6,00
Avaliar	8,00 de um máximo de 10,00(80%)

Questão 1

Parcialmente correto

Atingiu 0,80 de 1,00

Todos os componentes de interface em Vaadin herdam a classe **com.vaadin.flow.component.Component**. Por sua vez, os elementos HTML implementados em Vaadin possuem uma classe Java correspondente que herda a classe **com.vaadin.flow.component.HtmlComponent**.

Pesquise na documentação oficial (ou diretamente no código) e assinale quais dos métodos a seguir são válidos a todos os objetos HTML em Vaadin:

Escolha uma ou mais:

- ☐ a. htmlcomp.getTarget();
- ☒ b. htmlcomp.isVisible();
- ☒ c. htmlcomp.setAlt("Texto alternativo");
- ☒ d. htmlcomp.setWidth("200px");
- ☒ e. htmlcomp.getClassNames();
- ☒ f. htmlcomp.setValue(123);
- ☒ g. htmlcomp.getTitle();
- ☒ h. htmlcomp.setStyle("border: 1px solid red;");
- ☒ i. htmlcomp.getText();
- ☒ j. htmlcomp.setId("semid");

Sua resposta está parcialmente correta.

Você selecionou muitas opções.

As respostas corretas são: htmlcomp.setWidth("200px");, htmlcomp.setId("semid");, htmlcomp.getClassNames();, htmlcomp.isVisible();, htmlcomp.getTitle();

Questão **2**

Incorreto

Atingiu 0,00 de 1,00

Com base no código a seguir, assinale a alternativa correta:

```
Button btnEsconder = new Button("Esconder");  
Button btnMostrar = new Button("Mostrar", evt -> {  
    btnMostrar.setVisible(false);  
    btnEsconder.setVisible(!btnEsconder.isVisible());  
});
```

Escolha uma opção:

- ☐ a. Ao clicar no botão Mostrar apenas o botão ocultar se torna oculto
- ☐ b. Ao clicar 2 vezes no botão Mostrar o botão Esconder se tornará oculto e depois visível
- ☒ c. Ao clicar no botão Esconder o botão mostrar será exibido caso ele esteja oculto, caso contrário ele se tornará oculto
- ☐ d. Ao clicar no botão Mostrar ambos os botões se tornam ocultos
- ☐ e. Ao clicar no botão Esconder ambos os botões se tornam ocultos



Sua resposta está incorreta.

A resposta correta é: Ao clicar no botão Mostrar ambos os botões se tornam ocultos.

Questão **3**

Correto

Atingiu 1,00 de 1,00

Assinale os comandos de ligação com a correta definição da classe de visão:

```
@Route("teste/")
public class TesteView extends Div implements
HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, @OptionalParameter
String parameter) {
    }
}
```

new RouterLink("Página de teste", TesteView.class);



```
@Route("teste/")
public class TesteView extends Div implements
HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, @WildcardParameter
String parameter) {
    }
}
```

new RouterLink("Página de teste", TesteView.class, "teste/2");



```
@Route("main/teste")
public class MainView extends Div implements
HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, @WildcardParameter
String parameter) {
    }
}
```

new RouterLink("Início", MainView.class, "teste/abc");



```
@Route("main/")
public class MainView extends Div {
    ...
}
```

new RouterLink("Início", MainView.class);



```
@Route("main/")
public class MainView extends Div implements
HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, String parameter)
{
    }
}
```

new RouterLink("Início", MainView.class, "default");



Sua resposta está correta.

A resposta correta é:

```
@Route("teste/")
public class TesteView extends Div implements HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, @OptionalParameter String parameter) {
    }
}
```

→ new RouterLink("Página de teste", TesteView.class);,

```
@Route("teste/")
public class TesteView extends Div implements HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, @WildcardParameter String parameter) {
    }
}
```

→ new RouterLink("Página de teste", TesteView.class, "teste/2");

```
@Route("main/teste")
public class MainView extends Div implements HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, @WildcardParameter String parameter) {
    }
}
```

→ new RouterLink("Início", MainView.class, "teste/abc");

```
@Route("main/")
public class MainView extends Div {
    ...
}
```

→ new RouterLink("Início", MainView.class);

```
@Route("main/")
public class MainView extends Div implements HasUrlParameter<String> {
    ...
    public void setParameter(BeforeEvent event, String parameter) {
    }
}
```

→ new RouterLink("Início", MainView.class, "default");

Questão 4

Correto

Atingiu 1,00 de 1,00

Para se criar uma visão basta definir uma rota e herdar um componente do tipo *container*, ou seja, um componente que pode conter outros componentes. Assinale, dentre as alternativas a seguir, qual componente **NÃO** pode ser utilizado para este fim:

Escolha uma opção:

- ☐ a. Div
- ☐ b. VerticalLayout
- ☐ c. HorizontalLayout
- ☒ d. MixLayout
- ☐ e. AppLayout



Sua resposta está correta.

A resposta correta é: MixLayout.

Questão **5**

Correto

Atingiu 1,00 de 1,00

Para utilizar imagens como recursos em uma aplicação utilizando o Vaadin, as imagens devem ser armazenadas em um diretório específico. Qual é este caminho a partir do diretório principal do projeto?

Resposta: 

A resposta correta é: /src/main/webapp/.

Questão **6**

Correto

Atingiu 1,00 de 1,00

Existem duas formas de se criar um ícone: por meio do construtor ou por meio de uma *factory*. Considerando que se deseja criar um ícone chamado **YOUTUBE**, qual o código necessário para completar o exemplo a seguir:

Observação: Informe apenas o código para substituir o sinal de interrogação. Utilize a forma de sua preferência.

Resposta: 

A resposta correta é: new Icon(VaadinIcon.YOUTUBE);.

[◀ Exemplo dos componentes](#)[Vaadin Framework - Formulários ▶](#)