

APLICACIONS INTERACTIVES MULTIPLATAFORMA

PRÀCTICA #1



Nom:

Aitor Javier

Cognoms:

Santaeugenia Marí

Data lliurament:

27/11/2016

INDEX

ETAPES	Pàgina 3
ETAPA 1	Pàgina 3
ETAPA 2	Pàgina 4
ETAPA 3	Pàgina 5
ETAPA 4	Pàgina 9
ETAPA 5	Pàgina 10
CONCLUSIO	Pàgina 13
BIBLIOGRAFIA	Pàgina 14
RECURSOS	Pàgina 14

ETAPES

Etapa1 | Creació de les classes per a les pantalles i associació amb els seus asets gràfics (símbols ja creats en *AnimateCC*)

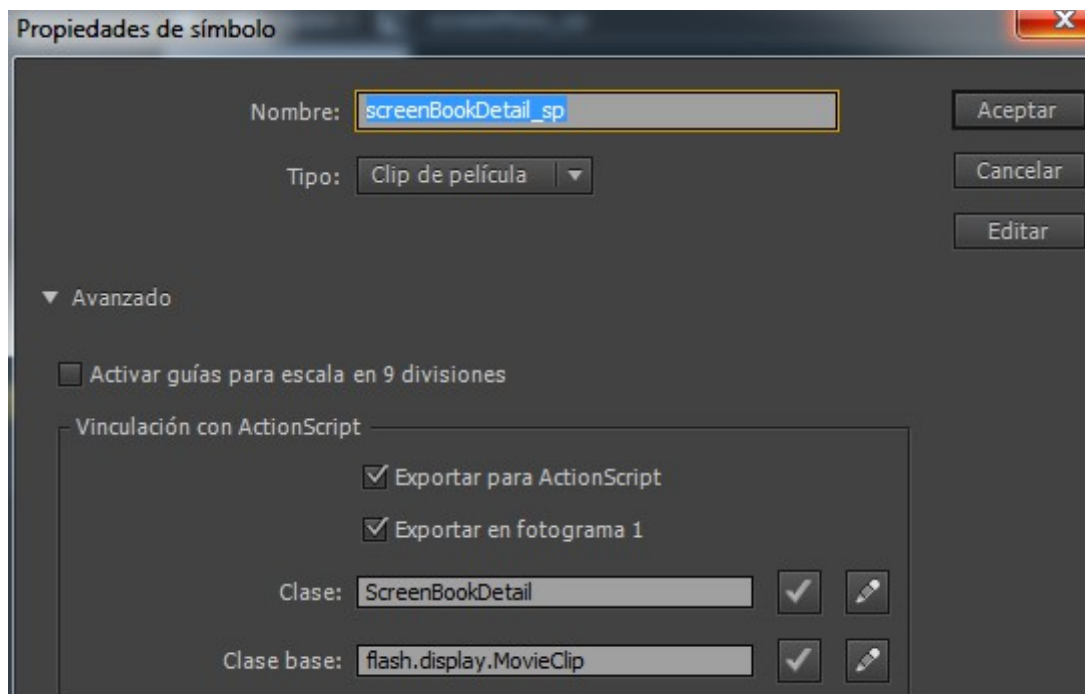
Creareu les classes per a les pantalles: pantalla Inici, pantalla "Cercar", pantalla "Resultat cerca" i pantalla "Detall llibre". Dit més precisament, i per utilitzar tots els mateixos noms per a les classes, haureu de:





- Crear la classe **ScreenMenu** i associar-la al seu símbol **screenMenu_sp**
- Crear la classe **ScreenSearch** i associar-la al seu símbol **screenSearch_sp**
- Crear la classe **ScreenSearchResult** i associar-la al seu símbol **ScreenSearchResult_sp**
- Crear la classe **ScreenBookDetail** i associar-la al seu símbol **screenBookDetail_sp**

En el .fla del projecte, fent *Ctrl* + *L* (*Cma* + *L* en Mac) podreu veure, a la carpeta "screen" de la llibreria, els diferents asets gràfics (símbols) per a cadascuna de les pantalles.

Important: per a aquesta primera part de la Pràctica Final no tractarem encara el tema de favorits.

- Creem les classes i las vinculem de la següent forma, i les obrim mitjançant "*Flash builder*" en el nostre cas



	screenBookDetail_sp	ScreenBookDetail
	screenMenu_sp	ScreenMenu
	screenSearch_sp	ScreenSearch
	screenSearchResult_sp	ScreenSearchResult

Etapa2 | Navegació entre pantalles

Una de les funcions principals que tindrà la nostra classe del document (Main) serà la de ocupar-se de posar la pantalla que correspongui en tot moment. Més endavant, pel que fa el funcionament de cada pantalla procurarem incloure-ho a la pròpia classe de cada pantalla.

Modificar Main perquè s'ocupi de la gestió de les pantalles:

Important: a la *timeline* del *.fla* del projecte veureu que tenim ja un objecte *screenContainer_sp*. Posarem les pantalles dins d'aquest objecte. Això ens permet tenir una *displayList* més ordenada i ens facilita poder comprovar que sempre tenim un únic fill dins de la *displayList* de *screenContainer_sp*.

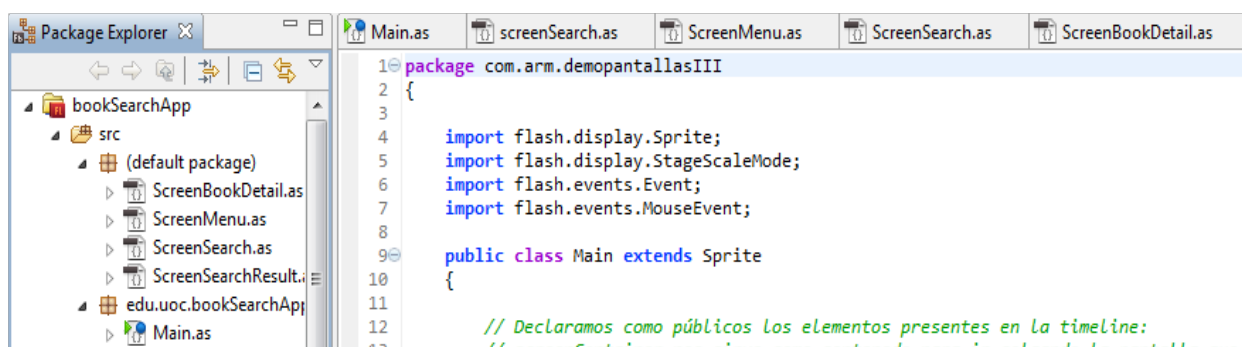
Important2: amb l'enunciat trobareu un projecte d'exemple (demopantallas2) per veure com controlar el canvi de pantalla des de la classe del document.

En aquesta primera part de la Pràctica creareu la navegació bàsica:

- A la pantalla inicial: en fer clic sobre "Cercar Llibre" canviarem a la pantalla de cerca. Per ara el clic sobre "Els meus favorits" no farà res.
- A la pantalla de cerca: en fer clic sobre "Cercar Llibre" arribarem a la pantalla de resultat de cerca. També podem tornar enrere (a la pantalla menú), amb la fletxa superior esquerra.
- A la pantalla de resultat d'una recerca: en fer clic sobre qualsevol ítem de la llista arribarem a la pantalla detall d'un llibre. També podrem tornar enrere (pantalla resultat de la recerca) amb la fletxa superior esquerra.
- A la pantalla de detall d'un llibre: podrem tornar al llistat de resultats amb la fletxa superior esquerra.

Important: Recordeu que per a aquesta primera part de la Pràctica no tractarem encara el tema de favorits. Recordeu també que tot el control de canvi de pantalles es realitza des de la classe del document.

- S'ha realitzat correctament tota la navegació, i aquesta ha estat realitzada des de el *main.as*. El més complicat es saber diferenciar i controlar els *MovieClips* i els *Button* o *SimpleButton*.
- També cal saber diferenciar els tipus de variables que tenim que crear i vincular-los amb els elements del *.fla* (on alguns caps tenen *<instancia>* per nom, i tenim que tractar-los amb la nomenclatura del programa).



Diferents classes creades per l'exercici i el control d'aquestes al main.as

Etapas 3 | Event RESIZE (totes les pantalles)

A l'hora de crear una aplicació *Multidispositius* un dels problemes principals amb què ens trobem és la multiplicitat de dispositius del mercat actual. La nostra aplicació s'ha d'adaptar a diferents dimensions, resolucions i/o proporcions. D'altra banda la nostra aplicació hauria també d'adaptar el seu *layout* per a qualsevol canvi d'orientació.

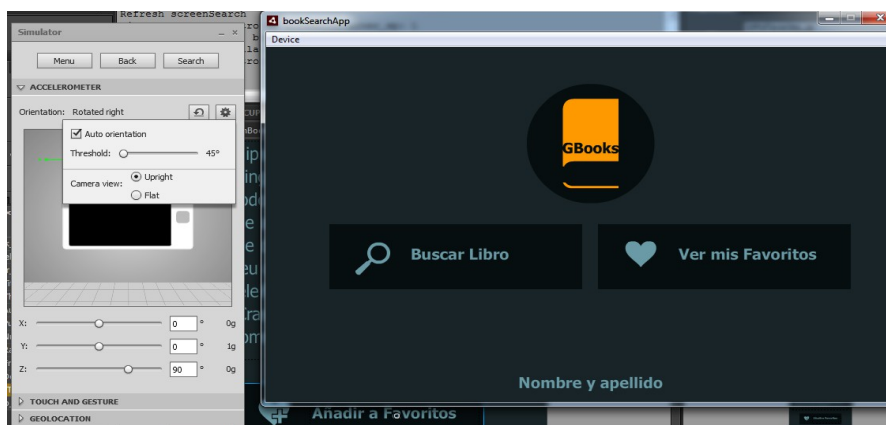
Un *layout* que funciona en mode *portrait* pot no funcionar en mode *Landscape*, i viceversa.

En aquesta etapa haureu de modificar les diferents classes que controlen cadascuna de les pantalles perquè escoltin l'esdeveniment **Event.RESIZE** (això ens permetrà tant detectar les diferents resolucions com un canvi d'orientació) i executin llavors el mètode **resizeHandler()** de la classe que resituarà els elements de la pantalla com correspongui.

Important: de nou, adjunt amb l'enunciat, trobareu un projecte d'exemple d'ús de **Event.RESIZE** que us hauria d'orientar per resoldre aquesta etapa.

- **Imatges LANDSCAPE**

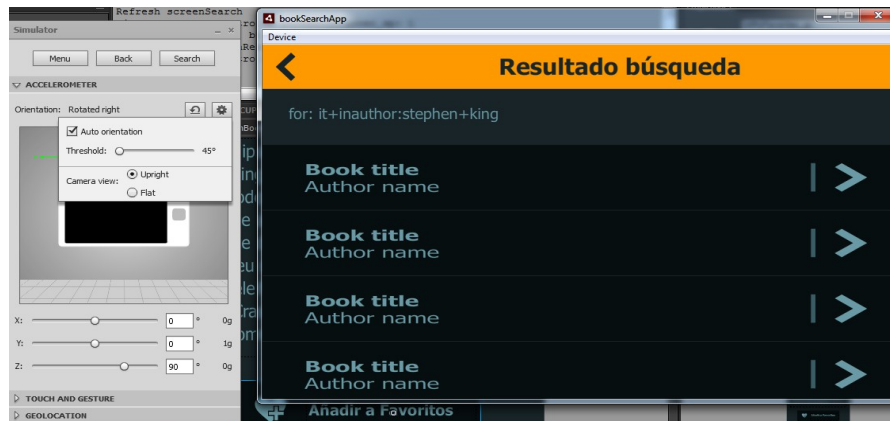
- *screenMenu*



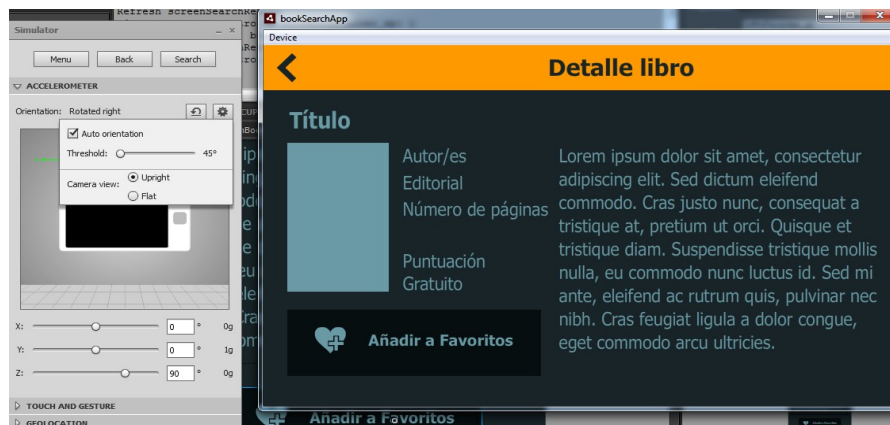
- *screenSearch*



– *screenSearchResult*

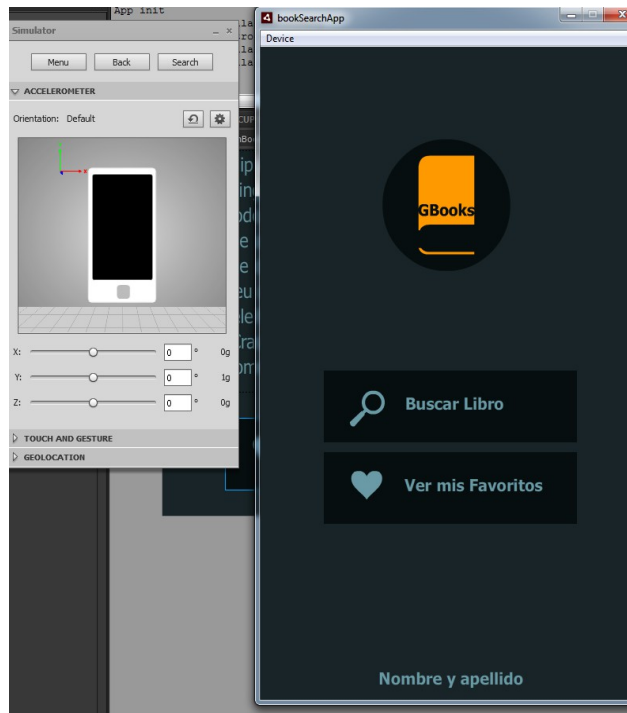


– *screenBookDetail*

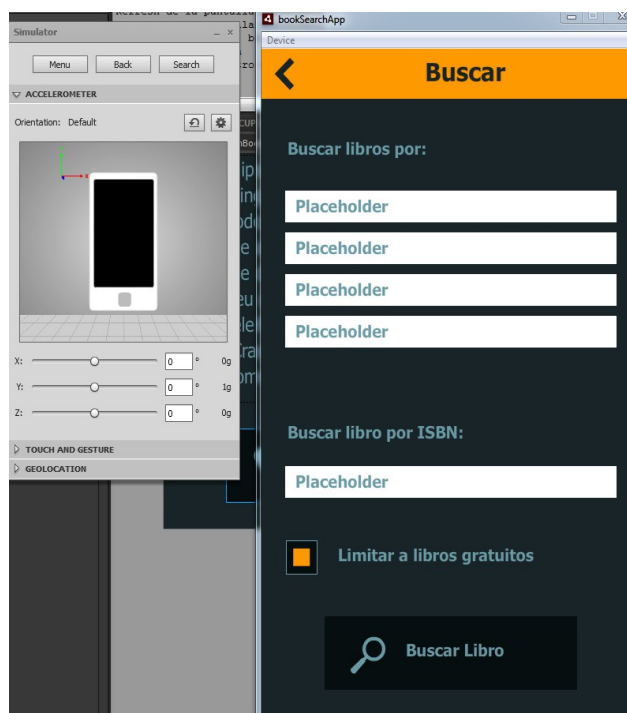


– **Imatges *PORTRAIT***

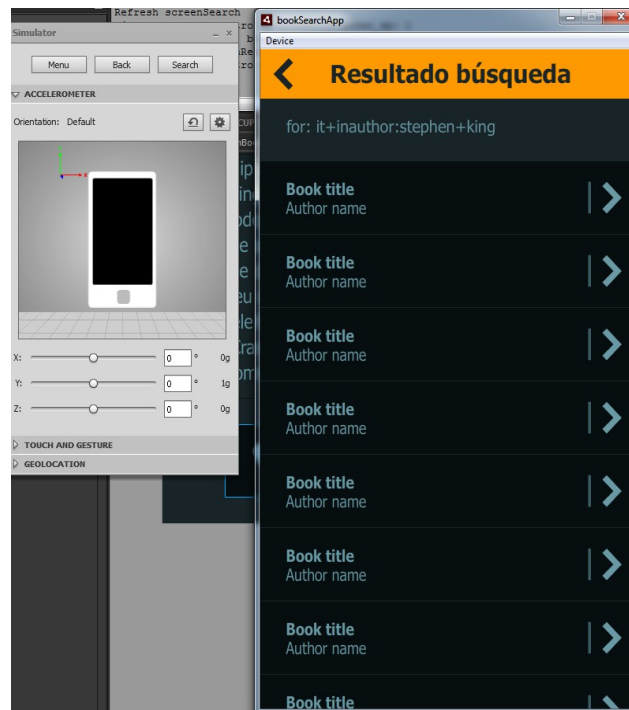
– *screenMenu*



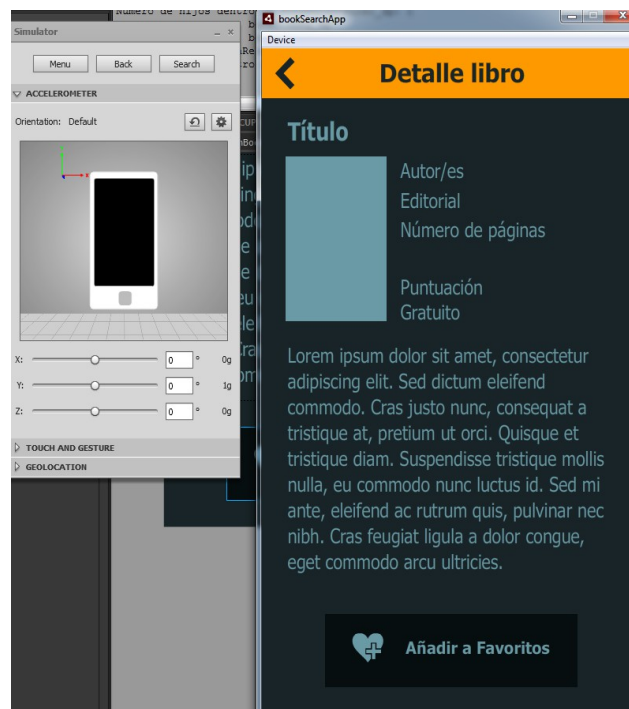
– *screenSearch*



— *screenSearchResult*



— *screenBookDetail*



- Hem realitzat el *refreshScreen* en cada una de les diferents classes creades per les diferents pantalles
- Cada *refreshScreen* canviarà les mides dels elements per pantalla per tal de deixar-ho correcte per el funcionament tant de *LANDSCAPE* com en *PORTRAIT*.
- Hem afegit el *public function refreshScreen(ev:Event=null):void {}* per tal de que pugui passar el valor de *null* en les diferents navegacions entre pantalles (afegit a les diferents classes de pantalles).

```
public function refreshScreen(ev:Event=null):void {
```

- Al mateix temps, afegim dins de cada *public function* (és a dir dins de *screenMenu*, *screenSearch* i cada pantalla) el *listener* del *refreshScreen*.

```
//ESCOLTADOR PER EL REFRESH
addEventListener(Event.ADDED_TO_STAGE, refreshScreen);
```

- Aquest dos punts anteriors el realitzem per tal de que la navegació per pantalla amb canvis de "resolució", és realitzi correctament, ja que sense escoltador el canvi de *PORTRAIT* a *LANDSCAPE* no es realitzava correctament (tenies que passar pantalla i modificar manualment la pantalla perquè funcionés).

Etap4 | Creació classes suplementàries i associació amb els seus asets gràfics

Ja tenim una base de l'aplicació feta. Podem navegar entre pantalles i aquestes s'adapten al nostre dispositiu. Ara vegem si podem optimitzar certs elements.

Un dels avantatges de la POO és la seva modularitat, permetre la creació de 'components' que podem reutilitzar. Components que o bé tenen un comportament particular o que utilitzem més d'una vegada. Seria doncs interessant crear una classe particular per a ells i els seus símbols (assets gràfics)




Per a aquesta part de la pràctica veiem com a mínim 3 casos on això pot ser útil:

- A la pantalla "Cercar": l'input d'entrada de dades, que apareix repetit diverses vegades i té sempre el mateix comportament.
- A la pantalla "Cercar": un component de tipus *checkbox* que bé podríem utilitzar en altres aplicacions.
- A la pantalla "Resultat recerca": l'ítem d'un llibre, que apareix repetit diverses vegades i té sempre el mateix comportament.

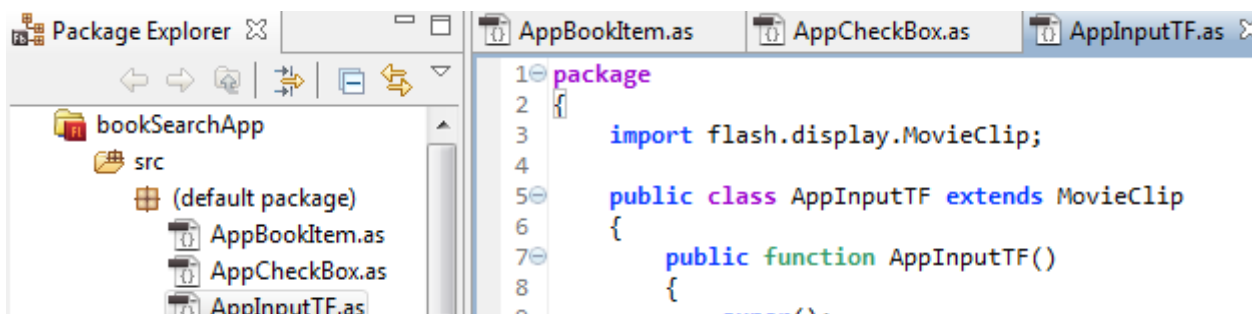
En aquesta etapa haureu de:

- Crear la classe ***AppInputTF*** i associar-la al seu símbol ***appInputTF_sp***
- Crear la classe ***AppCheckBox*** i associar-la al seu símbol ***appCheckBox_sp***
- Crear la classe ***AppBookItem*** i associar-la al seu símbol ***appBookItem_sp***

- Feim el mateix en que en la [etapa 1](#)

Nombre	Vinculació	Número d...
 appBookItem_sp	AppBookItem	8
 appCheckBox_sp	AppCheckBox	1
 appInputTF_sp	AppInputTF	5

- Podem veure com han quedat vinculats i les classes creades



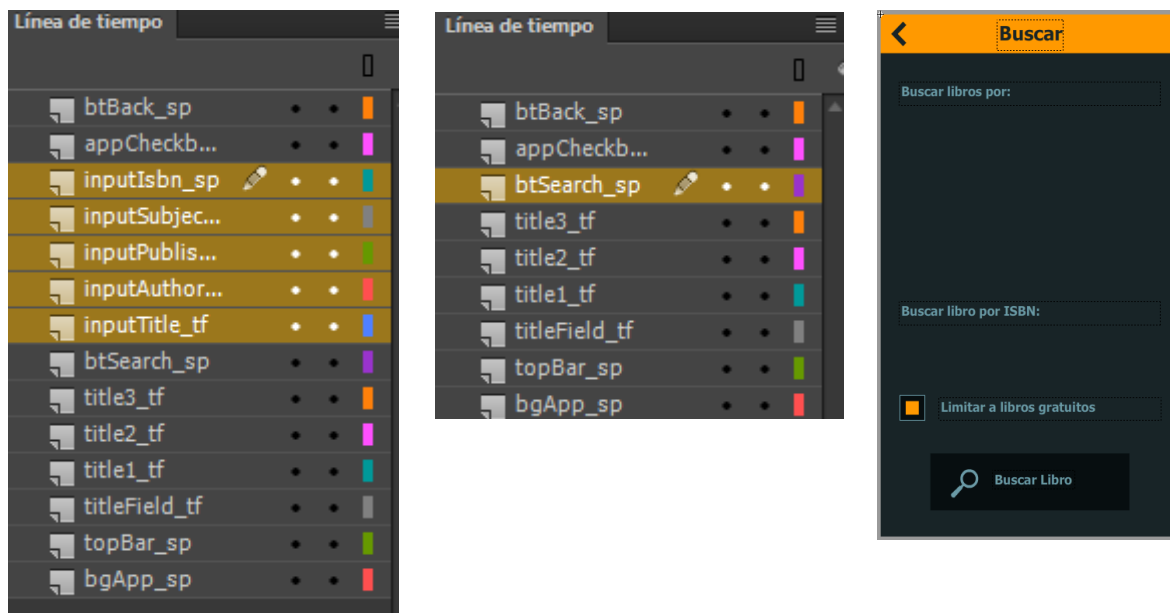
Etapa5 | Pantalla "Cercar" ús de la classe *AppInputTF*

Aquesta classe hauria de rebre com a paràmetre el text que volem que aparegui com a *placeholder*. Hauria de permetre a l'usuari introduir el text que li interessi (en fer clic sobre ell el text de *placeholder* desapareix i si l'usuari no introdueix cap contingut el text de *placeholder* reapareix en sortir del camp). Finalment hauria de tenir una funció pública *getInputTxt()* per poder recuperar el text introduït al camp.

Haureu de crear la classe *AppInputTF* per a que actuï d'aquesta forma.

Un cop fet haureu:

- Dins *AnimateCC* i dins el símbol "*screenSearch_sp*" eliminar els camps d'input presents a la *timeline*
- Primer de tot eliminem el que ens demana l'exercici. Per això ho deixem com en les fotografies de a continuació.



- Dins de la classe *ScreenSearch* crear els diferents input: crear les diferents instàncies de *AppInputTF*, situar-les i afegir-les a la *displayList* de la pantalla.

Problema no resolts:

- Si **eliminem els comentaris a la línia nº 85** (*addInput()*) i **línia nº209 a 288** (*private function addInput(ev:Event=null):void { }*) **del screenSearch.as**, i al mateix temps feim el punt 1 d'aquest exercici (és a dir, **eliminem els input del screenSearch**), podem veure com ens col·loca els input (sense els *placeholders*), però si ho fem en mode "*landscape*" els input s'allarguen sempre mitjançant l'amplada de tot el *stage*.

Per aquest motiu, **varem voler enviar una versió amb els input dels *timeline presents***, per poder veure com el *refreshScreen* funciona perfectament (fins que no varem treballar amb l'apartat 5).

```
private function init(ev:Event):void {

    // Hacemos antes un poco de limpieza
    // y eliminamos el escuchador que ya no utilizamos
    removeEventListener(Event.ADDED_TO_STAGE,init);

    // Siempre es aconsejable gestionar nosotros mismos el escalado de nuestra aplicación.
    // Por ello situaremos siempre nuestra aplicación en la esquina superior izquierda
    stage.align = StageAlign.TOP_LEFT;
    // y inhabilitaremos el escalado automático
    stage.scaleMode = StageScaleMode.NO_SCALE;

    // Añadimos el escuchador de eventos para detectar cualquier cambio del as medidas de la pantalla
    // Nota: Esto incluye los cambio de orientación.
    stage.addEventListener(Event.RESIZE, refreshScreen);
    //stage.addEventListener(Event.RESIZE,

//addInput();

}

//TESTING PUNT 5
/*private function addInput(ev:Event=null):void {
    //VARIABLES PER EL PLACEMENT
    // Para más comodidad recuperamos las nuevas medidas de nuestra aplicación
    var stageW:int = stage.stageWidth;
    var stageH:int = stage.stageHeight;
    // También puede ser interesante el uso de una variable
    // para obtener espaciados iguales entre ciertos elementos
    var gutter:int = 30;

    // Adaptamos el fondo al espacio disponible
    bgScreen_sp.width = stageW;
    bgScreen_sp.height = stageH;

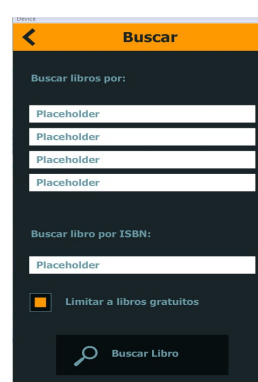
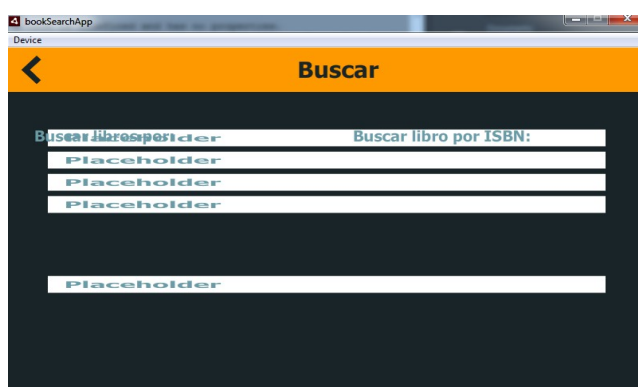
    //var _placeholder = "Titulo";
    //-----
    //bookTitle_appInputTF
    var bookTitle_appInputTF:AppInputTF = new AppInputTF("Titulo");

    bookTitle_appInputTF.width = 420;
    bookTitle_appInputTF.height = 38;
    bookTitle_appInputTF.x = 30;
    bookTitle_appInputTF.y = 170;
}
```

- Tot això per culpa d'aquest error, que no hem sabut com solucionar-ho:

An ActionScript error has occurred:
 TypeError: Error #1010: A term is undefined and has no properties.
 at screenSearch/refreshScreen() [C:\Users\FRKA\Desktop\PR1 ARM\bookSearchApp true\src\screenSearch.as:190]
 at flash.display::DisplayObjectContainer/addChild()
 at edu.uoc.bookSearchApp::Main/refreshScreen() [C:\Users\FRKA\Desktop\PR1 ARM\bookSearchApp true\src\edu\uoc\bookSearchApp\Main.as:195]
 at edu.uoc.bookSearchApp::Main/changeScreen() [C:\Users\FRKA\Desktop\PR1 ARM\bookSearchApp true\src\edu\uoc\bookSearchApp\Main.as:172]

- Entenc que l'error està dins del *screenSearch* i si mirem la línia podem veure com es dins del *refreshScreen*, possiblement en la cridada dels Input que acabem de crear, però ho hem intentat de varies maneres i no hi ha mode de que el *refreshScreen* funcioni correctament en aquesta etapa.
- Així és com es veu la pantalla *screenSearch* després de treballar a l'etapa 5 amb el error anterior, en *LANDSCAPE* malament, però en *PORTRAIT* perfecte.



- Al mateix temps, hem introduït un *tracer(_placeholder)* per veure si ens agafava els noms dels *_placeholders* i els passava als diferents inputs creats, i podem veure com si que ho feia correcte.

Titulo
Autor
Editorial
Tematica
ISBN

Sortida consola:

tracer(_placeholder)

```
public function AppInputTF(input_tf)
{
    //__placeholder=placeholder;
    _input_tf=input_tf;
    //trace(_input_tf);

    super();
    // We assure the class has been instantiated
    // and added to the application displayList
    addEventListener(Event.ADDED_TO_STAGE, init);
    getInputTxt(_input_tf);
}

//FUNCIO QUE ENS DEMANA L'ANUNCIAT
public function getInputTxt(_input_tf)
{
    var _placeholder = _input_tf;
    //IMPRIMEIX TOTS ELS PLACEHOLDER
    trace(_placeholder);

    // Nos aseguramos que la clase del documento ha sido instanciada
    // y añadida a stage
    addEventListener(Event.ADDED_TO_STAGE, init);
}

private function init(ev:Event):void {
    // Hacemos antes un poco de limpieza
    // y eliminamos el escuchador que ya no utilizamos
    removeEventListener(Event.ADDED_TO_STAGE, init);

    //getInputTxt();
}
```

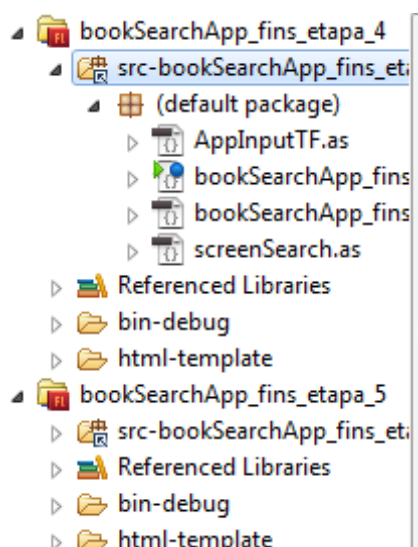
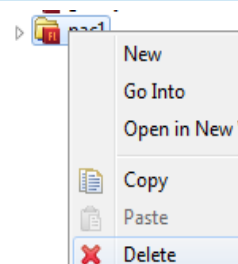
CONCLUSIÓ

El treball ha sigut complicat per varis motius. Primer de tot, que vaig voler eliminar el primer exemple que apareixia en el *Flash Builder* anomenat *bookSearchApp* (el del nostre exercici), ja que vaig voler guardar en diferents etapes a partir de la 4 (és a dir, *bookSearchApp_etapa4* i *bookSearchApp_etapa5*, *bookSearchApp_etapa6* i *bookSearchApp_etapa7*).

El problema es que al eliminar el primer, hem varen petar totes les posteriors còpies d'aquest, ja que el copiar i enganxar no és un mètode correcte per realitzar una còpia d'un projecte existent, ja que comparteixen els documents.

Així doncs vaig tenir que realitzar tot el treball l'últim dia de la entrega, ja que l'opció de restaurar que apareix, no et recupera tots els documents (només algunes classes que hagin quedat en historial o temporal). Només fent fins la **etapa 5** ja que només vaig recuperar els *screenRefresh*.

Hem servirà de lliçó segurament aprendre a realitzar còpies i no instàncies d'un projecte.



Restauració dels documents, podem veure que només hi ha 3 classes de totes les que hi hauria d'haver

Bibliografia

- GÓNZALEZ SANCHO, MARCOS. (2016). "*Introducció i conceptes generals*". [en línia]. Catalunya: Universitat Oberta de Catalunya.
 - http://materials.cv.uoc.edu/continguts/PID_00192290/web/main/m1/portada.html
- GÓNZALEZ SANCHO, MARCOS. DE FUENMAYOR LÓPEZ, DANIEL. (2016). "*Cicle de desenvolupament d'una aplicació Rich Media*". [en línia]. Catalunya: Universitat Oberta de Catalunya.
 - http://materials.cv.uoc.edu/continguts/PID_00192290/web/main/m2/portada.html
- DE FUENMAYOR LÓPEZ, DANIEL. (2016). "*Desenvolupament d'aplicacions Rich Media en la Plataforma Flash*". [en línia]. Catalunya: Universitat Oberta de Catalunya.
 - http://materials.cv.uoc.edu/continguts/PID_00192290/web/main/m3/portada.html
- GÓNZALEZ SANCHO, MARCOS. (2016). "*Publicació i model de negoci*". [en línia]. Catalunya: Universitat Oberta de Catalunya.
 - http://materials.cv.uoc.edu/continguts/PID_00192290/web/main/m4/portada.html
- GÓNZALEZ SANCHO, MARCOS. (2013). "*Tecnologías para desarrollo de aplicaciones rich media*". [en línia]. Catalunya: Universitat Oberta de Catalunya.
 - <http://mosaic.uoc.edu/2013/07/31/tecnologias-para-desarrollo-de-aplicaciones-rich-media/#>
- MONCHO MAS, VICENT. (2016). "*Introducció a la programació orientada a objectes*". [en línia]. Catalunya: Universitat Oberta de Catalunya.
 - http://materials.cv.uoc.edu/continguts/PID_00220463/index.html
- QODE. (2014). "*Què es una webapp?*". [en línia]. Qode.pro.
 - <http://qode.pro/blog/que-es-una-web-app/>
- ROUSE, MARGARET. (2014). "*Rich Internet Application (RIA)*". [en línia].
 - <http://searchsoa.techtarget.com/definition/Rich-Internet-Application-RIA>
- ANÒNIM. (2012). "*¿Qué es RIA – Rich Internet Applications?*". [en línia]. PortalSaaS.com.
 - <http://blog.portalsaas.com/que-es-ria-rich-internet-applications/>
- MOOK COLIN. (2011). "*Essential ActionScript 3.0*". Editorial O'Reilly 2011.

Recursos Multimèdia

- Logotipus corporatiu de la Universitat Oberta de Catalunya. "*UOC logotipo azul papel*". [Imatge]. <http://www.uoc.edu/portal/es/universitat/coneix/marca/logotip-paper/index.html>
- Formació d'una Aplicació d'Internet enriquida "*RIA – Rich Internet Applications*", per *portalSaas.com*. [Imatge]. <http://blog.portalsaas.com/que-es-ria-rich-internet-applications/>
- La resta d'imatges són de creació pròpia.