

```
package br.com.optimedia.autor.view
{
    import br.com.optimedia.assets.CommandConstants;
    import br.com.optimedia.assets.NotificationConstants;
    import br.com.optimedia.autor.model.RepositoryManagerProxy;
    import br.com.optimedia.autor.model.SubjectManagerProxy;
    import br.com.optimedia.autor.view.components.SendFilePopUp;

    import flash.events.Event;

    import mx.controls.Alert;
    import mx.events.CloseEvent;

    import org.puremvc.as3.multicore.interfaces.INotification;
    import org.puremvc.as3.multicore.patterns.mediator.Mediator;

    public class SendFilePopUpMediator extends Mediator
    {
        public static const NAME:String = 'SendFilePopUpMediator';

        private var proxy:RepositoryManagerProxy;
        private var subjectManagerProxy:SubjectManagerProxy;

        public function SendFilePopUpMediator(viewComponent:Object=null)
        {
            super(NAME+viewComponent.uid, viewComponent);
        }

        override public function onRegister():void
        {
            trace(NAME+'.onRegister()');
            view.addEventListener(SendFilePopUp.UPLOAD_FILE_EVENT, sendFile);
            view.addEventListener(CloseEvent.CLOSE, closeMe);
            proxy = facade.retrieveProxy( RepositoryManagerProxy.NAME ) as
RepositoryManagerProxy;
            subjectManagerProxy = facade.retrieveProxy( SubjectManagerProxy.NAME ) as
SubjectManagerProxy;
        }

        override public function onRemove():void {
            view.removePopup();
        }

        public function get view():SendFilePopUp
        {
            return viewComponent as SendFilePopUp;
        }

        override public function listNotificationInterests():Array
        {
            return [NotificationConstants.UPLOAD_PRESENTATION_FILE_RESULT];
        }

        override public function handleNotification(note:INotification):void
    }
}
```

```
{  
    switch (note.getName())  
    {  
        case NotificationConstants.UPLOAD_PRESENTATION_FILE_RESULT:  
            Alert.show("Arquivo enviado com sucesso.", "OK");  
            subjectManagerProxy.getSubjects();  
            closeMe(null);  
            break;  
        default:  
            break;  
    }  
}  
  
private function closeMe(event:CloseEvent):void {  
    sendNotification(CommandConstants.SEND_FILE_POPUP_DISPOSE, view);  
}  
  
private function sendFile(event:Event):void {  
    proxy.uploadPresentationFile(view.fileVO, view.presentationID, view.fileType);  
}  
}
```