

```

<?xml version="1.0" encoding="utf-8"?>
<mx:VBox xmlns:mx="http://www.adobe.com/2006/mxml" height="100%" width="100%"
creationComplete="init()">

    <mx:Script>
        <![CDATA[
            import mx.rpc.events.ResultEvent;
            import mx.core.Application;

            import mx.events.CloseEvent;
            import mx.controls.Alert;
            import mx.managers.PopUpManager;
            import mx.controls.dataGridClasses.DataGridColumn;
            import mx.events.DataGridEvent;
            import mx.collections.ArrayCollection;
            import mx.controls.Button;
            import mx.containers.HBox;
            import br.com.optimedia.assets.vo.SlideVO;
            import br.com.optimedia.assets.vo.PresentationVO;
            import br.com.optimedia.assets.vo.SubjectVO;
            import br.com.optimedia.autor.AutorFacade;

            public static const SAVE_SUBJECT_EVENT:String = "SAVE_SUBJECT";
            public static const SAVE_PRESENTATION_EVENT:String = "SAVE_PRESENTATION";
            public static const DELETE_SUBJECT_EVENT:String = "DELETE_SUBJECT_EVENT";
            public static const DELETE_PRESENTATION_EVENT:String = "DELETE_PRESENTATION_EVENT";
            public static const LOCK_UNLOCK_PRESENTATION_EVENT:String =
"LOCK_UNLOCK_PRESENTATION_EVENT";

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

            [Bindable]
            public var subjectArray:ArrayCollection = new ArrayCollection();

            [Bindable]
            public var presentationArray:ArrayCollection = new ArrayCollection();

            [Bindable]
            public var presentationSkins:ArrayCollection;

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

            private function onSubjectSelectionChange():void {
                if( subjectGrid.selectedIndex != -1 ) {
                    presentationArray = SubjectVO(subjectGrid.selectedItem).presentationArray;
                }
                if( presentationGrid.selectedIndex == -1 ) {
                    slideEditBtn.enabled = false;
                }
            }

            private function slidesNumLabelFunction(item:Object, column:DataGridColumn):String {

```

```

        var counter:uint = 0;
        for each( var slide:SlideVO in item.slidesArray ) counter++;
        return counter.toString();
    }

    public var newSubjectPopUp:NewSubjectPopUp;
    private function subjectEdit(subject:SubjectVO):void {
        newSubjectPopUp = new NewSubjectPopUp();
        PopUpManager.addPopUp( newSubjectPopUp, this, true );
        PopUpManager.centerPopUp( newSubjectPopUp );
        newSubjectPopUp.subjectVO = subject.clone();
        newSubjectPopUp.saveBtn.addEventListener(MouseEvent.CLICK, saveSubject);
    }

    private function saveSubject(event:MouseEvent):void {
        this.dispatchEvent( new Event(SAVE_SUBJECT_EVENT) );
    }

```

```

    public var newPresentationPopUp:NewPresentationPopUp;
    private function presentationEdit(presentation:PresentationVO):void {
        newPresentationPopUp = new NewPresentationPopUp();
        PopUpManager.addPopUp( newPresentationPopUp, this, true );
        PopUpManager.centerPopUp( newPresentationPopUp );
        newPresentationPopUp.presentationSkins = presentationSkins;
        newPresentationPopUp.presentationVO = presentation.clone();
        newPresentationPopUp.selectedSkin();
        newPresentationPopUp.presentationVO.subject_id =
SubjectVO(subjectGrid.selectedItem).subject_id;
        newPresentationPopUp.saveBtn.addEventListener(MouseEvent.CLICK,
savePresentation);
    }

```

```

    private function savePresentation(event:MouseEvent):void {
        this.dispatchEvent( new Event(SAVE_PRESENTATION_EVENT) );
    }

```

```

    public function editItem(data:*) :void {
        if( data is SubjectVO ) {
            subjectEdit( data );
        }
        else if( data is PresentationVO ) {
            presentationEdit( data );
        }
    }

```

```

    public function deleteItem(data:Object):void {
        Alert.noLabel = "Não";
        Alert.yesLabel = "Sim";
        if( data is SubjectVO ) {

```

```

            Alert.show("Tem certeza que deseja apagar o módulo
"+subjectGrid.selectedItem.title+"?", "Atenção",Alert.YES|Alert.NO, null, deleteSubject, null,
Alert.NO);

```

```

            function deleteSubject(event:CloseEvent):void {
                if(event.detail == Alert.YES) {

```

```

        dispatchDeleteSubjectEvent();
    }
}
}
else if( data is PresentationVO ) {
    Alert.show("Tem certeza que deseja apagar o tema
"+presentationGrid.selectedItem.title+"?", "Atenção",Alert.YES|Alert.NO, null,
deletePresentation, null, Alert.NO);
    function deletePresentation(event:CloseEvent):void {
        if(event.detail == Alert.YES) {
            dispatchDeletePresentationEvent();
        }
    }
}
}
private function dispatchDeleteSubjectEvent():void {
    this.dispatchEvent( new Event(DELETE_SUBJECT_EVENT) );
}
private function dispatchDeletePresentationEvent():void {
    this.dispatchEvent( new Event(DELETE_PRESENTATION_EVENT) );
}

public function openSendFilePopUp(presentationID:uint, fileType:String):void {
    var popUp:SendFilePopUp = new SendFilePopUp();
    popUp.presentationID = presentationID;
    popUp.fileType = fileType;
    PopUpManager.addPopUp( popUp, this, true );
    PopUpManager.centerPopUp( popUp );
}

public function showPublishPopUp(data:Object):void {
    var popUp:PublishPresentationPopUp = new PublishPresentationPopUp();
    popUp.presentationVO = data as PresentationVO;
    PopUpManager.addPopUp( popUp, Application.application as DisplayObject, true );
    PopUpManager.centerPopUp( popUp );
}
[Bindable]
public var userRole:String;
[Bindable]
public var userID:int;
public function setRoleID(userRole:String):void {
    this.userRole = userRole;
    if( userRole == AutorFacade.IS_EDITOR ) {
        newSubjectBtn.visible = false;
        newPresentationBtn.visible = false;
    }
    else if( userRole == AutorFacade.IS_ADMIN ) {
        newSubjectBtn.visible = true;
        newPresentationBtn.visible = true;
    }
    else if( userRole == AutorFacade.IS_AUTHOR ) {
        newSubjectBtn.visible = true;
        newPresentationBtn.visible = true;
    }
}

```

```

        else if( userRole == AutorFacade.IS_OBSERVER ) {
            newSubjectBtn.visible = false;
            newPresentationBtn.visible = false;
            slideEditBtn.label = "Ver slides";
        }
    }

    private function slideEditionEnabled():void {
        if( presentationGrid.selectedIndex!=-1 ) {
            if( userRole == AutorFacade.IS_OBSERVER ) {
                slideEditBtn.enabled = true;
                slideEditBtn.toolTip = "Clique para ver os slides da apresentação
selecionada";
            }
            else if( PresentationVO(presentationGrid.selectedItem).section_id == 0 ) {
                slideEditBtn.enabled = true;
                slideEditBtn.toolTip = "Clique para editar os slides da apresentação
selecionada";
            }
            else {
                slideEditBtn.enabled = false;
                slideEditBtn.toolTip = "Não é possível editar os slides de
apresentações publicadas";
            }
        }
    }

    public function toggleLock(item:PresentationVO):void {
        dispatchEvent( new ResultEvent( LOCK_UNLOCK_PRESENTATION_EVENT, false, true,
item ) );
    }
}]]>
</mx:Script>

<mx:Button id="newSubjectBtn" label="Novo Módulo" click="subjectEdit( new SubjectVO()
)"/>

<mx:DataGrid id="subjectGrid" dataProvider="{subjectArray}"
change="onSubjectSelectionChange()" width="100%" height="100%">
    <mx:columns>
        <mx:DataGridColumn headerText="Módulo" dataField="title"/>
        <mx:DataGridColumn headerText="Descrição" dataField="description"/>
        <!--<mx:DataGridColumn headerText="Status" dataField="status" width="50"
textAlign="center" />-->
        <mx:DataGridColumn headerText="Opções" width="100" textAlign="center"
visible="{userRole==AutorFacade.IS_ADMIN || userRole==AutorFacade.IS_AUTHOR}">
            <mx:itemRenderer>
                <mx:Component>
                    <mx:HBox horizontalScrollPolicy="off" verticalScrollPolicy="off"
horizontalAlign="center">
                        <mx:Script>
                            <![CDATA[
                                import br.com.optimedia.autor.AutorFacade;
                                import br.com.optimedia.assets.ImgAssets;

```

```

        ]]>
    </mx:Script>
    <mx:LinkButton icon="{ImgAssets.editIcon}"
click="parentDocument.editItem(data)"
        tooltip="Editar Módulo"
visible="{parentDocument.userRole==AutorFacade.IS_ADMIN ||
parentDocument.userRole==AutorFacade.IS_AUTHOR}"/>
    <mx:LinkButton icon="{ImgAssets.deleteIcon}"
click="parentDocument.deleteItem(data)"
        tooltip="Remover Módulo"
visible="{parentDocument.userRole==AutorFacade.IS_ADMIN ||
parentDocument.userRole==AutorFacade.IS_AUTHOR}"/>
    </mx:HBox>
</mx:Component>
</mx:itemRenderer>
</mx:DataGridColumn>
</mx:columns>
</mx:DataGrid>

    <mx:Button id="newPresentationBtn" label="Novo Tema" click="presentationEdit( new
PresentationVO() )" enabled="{subjectGrid.selectedIndex!=-1}"/>

    <mx:DataGrid id="presentationGrid"
dataProvider="{SubjectVO(subjectGrid.selectedItem).presentationArray}" width="100%" height="100%"
    itemClick="slideEditionEnabled()">
    <mx:columns>
        <mx:DataGridColumn headerText="Tema" dataField="title"/>
        <mx:DataGridColumn headerText="Descrição" dataField="description"/>
        <mx:DataGridColumn headerText="Créditos" width="100">
            <mx:itemRenderer>
                <mx:Component>
                    <mx:Canvas horizontalScrollPolicy="off" verticalScrollPolicy="off">
                        <mx:Script>
                            <![CDATA[
                                import br.com.optimedia.assets.ImgAssets;
                            ]]>
                        </mx:Script>
                        <mx:Label text="{data.img_credit}" verticalCenter="0"/>
                        <mx:LinkButton icon="{ImgAssets.editIcon}" right="0"
verticalCenter="0"
click="parentDocument.openSendFilePopUp(data.presentation_id, 'img_credit')"
                            tooltip="Editar imagem de Crédito"/>
                    </mx:Canvas>
                </mx:Component>
            </mx:itemRenderer>
        </mx:DataGridColumn>
        <mx:DataGridColumn headerText="Introdução" width="100">
            <mx:itemRenderer>
                <mx:Component>
                    <mx:Canvas horizontalScrollPolicy="off" verticalScrollPolicy="off">
                        <mx:Script>
                            <![CDATA[
                                import br.com.optimedia.assets.ImgAssets;

```



```

        private function iconFunction(item:Object):Class {
            var resultIcon:Class = ImgAssets.unpublishedIcon;
            if( item.section_id != 0 ) {
                resultIcon = ImgAssets.publishedIcon;
            }
            return resultIcon;
        }

        private function lockIconFunction(item:Object):Class {
            var resultIcon:Class = ImgAssets.unlockedIcon;
            if( item.locked_by != 0 ) {
                resultIcon = ImgAssets.lockedIcon;
            }
            return resultIcon;
        }

        private function lockToolTipFunction(item:Object):String {
            var resultString:String = "Bloquear";
            if( item.locked_by != 0 ) {
                resultString = "Desbloquear";
            }
            return resultString;
        }

        private function showLockIcon(item:Object):Boolean {
            var result:Boolean = false;
            if( parentDocument.userRole == AutorFacade.IS_ADMIN )

result = true;

            else if( item.locked_by != 0 && parentDocument.userID

== item.locked_by ) result = true;

            return result;
        }
    ]]>
</mx:Script>
<mx:LinkButton icon="{ImgAssets.editIcon}"
click="parentDocument.editItem(data)"
tooltip="Editar Tema"
visible="{parentDocument.userRole!=AutorFacade.IS_EDITOR}"/>
<mx:LinkButton icon="{ImgAssets.deleteIcon}"
click="parentDocument.deleteItem(data)"
tooltip="Remover Tema"
visible="{parentDocument.userRole!=AutorFacade.IS_EDITOR}"/>
<mx:LinkButton icon="{iconFunction(data)}"
click="parentDocument.showPublishPopUp(data)"
tooltip="{tooltipFunction(data)}"
visible="{parentDocument.userRole!=AutorFacade.IS_EDITOR}"/>
<mx:LinkButton icon="{lockIconFunction(data)}"
click="parentDocument.toggleLock(data)"
tooltip="{lockToolTipFunction(data)}"
visible="{showLockIcon(data)}"
includeInLayout="{showLockIcon(data)}"/>
</mx:HBox>
</mx:Component>
</mx:itemRenderer>

```

```
        </mx:DataGridColumn>
    </mx:columns>
</mx:DataGrid>

<mx:HBox width="100%">

    <mx:Spacer width="100%" />
    <mx:Button label="Editar Slides" id="slideEditBtn" enabled="false" />

</mx:HBox>
</mx:VBox>
```