



fluiig
FLOWING
PRODUCTIVITY



WCM

TREINAMENTO DE DESENVOLVIMENTO

O **Web Content Management (WCM)** é uma ferramenta que promove gestão de conteúdo, colaboração e recursos de administração de portais na Web



1)

ESTRUTURA DE PÁGINAS

PÁGINA PRIVADA

Novo Registro

Ícone  [Trocar ícone](#)

Título da página* Analytics

Descrição * Widgets Analytics

Identificador único * analytics

URL personalizada para acesso [?](#)

Autorização* AUTHENTICATED

Página raiz [Localizar](#)

Opções de Menu

- Habilitar sub-páginas
- Exibir sub-páginas no sub-menu
- Página agrupadora

Outras opções

- Habilitar pathparam [?](#)

Na criação de uma página, selecionar a opção **Authenticated** no campo Autorização determina que a página estará visível apenas para os usuários autenticados

PÁGINA PRIVADA - URL

Elemento	Valor
Nome do servidor	meuserver.com.br
Porta utilizada pelo servidor fluig	8080
Código da empresa corrente	001

Considerando os dados acima, por exemplo, a URL da página seria
http://meuserver.com.br:8080/portal/p/001/analytics, onde:

Elemento	Descrição
/portal	Valor fixo da plataforma
/p	Indicativo de página privada
analytics	Campo “Identificador único” no cadastro da página

PÁGINA PÚBLICA

Novo Registro

Ícone  Trocar ícone	Título da página* Campanha do Agasalho	Autorização* PUBLIC
Descrição * Campanha destinada a arrecadação de roupas para crianças carentes.	Página raiz <input type="text"/> Localizar	Opções de Menu <input type="checkbox"/> Habilitar sub-páginas <input type="checkbox"/> Exibir sub-páginas no sub-menu <input type="checkbox"/> Página agrupadora
Identificador único * campanha_agasalho	Outras opções <input type="checkbox"/> Habilitar pathparam ?	URL personalizada para acesso <input type="text"/>

Na criação de uma página, selecionar a opção **Public** no campo Autorização determina que a página estará visível para usuários autenticados e visitantes externos

PÁGINA PÚBLICA - URL

Elemento	Valor
Nome do servidor	meuserver.com.br
Porta utilizada pelo servidor fluig	8080
Código da empresa corrente	001

Considerando os dados acima, por exemplo, a URL da página seria
http://meuserver.com.br:8080/portal/001/campanha_agasalho



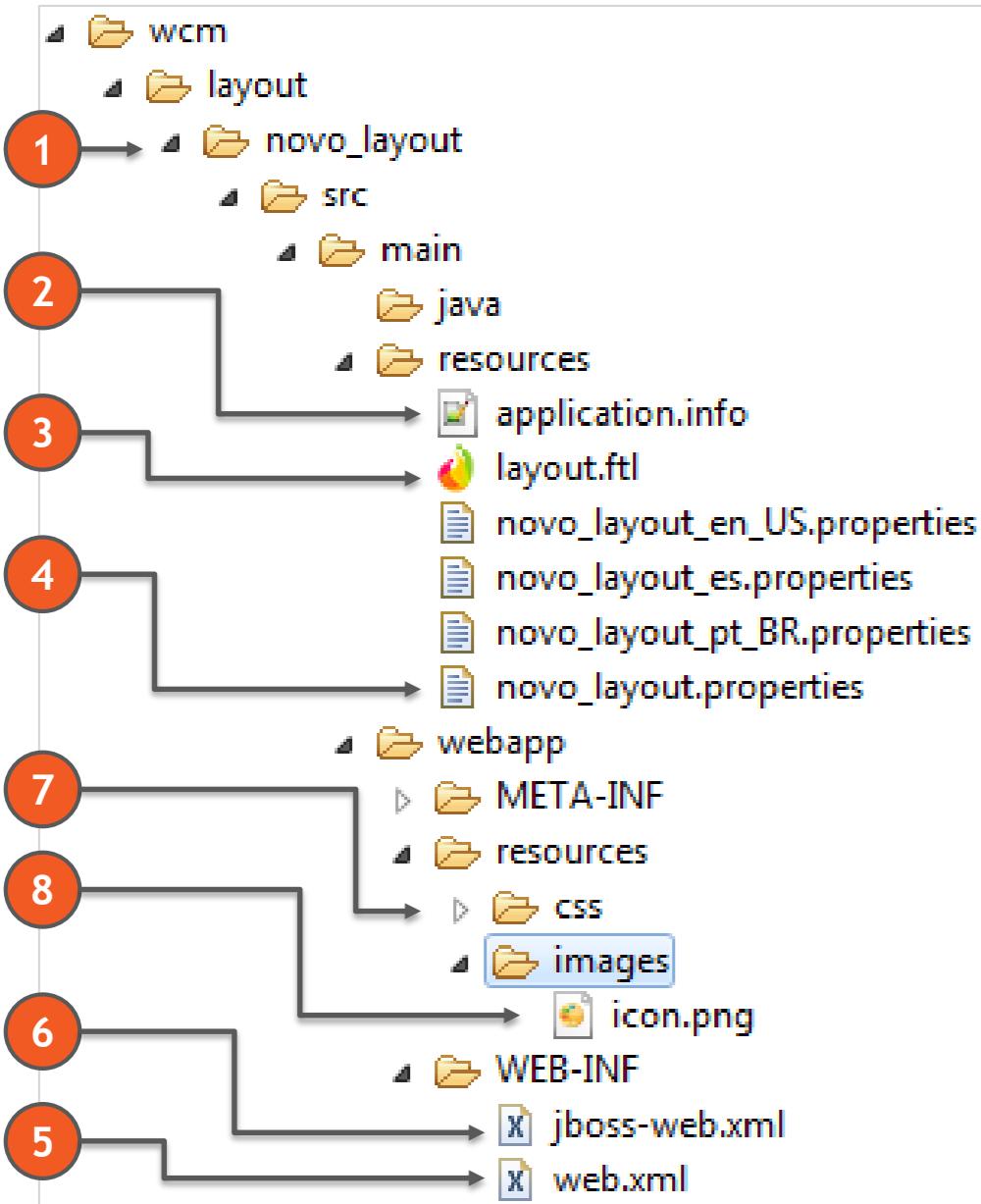
A ausência do elemento /p na URL indica que o acesso a essa página será público, sem necessidade de autenticação

FREEMARKER

- O desenvolvimento de layouts e widgets é baseado no framework de **templates FreeMarker (.ftl)**
- O FreeMarker é responsável pela camada de **visualização**: sobre ele é escrito todo o HTML do componente, enquanto a lógica é tratada separadamente
- Comandos FreeMarker no fluig Studio: 

- Toda página do fluig é constituída por um *layout* que define sua estrutura, ou seja, o posicionamento e tamanho dos *slots*
- Layouts personalizados podem ser criados utilizando o [fluig Studio](#)
- A estrutura de arquivos de um layout é semelhante à de um widget comum, inclusive o empacotamento em formato WAR

LAYOUTS



- | | |
|---|---|
| 1 | Nome/Identificador do layout |
| 2 | Arquivo de configuração do layout onde são armazenadas informações, como o código do layout, título e desenvolvedor |
| 3 | Contém a configuração dos slots |
| 4 | Arquivo de strings traduzíveis utilizadas pelo layout |
| 5 | Descriptor padrão de uma aplicação Java para web |
| 6 | Descriptor para Jboss. Deve conter a propriedade context-root, que representa o contexto web do layout. Recomendado que seja o próprio código do layout |
| 7 | Folhas de estilo do layout (caso seja necessário) |
| 8 | Ícone que representa o layout |

APPLICATION.INFO

- Para o sistema, o que diferencia um layout de um widget comum é a definição da propriedade “application.type = layout” no arquivo descriptor **application.info**
- É através dele que a aplicação tem conhecimento das propriedades do layout, identificação do desenvolvedor, arquivos necessários para sua renderização, arquivos de linguagem e outras propriedades

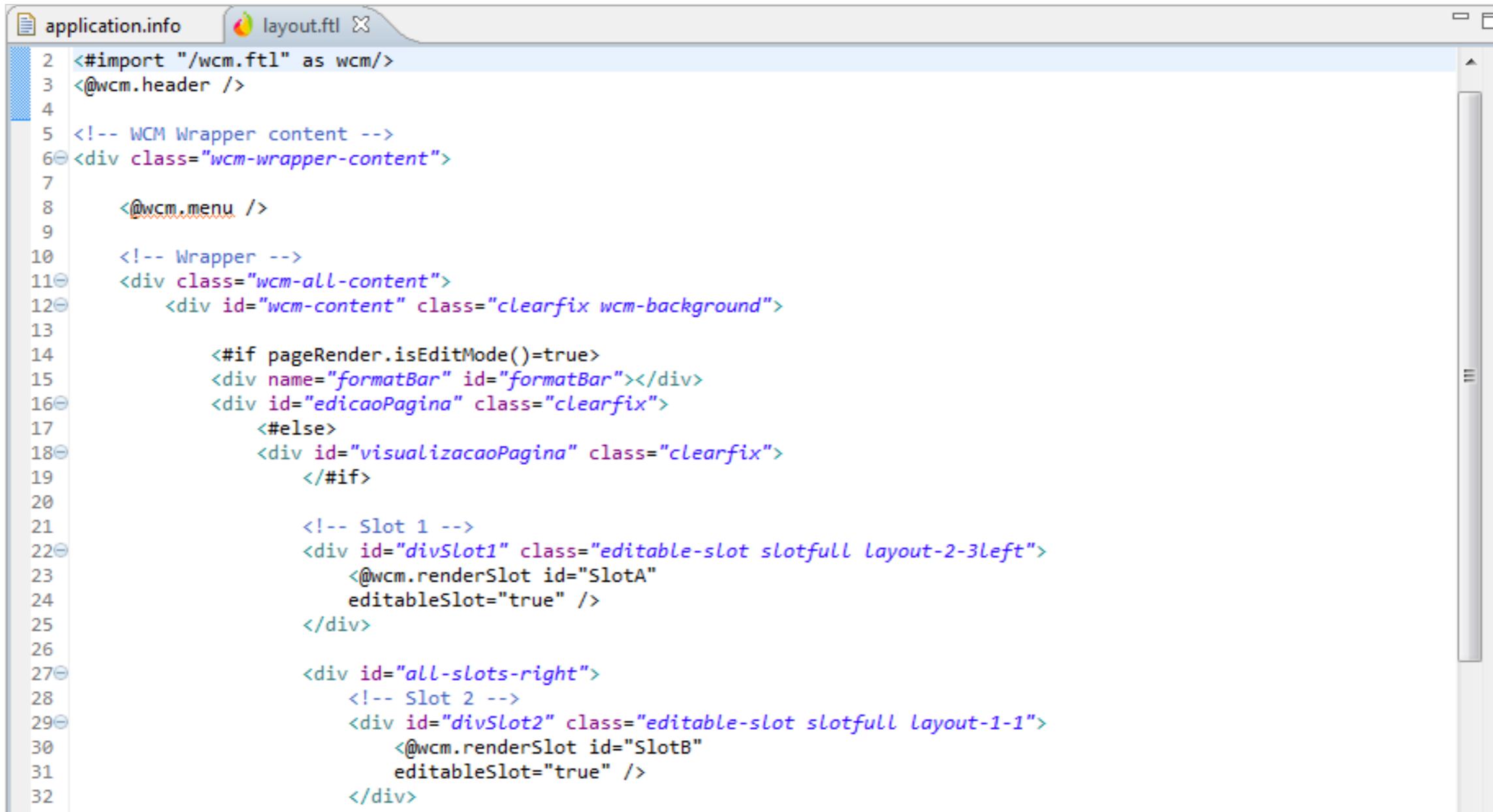


APPLICATION.INFO

```
application.info ×
1 application.type=layout
2 application.code=novo_layout
3 application.title=Novo Layout
4 application.description=Novo Layout
5 application.category=SYSTEM
6 application.renderer=freemarker
7 developer.code=JVD009371
8 developer.name=john.steel
9 developer.url=http://www.fluig.com
10 layout.defaultSlot=SlotA
11 layout.file=layout.ftl
12 locale.file.base.name=novo_layout
13 slot.SlotMenu=menu
14 slot.SlotLogin=sociallogin
15 application.category=APPLICATION
16 application.icon=icon.png
17 application.resource.css.1=/resources/css/wcm_widget.css
18 application.resource.js.2=/resources/js/wcm_widgets.js
19 |
```

- O principal arquivo de um projeto de layout é o “layout.ftl”
- É neste arquivo que será definido, via marcação HTML e Freemarker, a quantidade e disposição dos slots
- Para facilitar o desenvolvimento, foram disponibilizados alguns métodos e variáveis da camada de negócio:
 - Variáveis: sintaxe `${nome-da-variável}` em qualquer lugar do FTL
 - Métodos: objeto `pageRender`

LAYOUT.FTL



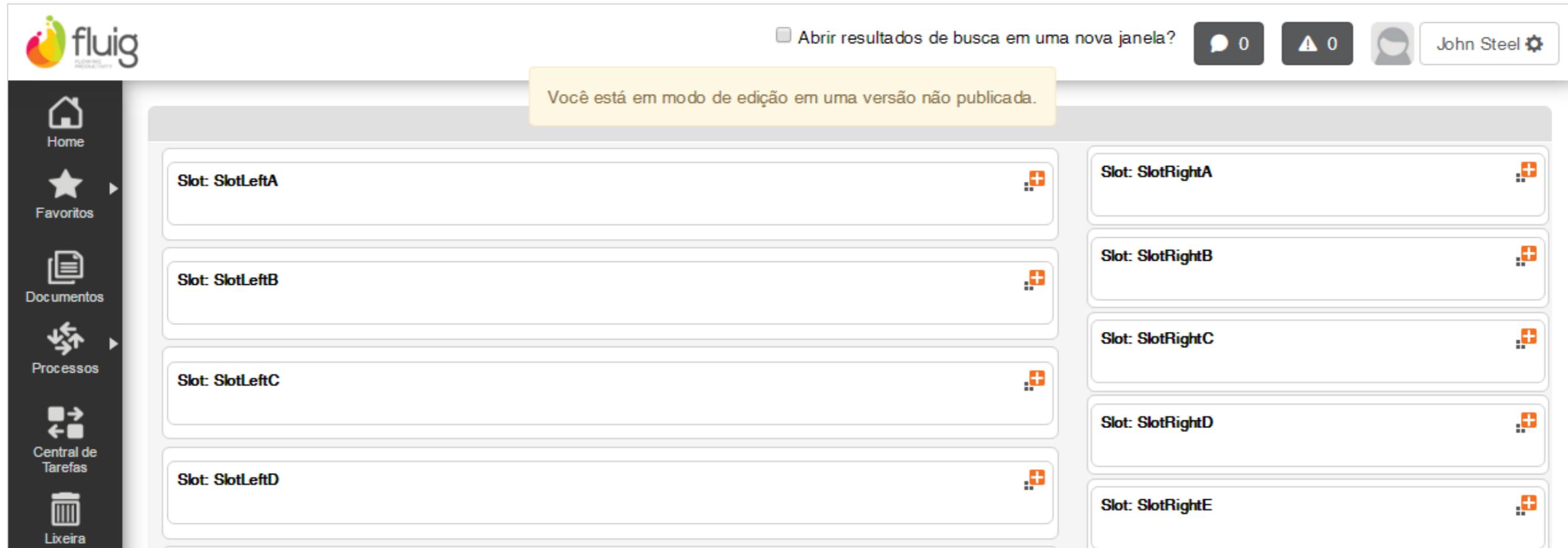
```
application.info layout.ftl
2 <#import "/wcm.ftl" as wcm/>
3 <@wcm.header />
4
5 <!-- WCM Wrapper content -->
6<div class="wcm-wrapper-content">
7
8     <@wcm.menu />
9
10    <!-- Wrapper -->
11<div class="wcm-all-content">
12    <div id="wcm-content" class="clearfix wcm-background">
13
14        <if pageRender.isEditMode()=true>
15            <div name="formatBar" id="formatBar"></div>
16            <div id="edicaoPagina" class="clearfix">
17                <#else>
18                <div id="visualizacaoPagina" class="clearfix">
19                    </if>
20
21        <!-- Slot 1 -->
22<div id="divSlot1" class="editable-slot slotfull layout-2-3left">
23    <@wcm.renderSlot id="SlotA"
24    editableSlot="true" />
25</div>
26
27<div id="all-slots-right">
28    <!-- Slot 2 -->
29<div id="divSlot2" class="editable-slot slotfull layout-1-1">
30    <@wcm.renderSlot id="SlotB"
31    editableSlot="true" />
32</div>
```

DESENVOLVIMENTO DE LAYOUTS

- A biblioteca `wcm.ftl` contém elementos básicos que podem ser inseridos na montagem de um layout, tais como menu, cabeçalho e rodapé da página: 
- Como retirar os elementos básicos de um layout personalizado? 

 Mesmo em páginas públicas, o usuário está sujeito à aplicação de permissões. Portanto, caso algum componente ou widget utilizado não seja público, o usuário pode não visualizar o elemento ou ser direcionado para a tela de login

SLOTS



The screenshot shows the fluig interface with a sidebar on the left containing navigation icons for Home, Favoritos, Documentos, Processos, Central de Tarefas, and Lixeira. The main area displays a grid of slots:

Slot: SlotLeftA	Slot: SlotRightA
Slot: SlotLeftB	Slot: SlotRightB
Slot: SlotLeftC	Slot: SlotRightC
Slot: SlotLeftD	Slot: SlotRightD
	Slot: SlotRightE

A message box in the center says "Você está em modo de edição em uma versão não publicada." (You are in edit mode in an unpublished version). At the top right, there are buttons for "Abrir resultados de busca em uma nova janela?" (Open search results in a new window), notifications (0), user profile (John Steel), and settings.

São áreas pré-definidas dentro do *layout* onde os *widgets* podem ser distribuídos na ordem desejada para atender a cada necessidade

```
1 <div class="editable-slot slotfull layout-1-1" id="slotContainer001">  
2   <@wcm.renderSlot id="Slot001" decorator="false" editableSlot="false" />  
3 </div>
```



1	editable-slot e slotfull: Usado pelo renderizador para montar o editor individual de slot layout-1-1: Usado para posicionar o slot em um determinado ponto da tela
2	slotContainer001: Identificação única do container. Não pode haver outro com o mesmo código
3	Slot001: Identificação única do slot. Não pode haver outro elemento com esse mesmo código
4	true/false: As widgets posicionadas neste slot exibem (true) ou não exibem (false) seus títulos
5	true: Torna possível inserir e retirar widgets dentro do slot bem como mudar o posicionamento delas false: Não permite qualquer edição no conteúdo do Slot. Slots de cabeçalho, rodapé e login geralmente possuem este valor

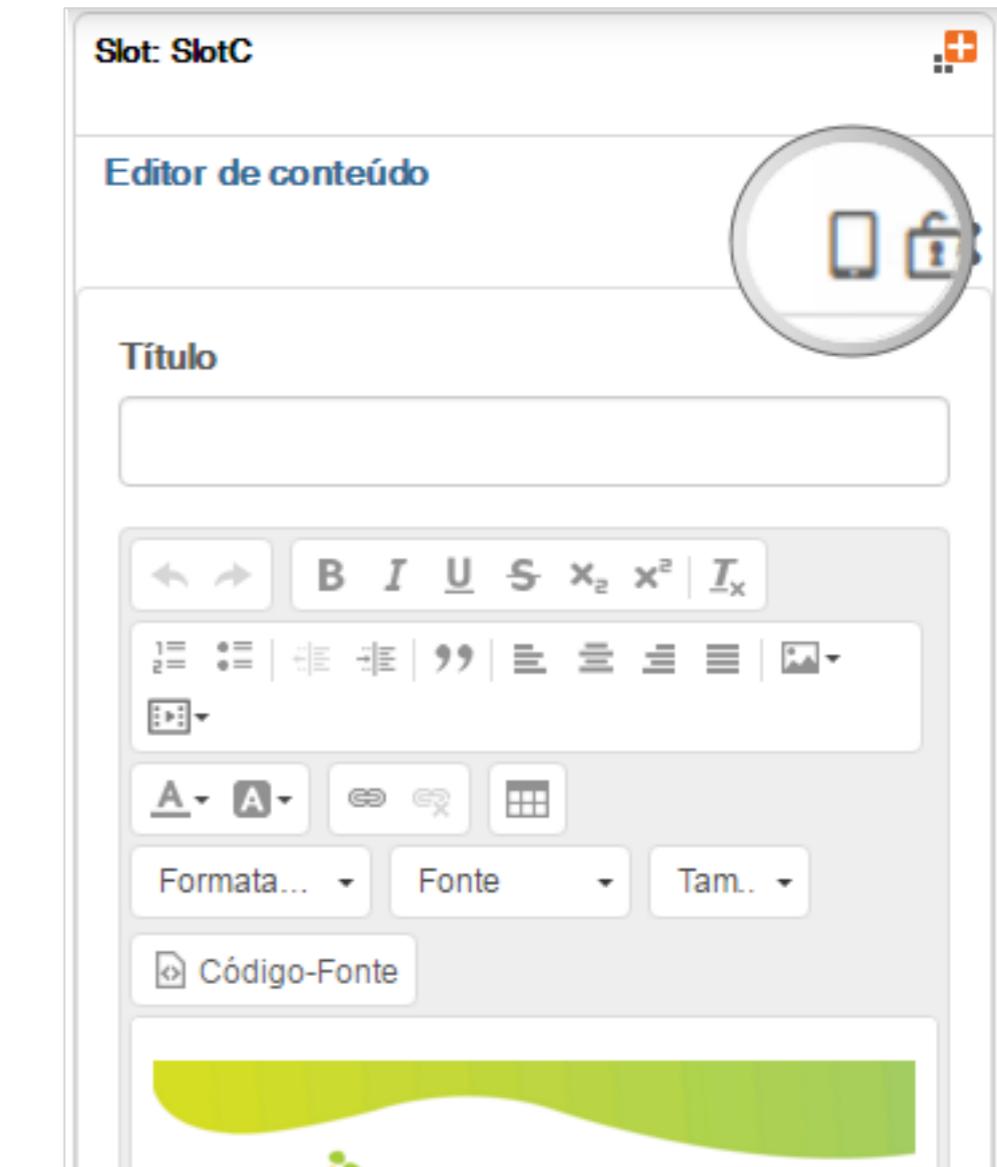
SLOTS

- Algumas classes são obrigatórias para o correto funcionamento do slot
- Para percorrer um determinado vetor, é utilizado o comando *list*
- No exemplo abaixo, será iterado pelo arranjo de widgets

```
<div class="editable-slot slotfull layout-1-1" id="slotFull1">
    <div id="SlotC" slot="true" decorator="false" editableSlot="true">
        <#list (pageRender.getInstancesIds("SlotC"))! as id>
            ${pageRender.renderInstanceNoDecorator(id)}
        </#list>
    </div>
</div>
```

WIDGETS

- Os **widgets** são componentes de tela que oferecem recursos específicos para o acompanhamento de Tarefas, Processos ou Documentos, entre muitos outros
- A organização de cada widget nas páginas é personalizável via edição de página



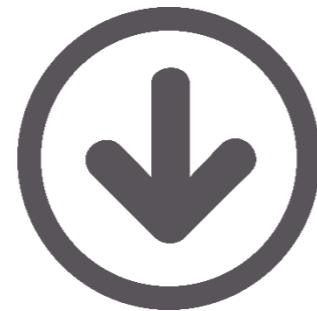


KIT INTRANET

O Kit Intranet oferece widgets,
técnicas e código de referência
para desenvolvedores:



Vídeo



Download

The screenshot shows the fluig intranet interface. At the top, there's a search bar with placeholder text "What would you like to find?", a user profile for "Adriano Amarante", and a notification badge showing "0". Below the header, there's a large banner image of a painting. To the right of the banner, the word "Convênios" is displayed above a list of partner logos and descriptions:

- Livraria**: 20% de desconto em periódicos. Estacionamento grátis. Parcelamento sem juros - qualquer valor.
- Academia**: Sem agendamento. 20% de desconto no personal trainer. Vaga para estacionamento.
- Padaria**: Agendamento para Café Colonial. Entrega gratuita - cesta de café da manhã. Café expresso gratuito (1...
- Farmácia**: 10% de desconto à vista. Medição de pressão gratuita (1 por dia). Medicamentos genéricos: compre 1, ...
- Restaurante**: Acompanhante não paga. 15% de desconto no estacionamento. Sobremesa grátis.

Below the conveniences section, there's a "Notícias" (News) section with a post titled "A aposta na educação corporativa" from "26/06/2015 - 18:00". The post features an illustration of a person with a graduation cap and a smartphone. The text discusses the company's investment in corporate education through a new LMS component.

At the bottom of the news section, there's a "Aniversariantes de julho" (July Birthdays) section featuring a photo of Edward Philips and the text "Hoje é aniversário de Edward Philips".

2)

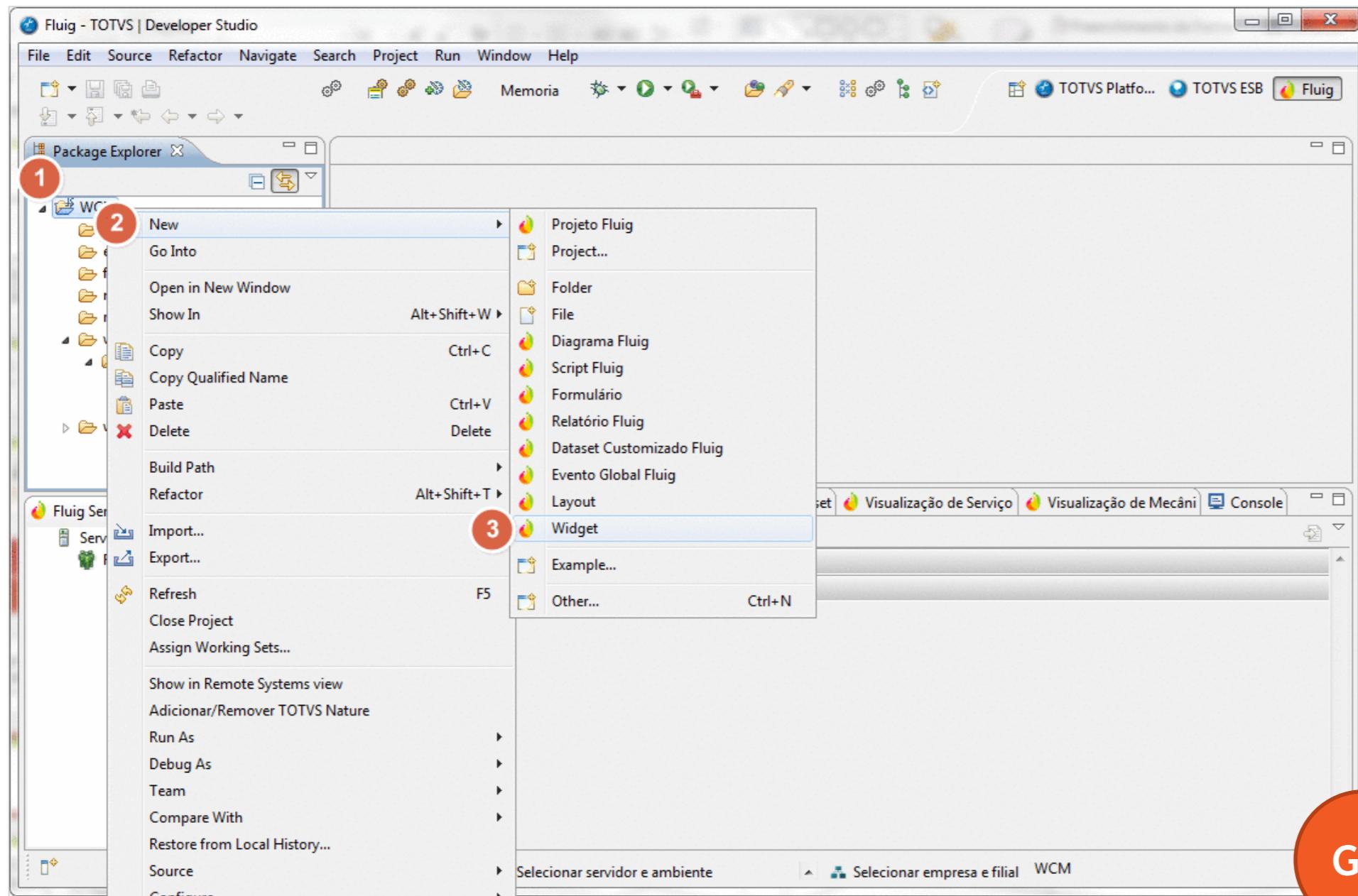
WIDGETS PERSONALIZADOS



WIDGETS PERSONALIZADOS

- Widgets personalizados são criados pelo próprio usuário, utilizando ou não templates de código disponíveis no fluig Studio
- Por intermédio deles, o usuário pode fornecer conteúdo personalizado, renderizar componentes de ERP ou mesmo de outros sistemas de terceiros
- **Como criar um widget que busque conteúdo:** 

NOVO WIDGET



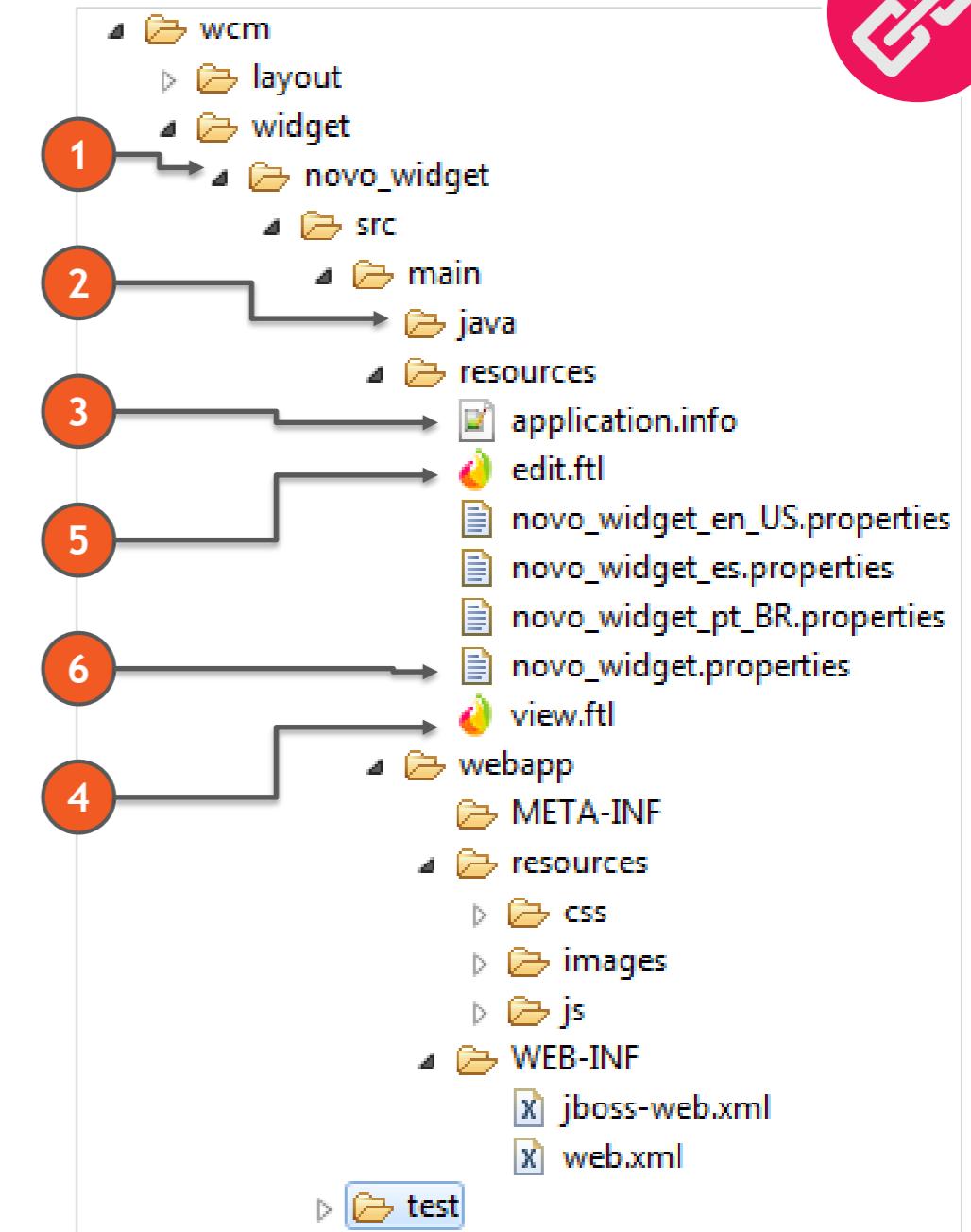
Snippets: 

GIF

WIDGETS PERSONALIZADOS

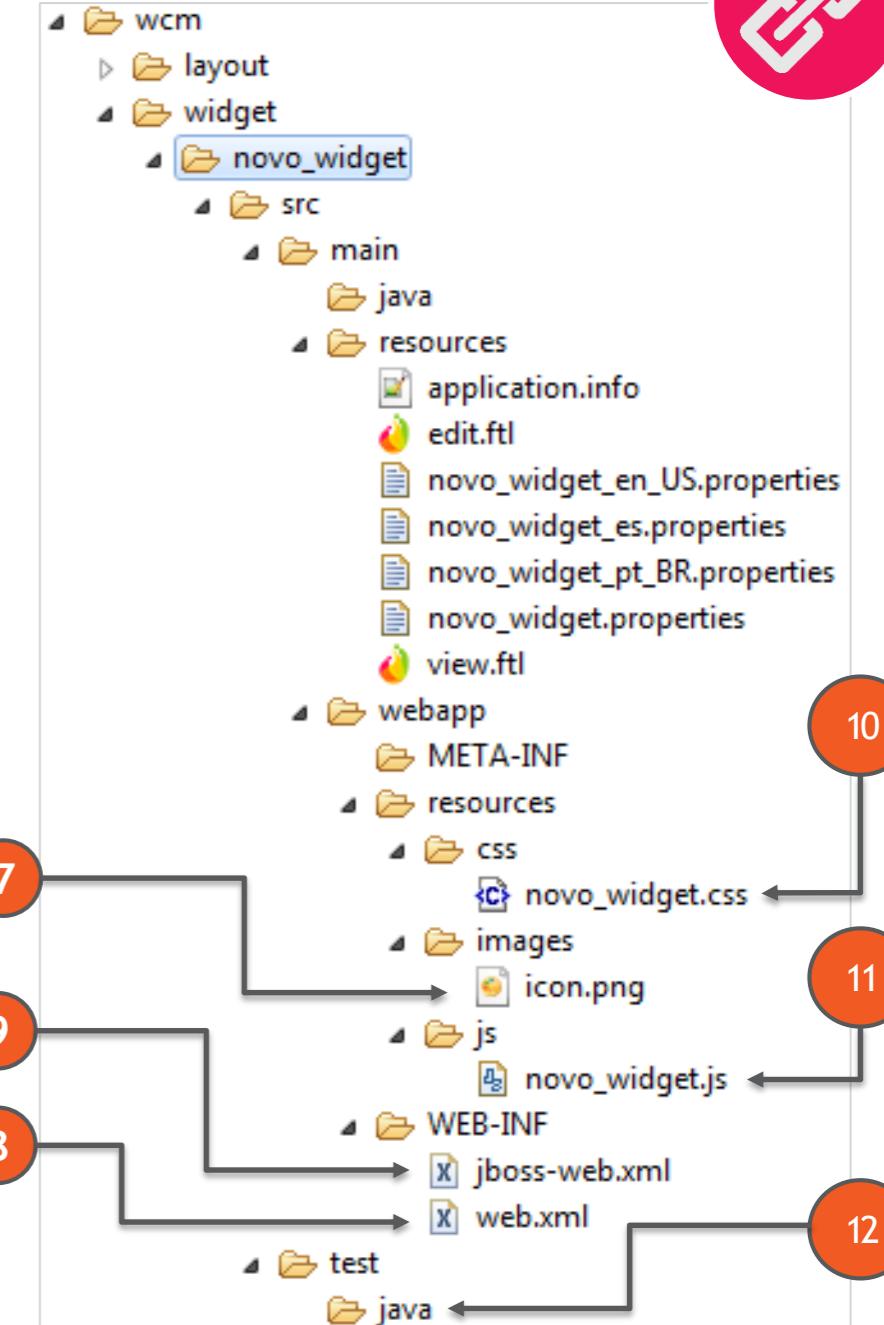


1	Nome/Identificador do Widget
2	Classes Java, caso exista alguma regra de negócio específica do Widget
3	Arquivo de configuração do Widget, onde é persistida, entre outras informações, o código da Widget, título e desenvolvedor
4	Arquivo de template do Freemarker, que será interpretado durante a renderização do Widget
5	Arquivo que será interpretado durante a renderização do Widget em modo de edição. Usado para configurar opções específicas para renderização do Widget, como por exemplo, filtros de data, cotação do dólar, etc.
6	Arquivo de strings traduzíveis utilizadas pelo Widget. Deve possuir derivações de acordo com o idioma suportado pelo Widget



WIDGETS PERSONALIZADOS

7	Ícone que representa a Widget. Será utilizado para representar a Widget no menu lateral da aplicação, ou em qualquer tela que necessite de representação visual do componente
8	Descriptor padrão de uma aplicação Java para web
9	Descriptor para Jboss. Deve conter a propriedade context-root, que representa o contexto web do layout. Recomendado que seja o próprio código do layout
10	Folha de estilo da Widget
11	Arquivo JavaScript da Widget (caso seja necessário)
12	Pasta específica para a construção de testes unitários





MÃOS À OBRA!

EXERCÍCIOS DE FIXAÇÃO - WCM

1. Importar o Kit Intranet para um projeto no Studio
2. Criar um widget que diga “Olá Mundo” sem a utilização de template
3. Criar um layout que utilize variáveis FTL e métodos do objeto pageRender
4. Neste layout, substitua o menu lateral por um menu superior

3)

SUPER WIDGET

- O Super Widget reúne processos comuns de desenvolvimento com o objetivo de padronizar e agilizar a criação de widgets
- Funciona como uma *super class* JavaScript

```
1  var MyWidget = SuperWidget.extend({  
2  
3      //method triggered when the widget loads  
4      init: function() {  
5          },  
6  
7      //widget variables  
8      instanceId: null,  
9      DOM: null,  
10  
11      //standard function of the widget  
12      someFunc: function() {  
13          }  
14  });
```



EXEMPLO DE IMPLEMENTAÇÃO

No fluig Studio, crie um novo widget utilizando o template **Documentos Favoritos** e acesse os arquivos abaixo para verificar a implementação:

- src/main/resources/view.ftl
- src/main/webapp/resources/js/favorites.js

Documentos favoritos				
	Descrição	Tamanho	Código	Versão
<input type="checkbox"/>	Imagens	19	30	1.000
<input type="checkbox"/>	Fluxo da URA.pdf	1,77 Mb	31	2.000
<input type="checkbox"/>	Configuração Java para WebDAV.docx	317,49 Kb	35	1.000
<input type="checkbox"/>	Visão Geral LMS - SCORM	7,25 Mb	65	1.000

Confira as estruturas CSS, componentes e melhores práticas disponíveis para agilizar o desenvolvimento de páginas e widgets: <http://style.fluig.com>

- Reúne elementos de interface para o desenvolvimento responsivo de páginas, formulários e widgets - inclusive Mobile
- Facilita a reutilização de código-fonte
- Garante a compatibilidade, consistência e uniformidade visual

4)

INTERNACIONALIZAÇÃO

INTERNACIONALIZAÇÃO

O conceito de internacionalização (também chamado de Localização ou i18n) consiste no desenvolvimento e/ou adaptação de um produto para outros idiomas

No fluig, o i18n é suportado através de arquivos de propriedades distintos para cada idioma (*.properties*)

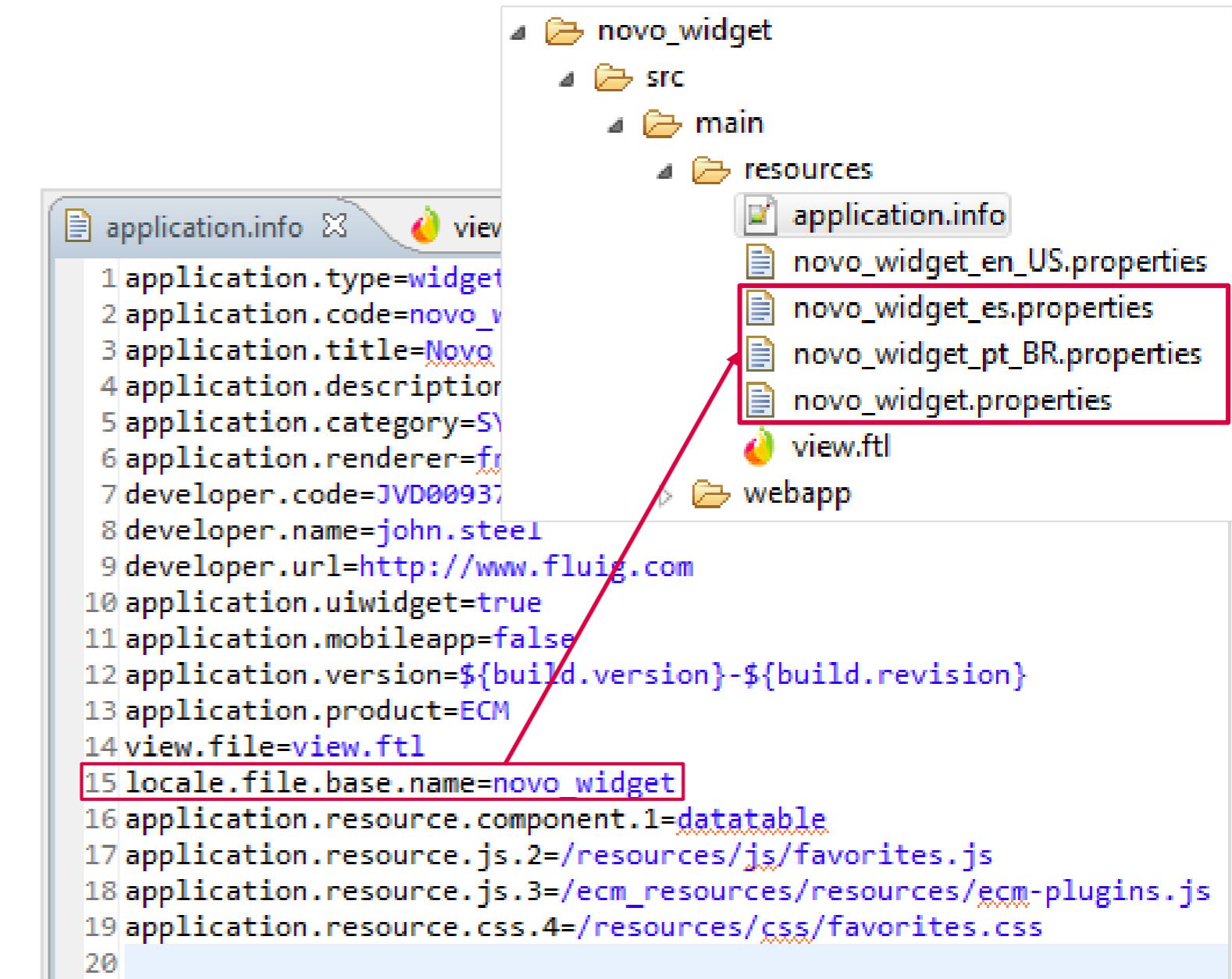


INTERNACIONALIZAÇÃO

Dentro do arquivo **application.info** deve existir uma propriedade chamada **locale.file.base.name**

É importante que o valor desta propriedade **seja o mesmo** que o prefixo dos arquivos de internacionalização

Utilização no desenvolvimento: 



5)

API PÚBLICA

- Application Programming Interface (Interface de Programação de Aplicativos)
- É um conjunto de rotinas e padrões estabelecidos pela plataforma para que aplicações externas possam utilizar seus recursos
- [Documentação](#) fluig API + [exemplo](#)
- api.fluig.com



CLIENTE REST



Instalação da extensão Advanced Client Rest para Google Chrome



CLIENTE REST

- A extensão Advanced Client REST é utilizada para realizar testes com os métodos disponíveis na API Pública
- Utilize a opção **GET** ou **POST**, conforme o tipo de chamada do método
 - GET: utilizado para solicitar um retorno do servidor
 - POST: utilizado para envio de dados ao servidor através de código JSON (no campos Raw) ou Form



CLIENTE REST

The screenshot shows a REST client application window. On the left, a sidebar lists navigation options: Request, Socket, Projects, Saved, History, Settings, About, RATE THIS APPLICATION ▾, and DONATE. The main area is titled "Documents" and contains a URL input field with the value "http://10.80.80.219:8081/api/public/ecm/document/34/1000", which is highlighted with a red arrow labeled "URL da API". Below the URL are several radio buttons for selecting HTTP methods: GET (selected), POST, PUT, PATCH, DELETE, HEAD, OPTIONS, and Other. Underneath these are three tabs: Raw, Form, and Headers, with Headers selected. The Headers section is currently empty. Below Headers are three more tabs: Raw, Form, and Files (0), with Payload selected. The Payload section is also empty. At the bottom right of the main area are "Clear" and "Send" buttons, with "Send" being highlighted by a yellow circle containing a hand cursor icon. Below the main area, the status bar shows "Status: 200: OK ? Loading time: 76ms". Further down, there are sections for "Response headers (7)", "Request headers (7)", and "Redirects (0)". The "Response headers (7)" section lists the following key-value pairs:

Header	Value
Server	Apache-Coyote/1.1
Cache-Control	private
Expires	Wed, 31 Dec 1969 21:00:00 BRT
Cache-Control	no-cache, no-transform, no-store
Content-Type	application/json;charset=UTF-8
Transfer-Encoding	chunked
Date	Thu, 03 Mar 2016 18:09:28 GMT

A large orange circle on the right side of the screenshot contains the text "GIF".

GET: /public/ecm/document/{documentId}/{versionId}



CLIENTE REST

The screenshot shows a REST client interface with the following details:

- Request:** [Unnamed]
- Socket:** http://10.80.80.219:8081/api/public/2.0/communities/create (highlighted by a red arrow labeled "URL da API")
- Method:** GET (radio button selected)
- Headers:** Content-Type: application/json (highlighted by a red arrow labeled "Content-Type")
- Payload:** (Raw, Form, Files (0), Payload)
 - ENCODE PAYLOAD** (highlighted by a red arrow labeled "Dados da comunidade")
 - DECODE PAYLOAD**

```
{  
    "alias" : "recursos-humanos",  
    "name" : "Recursos Humanos",  
    "description" : "Comunidade de RH",  
    "adminAlias" : "adm",  
    "hidden" : "false",  
    "privateContent" : "false",  
    "approvalRequired" : "false"  
}
```
- Buttons:** Clear, Send

POST: /public/2.0/communities/create

GIF

- Para obter dados de uma fonte externa à plataforma, a forma mais aconselhável é via chamada REST
- Configuração das chamadas REST no widget:
 - Criação da classe `ApplicationConfig`, que estende a classe `Application`
 - Criação da classe `REST`
 - Configuração dos arquivos `jboss-web.xml`, `web.xml` e `pom.xml`
- Integração de widget com serviços REST: 

INTEGRAÇÃO COM DATASETS DE FORMULÁRIOS

- Os dados preenchidos pelos usuários em campos de formulários podem ser armazenados em datasets e posteriormente consultados em widgets, por exemplo
- O fluig também dispõe de [datasets internos](#) que podem ser utilizados no desenvolvimento sobre a plataforma
- Integrando widget com formulários/datasets do fluig: 
- Exemplo de implementação Javascript no [widget Aniversariantes](#)

6

) COMPONENTES

WIDGET SLIDESHOW COM INSTAGRAM

- Widget SlideShow com integração ao Instagram: 
- Exemplo de implementação no Kit Intranet: 

Slot: SlotLeftA

Slideshow

Esta Widget Integra com

Instagram ▾

Estender Imagens

Conta do Instagram a ser consultada

instagram

Perfil no Instagram

Código de Cliente para Acesso

2b576a53492244cdgae50e01c6dc1f80

Gerar Novo Código de Acesso

DATATABLE

- O Datatable auxilia na criação de tabelas com paginação e recurso de busca: 
- Os dados da tabela podem ser obtidos por *array* de objetos ou via URL REST
- Exemplo de implementação no [widget de Notícias](#) do Kit Intranet

EXAMPLE

destroy create

Código	Nome	UF
1	Acre	AC
2	Alagoas	AL
3	Amazonas	AM
4	Amapá	AP
5	Bahia	BA
6	Ceará	CE
7	Distrito Federal	DF

<> >>

CONSUMO DE WS SOAP EM WIDGET

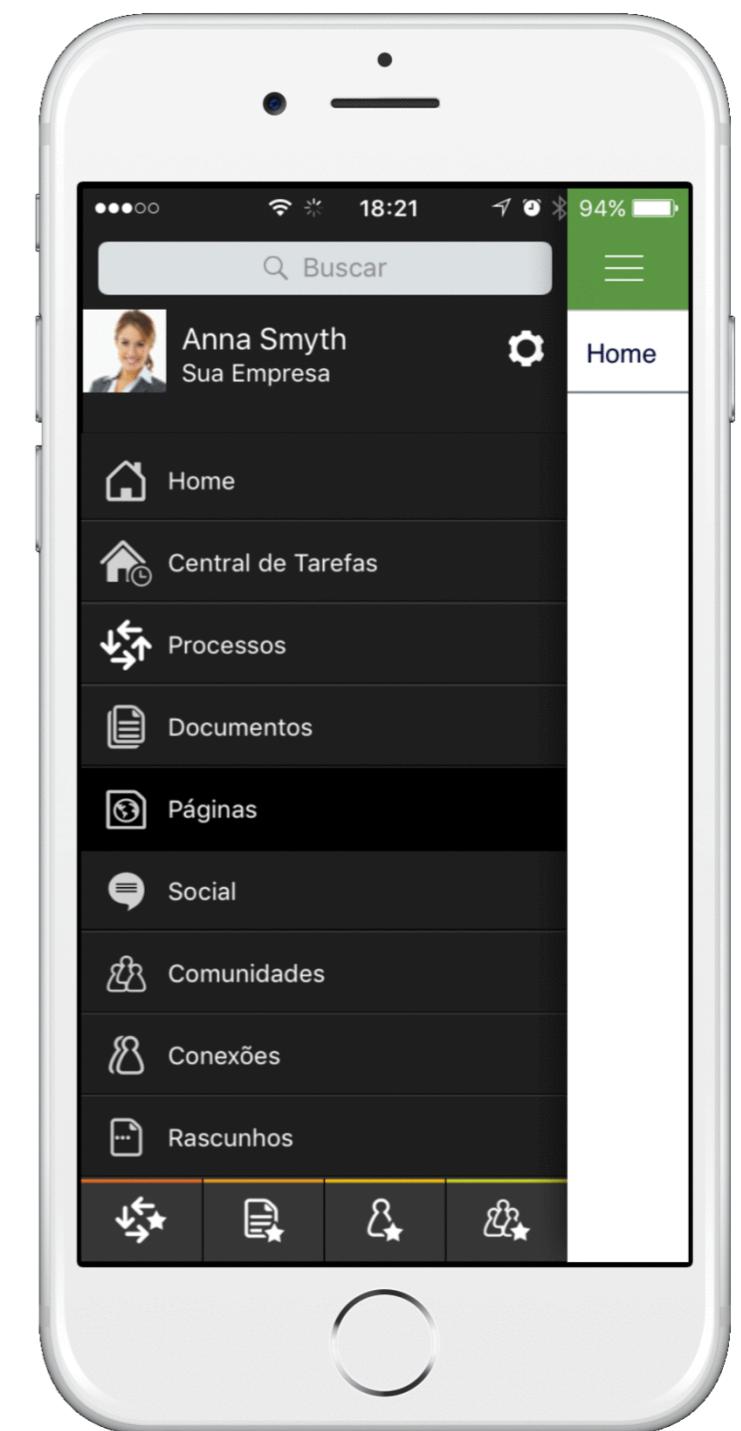
- Consumo de um WS SOAP de um Widget: 
- Exemplo de implementação SOAP no Kit Intranet (integração RM)
 - Widget Aniversariantes 
 - Widget Organograma 

7)

MOBILE

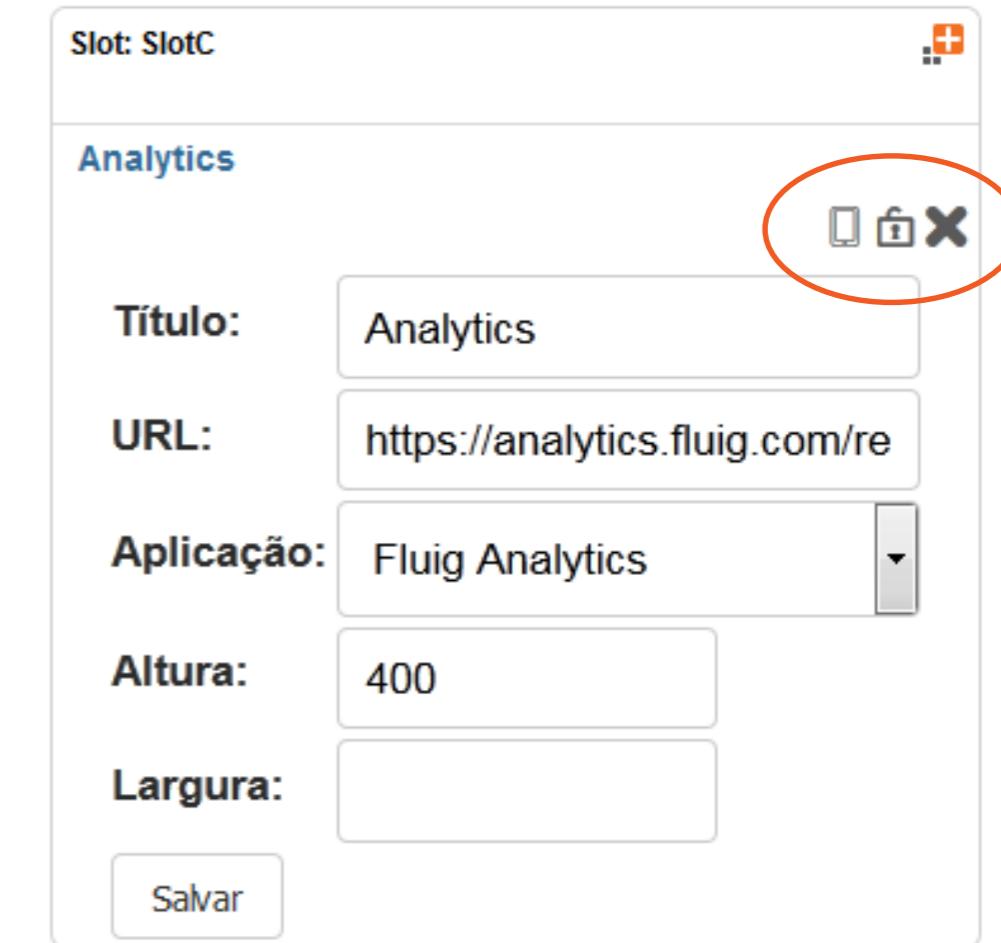
SUPORTE A PÁGINAS E WIDGETS MOBILE

- Deve ser adicionada a propriedade **application.mobileapp** no arquivo **application.info** com o valor *true*
- Boas Práticas para o desenvolvimento de widgets com exibição no fluig Mobile: 



SUPORTE A PÁGINAS E WIDGETS MOBILE

- A renderização das páginas no aplicativo mobile apresentará apenas o conteúdo das widgets habilitadas sequencialmente
- O widget precisa ser habilitado para exibição mobile na tela de edição da página
- Suporte à páginas e widgets Mobile: 



SUPORTE A PÁGINAS E WIDGETS MOBILE

A variável booleana `mobileAppMode` no template do FreeMarker determina se a widget está sendo renderizada através do aplicativo mobile:

```
if (WCMAPI.isMobileAppMode()) {  
    this.DOM.find('a').attr('href', '#');  
}
```



MÃOS À OBRA!



EXERCÍCIOS DE FIXAÇÃO

Chamadas ao serviços da API Pública por meio de um cliente REST

- O objetivo do exercício é fazer chamadas dos serviços da API Pública
- Fazer chamada ao serviço que retorne os documentos de uma determinada pasta. A pasta deve ser configurada na própria widget
- Utilize <http://<host>:<porta>/public/ecm/document/listDocument/{folderId}>



EXERCÍCIOS DE FIXAÇÃO

Construção de widget baseada na API

- O objetivo do exercício é desenvolver um widget baseada na API Pública
- Criar uma widget que exiba os documentos de uma pasta, a ser configurada através da interface da widget

Utilização de Dataset baseado no formulário para estruturação e exibição do widget

- O objetivo do exercício é utilizar um dataset para consultar as informações inseridas em um formulário
- Criar um formulário de Cadastro de Aniversariantes. Em seguida utilizar um dataset para consultar as informações contidas nesse formulário



EXERCÍCIOS DE FIXAÇÃO

Utilização de um componente Chart do fluig Style Guide

- O objetivo do exercício é desenvolver uma widget que inclua um gráfico de barras do Style Guide
- Crie uma enquete em uma comunidade e exiba os resultados em uma widget utilizando um gráfico de barras



EXERCÍCIOS DE FIXAÇÃO

Utilização de um componente Miscellaneous do fluig Style Guide

- O objetivo do exercício é desenvolver uma widget que inclua o componente Carousel do Style Guide
- Crie uma pasta no ECM do fluig e inclua imagens. Em seguida, exiba estas imagens em uma widget com o Carousel

CANAIS DE COMUNICAÇÃO FLUIG

- Site: fluig.com
- Documentação: dev.fluig.com
- Guia de Relacionamento: [!\[\]\(5ada69ed75f7e2e6a6490517d9128bf2_img.jpg\)](#)
- Suporte: suporte.fluig.com
- Comunidade DEV fluig: [!\[\]\(64f667f3b988e8687a512740d3752958_img.jpg\)](#)
- Blog: fluig.com/blog
- YouTube: youtube.com/fluigplatform
- SlideShare: pt.slideshare.net/fluig
- Scribd: scribd.com/fluigplatform



Acompanhe os canais sociais de fluig:

 /fluigplatform

 pt.scribd.com/fluigplatform

 /fluigplatform

 /company/fluig

 pt.slideshare.net/fluig

 /fluig.com/blog

www.fluig.com
0800 882 9191

OBRIGADO!

FLUIG EDUCATION CENTER

E-mail: fluig.education.center@fluig.com

Fone: (11) 2099-7337



fluig
FLOWING
PRODUCTIVITY