**Convención de Desarrollo**

**Idioma.-** El idioma a utilizar será el inglés debido a su practicidad y la facilidad para describir cuestiones técnicas. Otro motivo por el cual se desarrollara en Ingles es la posibilidad de una compañía extranjera adquiriendo nuestro sistema en un futuro.

**Variables.-** Las variables se escribirán usando cammel case en donde la primera letra de cada palabra será mayúscula con excepción de la primera letra de la primera palabra. Esto es para diferenciar una variable de una clase. Ejemplo *counter* o *personCollection*

Al nombrar variables lo más importante es que sean descriptivas y adecuadas, por lo que no usaremos abreviaciones. Es decir usaremos palabras completas aunque los nombres sean más largos.

**Variables privadas/protegidas.-** Los nombres de variables privadas o protegidas siguen el mismo estándar que las variables públicas. La única diferencia es que siempre contienen un guion bajo al inicio. Ejemplo *\_counter* o *\_personCollection*

La preferencia es siempre crear variables protegidas (protected). Ya que estas pueden ser accedidas por subclases. Cuando no se ha determinado la visibilidad de una variable SIEMPRE se declara protegida por default para promover encapsulación y si en el futuro es necesario se incrementa la visibilidad a pública. El caso contrario es muy difícil de ejecutar ya que otras clases pueden ser dependientes de variables públicas.

**Funciones.-** Los nombres de funciones siguen el mismo estándar que las variables públicas

**Paquetes.-** Los nombres de paquetes se escribirán únicamente con letras minúsculas siguiendo el esquema más usado por la industria – nombre de dominio inverso- que garantiza el menor número de colisiones entre clases. Ejemplo: *com.random.utils* o *com.random.model*

**Clases.-** Los nombres de clases se escribirán siempre con mayúscula inicial. Para nombres compuestos de más de una palabra se utilizara cammel case. Ejemplo: *Main* o *SelectionManager.*

**Constantes.-** Los nombres de constantes se escribirán únicamente en mayúsculas. Para nombres compuestos de más de una palabra se separan las palabras por medio de un guion bajo. Ejemplo *COMPLETE\_EVENT* o *LOAD\_ERROR*

**Tipo de un evento.-** Para el tipo de un evento siempre utilizaremos una constante estatica dentro de la clase que define el evento. La constante contenera un string que representa la misma palabra en cammel case en el mismo formato que una variable pública. Ejemplo

*public static const LOAD\_COMPLETE:String = “loadComplete”;*

Este es el estándar que sigue Flex Builder y nos permite sacar el mayor provecho del IDE y el code hinting.

**Metadata de un evento.-** Siempre que una clase dispare un evento deberemos incluir metadata en la cabezera de la clase para indicar al compilador que esa clase es capaz de disparar ese tipo de eventos. Esto resulta en el los eventos apareciendo en el code hinting al tratar de agregar un listener. Ejemplo:

*[Event(name=”loadComplete”, type=”com.random.events.GraphicEvent”)]*

*public class GraphicLoader {….*