

```
<?xml version="1.0" encoding="utf-8"?>
<mx:TitleWindow xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="vertical" width="350" maxHeight="512"
    showCloseButton="true" creationComplete="init()" horizontalAlign="center"
    horizontalScrollPolicy="off" verticalScrollPolicy="auto">
<mx:states>
    <mx:State name="url">
        <mx:SetProperty target="{fileTextInput}" name="editable" value="true"/>
        <mx:RemoveChild target="{fileBtn}"/>
        <mx:SetProperty target="{fileTextInput}" name="width" value="245"/>
        <mx:RemoveChild target="{preview}"/>
        <mx:SetProperty target="{fileLabel}" name="text" value="URL"/>
        <mx:SetStyle target="{label1}" name="left" value="5"/>
        <mx:SetStyle target="{fileLabel}" name="left" value="5"/>
    </mx:State>
    <mx:State name="text">
        <mx:SetProperty target="{fileLabel}" name="text" value="Texto"/>
        <mx:RemoveChild target="{fileBtn}"/>
        <mx:RemoveChild target="{preview}"/>
        <mx:RemoveChild target="{fileTextInput}"/>
        <mx:AddChild relativeTo="{hbox1}" position="lastChild">
            <mx:TextArea id="textArea" width="245" height="200" x="67" y="0"/>
        </mx:AddChild>
        <mx:SetStyle target="{fileLabel}" name="verticalCenter"/>
        <mx:SetStyle target="{fileLabel}" name="left" value="5"/>
        <mx:SetStyle target="{fileLabel}" name="top" value="5"/>
        <mx:SetStyle target="{label1}" name="left" value="5"/>
    </mx:State>
    <mx:State name="question">
        <mx:SetProperty target="{fileLabel}" name="text" value="Pergunta"/>
        <mx:RemoveChild target="{fileBtn}"/>
        <mx:RemoveChild target="{preview}"/>
        <mx:RemoveChild target="{fileTextInput}"/>
        <mx:AddChild relativeTo="{hbox1}" position="lastChild">
            <mx:TextInput id="perguntaTextInput" width="245" x="67"/>
        </mx:AddChild>
        <mx:AddChild relativeTo="{hbox1}" position="lastChild">
            <mx:Label text="Obs" y="28" x="5"/>
        </mx:AddChild>
        <mx:AddChild relativeTo="{hbox1}" position="lastChild">
            <mx:TextArea id="comentTextArea" width="245" x="67" y="29"/>
        </mx:AddChild>
        <mx:AddChild relativeTo="{hbox1}" position="lastChild">
            <mx:Button label="Adicionar Resposta" click="addQuestionItem()"
horizontalCenter="0" y="77"/>
        </mx:AddChild>
        <mx:SetStyle target="{fileLabel}" name="verticalCenter"/>
        <mx:SetStyle target="{fileLabel}" name="verticalCenter"/>
        <mx:SetStyle target="{fileLabel}" name="left" value="5"/>
        <mx:SetStyle target="{fileLabel}" name="top" value="5"/>
        <mx:SetStyle target="{label1}" name="left" value="5"/>
    </mx:State>
    <mx:State name="video">
        <mx:RemoveChild target="{preview}"/>
```

```
<mx:SetProperty target="{fileLabel}" name="text" value="Vídeo"/>
</mx:State>
<mx:State name="file">
    <mx:RemoveChild target="{preview}"/>
    <mx:SetProperty target="{fileLabel}" name="text" value="Arquivo"/>
    <mx:SetProperty target="{fileTextInput}" name="width" value="161"/>
</mx:State>
</mx:states>

<mx:Script>
<![CDATA[
    import mx.core.Container;
    import mx.events.StateChangeEvent;
    import flash.text.engine.GroupElement;
    import br.com.optimedia.assets.ImgAssets;
    import mx.collections.ArrayCollection;
    import mx.containers.VBox;
    import mx.events.FlexEvent;
    import mx.controls.CheckBox;
    import mx.containers.HBox;
    import br.com.optimedia.assets.vo.QuestionItemVO;
    import br.com.optimedia.assets.vo.QuestionVO;
    import br.com.optimedia.assets.vo.FileVO;
    import mx.formatters.SwitchSymbolFormatter;
    import mx.events.ItemClickEvent;
    import mx.managers.PopUpManager;
    import mx.controls.Alert;
    import mx.rpc.events.FaultEvent;
    import br.com.optimedia.autor.AutorFacade;

    public static const UPLOAD_FILE_EVENT:String = "UPLOAD_FILE_EVENT";
    public static const UPLOAD_TEXT_EVENT:String = "UPLOAD_TEXT_EVENT";
    public static const CREATION_QUESTION_EVENT:String = "CREATION_QUESTION_EVENT";

    [Bindable]
    private var fr:FileReference;
    private var imageTypes:FileFilter = new FileFilter("Files (*.jpg, *.jpeg, *.png,
*.swf)", "*.jpg; *.jpeg; *.png, *.swf");
    private var allTypes:Array = new Array(imageTypes);
    public var fileVO:FileVO;
    public var questionVO:QuestionVO;

    public var category:int;
    //public var presentationID:uint;
    public var subjectID:uint;

    private var autorFacade:AutorFacade = AutorFacade.getInstance();

    private function init():void {
        autorFacade.Startup( this );

        fileTextInput.text = "";
        newFileReference();
    }
]]>
```

```
}

private function newFileReference():void {
    fr = new FileReference();
    fr.addEventListener(Event.COMPLETE, onComplete);
    fr.addEventListener(Event.SELECT, onSelect);
}

private function browse():void {
    fr.browse(allTypes);
}

private function onComplete(e:Event):void {
    saveBtn.enabled = true;
    if( this.currentState == '' || this.currentState == null ) preview.source =
fr.data;
}

private function onSelect(e:Event):void {
    fr.load();
    fileTextInput.text = fr.name;
}

private function save():void {
    if( this.currentState == null || this.currentState == 'video' ||
this.currentState == '' || this.currentState == 'file' ) {
        if( fileTextInput.text != '' && nameTextInput.text != '' ) {
            var data:ByteArray = new ByteArray();
            fileVO = new FileVO();

            //Read the bytes into bytarray var
            fr.data.readBytes(data, 0, fr.data.length);

            // Create a new FileVO instance
            fileVO.filename = fr.name;
            fileVO.filidata = data;

            dispatchEvent( new Event(UPLOAD_FILE_EVENT) );
        }
        else Alert.show("É necessário escolher um arquivo e digitar um nome.", "Atenção");
    }
    else if( this.currentState == 'url' ) {
        if( fileTextInput.text != '' && nameTextInput.text != '' ) {
            dispatchEvent( new Event(UPLOAD_TEXT_EVENT) );
        }
        else Alert.show("É necessário digitar uma URL.", "Atenção");
    }
    else if( this.currentState == 'text' ) {
        if( textArea.text != '' && nameTextInput.text != '' ) {
            dispatchEvent( new Event(UPLOAD_TEXT_EVENT) );
        }
        else Alert.show("Os campos de texto não podem estar vazios.", "Atenção");
    }
}
```

```
else if( this.currentState == 'question' ) {  
  
    if (perguntaTextInput.text!="" && nameTextInput.text!="") {  
        if (ArrayCollection(questionVO.itemArray).length>0) {  
            //var numberCorrect:Number = 0;  
            var responseWhite:Boolean = false;  
            for (var i:uint=0;i<ArrayCollection(questionVO.itemArray).length;  
i++) {  
                var questionItem:QuestionItemVO = new QuestionItemVO();  
                questionItem = questionVO.itemArray[i] as QuestionItemVO;  
                //if (questionItem.correct_answer==true){  
                //    numberCorrect++;  
                //}  
                if ( questionItem.name==null || questionItem.name.length==0){  
                    responseWhite=true;  
                }  
            }  
            //if (numberCorrect>=1 && responseWhite==false) {  
            if ( responseWhite==false) {  
                questionVO.title= nameTextInput.text  
                questionVO.comment = comentTextArea.text;  
                questionVO.description = perguntaTextInput.text  
  
                dispatchEvent( new Event(CREATION_QUESTION_EVENT) );  
            }else {  
                //if (numberCorrect==0) {  
                //    Alert.show("Pelo menos um dos item devem ser verdadeiro");  
                //}else {  
                    Alert.show("O texto das respostas deve ser preenchido");  
                }  
            }  
            }else {  
                Alert.show("Deve ter pelo menos uma alternativa");  
            }  
            }else {  
                Alert.show("A pergunta e nome são obrigatória");  
            }  
        }  
    }  
  
    private function generalFault(event:FaultEvent):void {  
        Alert.show(event.fault.faultCode as String);  
    }  
  
    public function removePopup():void {  
        PopUpManager.removePopUp( this );  
        System.gc();  
    }  
  
    private function handleRadio(event:ItemClickEvent):void {  
        newFileReference();  
        category = event.currentTarget.selectedValue;  
        saveBtn.enabled = false;  
        fileTextInput.setEditable = false;  
    }  
}
```

```
questionVO = new QuestionVO();
switch( event.currentTarget.selectedValue ) {
    case 4:
        this.currentState = '';
        fileTextInput.text = '';
        imageTypes = new FileFilter("Files (*.flv)", "*.flv");
        allTypes = new Array(imageTypes);
        break;
    case 5:
        this.currentState = 'url';
        fileTextInput.text = 'http://';
        fileTextInput.setEditable = true;
        saveBtn.enabled = true;
        break;
    case 6:
        this.currentState = 'text';
        textArea.text = '';
        saveBtn.enabled = true;
        break;
    case 7:
        this.currentState = 'file';
        fileTextInput.text = '';
        imageTypes = new FileFilter("Files (*.doc, *.docx, *.xls, *.xlsx,
*.ppt, *.pptx, *.pps, *.ppsx, *.pdf, *.zip, *.odt, *.ods, *.odp)",
"*.doc; *.docx; *.xls; *.xlsx;
*.ppt; *.pptx; *.pps; *.ppsx; *.pdf; *.zip; *.odt; *.ods; *.odp");
        allTypes = new Array(imageTypes);
        break;
    case 8:
        this.currentState = 'question';
        //textArea.text = '';
        saveBtn.enabled = true;
        break;
    default:
        this.currentState = '';
        fileTextInput.text = '';
        imageTypes = new FileFilter("Files (*.jpg, *.jpeg, *.png, *.swf)",
"*.jpg; *.jpeg; *.png; *.swf");
        allTypes = new Array(imageTypes);
        break;
}
}

private function addQuestionItem():void {
    var itemVO:QuestionItemVO = new QuestionItemVO();
    ArrayCollection(questionVO.itemArray).addItem(itemVO);

    // caixa da itens das questoes
    var vBox:VBox= new VBox();
    vBox.setStyle("borderStyle", "solid");
    vBox.percentWidth = 95;
    vBox.x=0;
    vBox.y=0;
    vBox.setStyle("paddingBottom", 5);
```

```
vBox.setStyle("paddingLeft",5);
vBox.setStyle("paddingRight",5);
vBox.setStyle("paddingTop",5);

// caixa com dos item true ou false

var texto:TextInput = new TextInput();
texto.percentWidth=98;
texto.addEventListener(Event.CHANGE, function (e:Event):void {
itemVO.name=texto.text; });

var nameGrupo:String = "nome_"+ArrayCollection(questionVO.itemArray).length;

var repostFalse:RadioButton = new RadioButton();
repostFalse.selected=true;
repostFalse.groupName = nameGrupo;
repostFalse.addEventListener(Event.CHANGE, function (e:Event):void {
    if (repostFalse.selected==true) {
        itemVO.correct_answer=false;
    }
});

var repostTrue:RadioButton = new RadioButton();
repostTrue.groupName = nameGrupo;
repostTrue.addEventListener(Event.CHANGE, function (e:Event):void {
    if (repostTrue.selected==true) {
        itemVO.correct_answer=true;
    }
});

repostFalse.label = "Resposta errada";
repostFalse.id = "RadioBtnQuestionItem";

repostTrue.label = "Resposta correta"
repostTrue.id = "RadioBtnQuestionItem";

//botao de apagar itemQuestion
var btn:Button = new Button();
btn.width = 24;
btn.height = 24;

//btn.data = media;
btn.setStyle("icon", ImgAssets.crosscircle);
btn.addEventListener(MouseEvent.CLICK,function ():void {
    removeChild(vBox);

ArrayCollection(questionVO.itemArray).removeItemAt(ArrayCollection(questionVO.itemArray).getIndexOf(itemVO));
    }
);
btn.toolTip = "apagar pergunta";

// adicionar objs a stage
```

```
        var hBox:HBox = new HBox();
        var hBoxTextoInput:HBox = new HBox();
        hBoxTextoInput.percentWidth=100;
        vBox.addChild(btn);
        vBox.addChild(hBoxTextoInput);
        hBoxTextoInput.addChild(texto);
        hBoxTextoInput.addChild(btn);
        hBox.addChild(respostFalse);
        hBox.addChild(respostTrue);
        vBox.addChild(hBox);
        vBox.name = "questionAnswersVBox";

        this.addChildAt(vBox,this.getChildren().length-1);

        this.addEventListener(StateChangeEvent.CURRENT_STATE_CHANGE,
removeQuestionAnswers);

        //função para remover lista de respostas se o currentState mudar de question
para outro
        function removeQuestionAnswers(e:StateChangeEvent):void {
            Container(e.currentTarget).removeChild(
Container(e.currentTarget).getChildByName("questionAnswersVBox") );
Container(e.currentTarget).removeEventListener(StateChangeEvent.CURRENT_STATE_CHANGE,
removeQuestionAnswers);
        }

        PopUpManager.centerPopUp(this);
    }

/* private function removeQuestionItem(event:Event):void {
    this.removeChild(Button(event.currentTarget));
}

*/
]]>
</mx:Script>

<mx:RadioButtonGroup id="radioBtnGroup" itemClick="handleRadio(event)" />

<mx:HBox width="100%" horizontalAlign="center" horizontalScrollPolicy="off"
verticalScrollPolicy="off">
    <mx:RadioButton id="imageRadio" label="Imagen" value="3" groupName="radioBtnGroup" />
    <mx:RadioButton id="tableRadio" label="Tabela" value="1" groupName="radioBtnGroup" />
    <mx:RadioButton id="chartRadio" label="Gráfico" value="2" groupName="radioBtnGroup" />
    <mx:RadioButton id="questionRadio" label="Questão" value="8" groupName="radioBtnGroup" />
</mx:HBox>
<mx:HBox width="100%" horizontalAlign="center" horizontalScrollPolicy="off"
verticalScrollPolicy="off">
    <mx:RadioButton id="linkRadio" label="URL" value="5" groupName="radioBtnGroup" />
    <mx:RadioButton id="videoRadio" label="Video" value="4" groupName="radioBtnGroup" />
    <mx:RadioButton id="textRadio" label="Texto" value="6" groupName="radioBtnGroup" />
    <mx:RadioButton id="fileRadio" label="Arquivo" value="7" groupName="radioBtnGroup" />
</mx:HBox>
```

```
<mx:Canvas width="100%" horizontalScrollPolicy="off" verticalScrollPolicy="off">
    <mx:Label text="Nome" verticalCenter="0" id="label1" left="5"/>
    <mx:TextInput id="nameTextInput" x="67" width="245" verticalCenter="0"/>
</mx:Canvas>

<mx:Canvas width="100%" id="hbox1" horizontalScrollPolicy="off" verticalScrollPolicy="off">
    <mx:Label id="fileLabel" text="Imagen" verticalCenter="0" left="5"/>
    <mx:TextInput id="fileTextInput" editable="false" x="67" verticalCenter="0"
width="161"/>
    <mx:Button id="fileBtn" label="Procurar" click="browse()" x="236"/>
</mx:Canvas>

<mx:Image id="preview" maxHeight="200" maxWidth="250"/>

<mx:HBox width="100%">
    <mx:Spacer width="100%"/>
    <mx:Button id="saveBtn" label="Salvar" click="save()" enabled="false"/>
</mx:HBox>

</mx:TitleWindow>
```