DOCUMENTACION – DESARROLLO DE SOFTWARE

LINEAMIENTOS DE PROGRAMACIÓN

**CONFIGURACION WOKSPACE**

**ECLIPSE**

# Objetivo

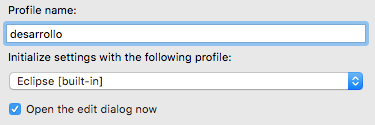
En este documento se presenta la configuración de la herramienta de desarrollo Eclipse que se debe aplicar por parte de cada programador con la finalidad de homologar y automatizar en cierto grado la aplicación de los lineamientos y mejores prácticas de programación en Java.

# Creación de un perfil de desarrollo

Dentro de las distintas opciones que ofrece Eclipse en cuanto a formateo de código, se encuentra el concepto de “Perfil”. Un perfil asocia a un conjunto de configuraciones y preferencias de formateo de código fuente, el cual puede ser aplicado a uno o más proyectos dentro de un mismo espacio de trabajo.

Antes de iniciar con las configuraciones descritas en esta sección, es conveniente crear un nuevo perfil a partir del cual se realizarán dichas configuraciones. Este perfil será el empleado por todos los proyectos definidos en Eclipse. Para crear un nuevo perfil, seguir las siguientes instrucciones:

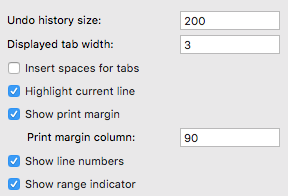
* Dentro de eclipse seleccionar la opción Window-> Preferences del menu seleccionar Java->Code Style ->Formatter
* Hacer clic en New, configurar un nuevo perfil llamado desarrollo como se muestra a continuación:



* Seleccionar Ok.
* Este perfil será empleado para aplicar las configuraciones descritas a continuación que tengan relación con el formato del código fuente.

## Configuración general de editores.

* Seleccionar General->Editors->Text Editors Configurar como se muestra a continuación.



Para mostrar el número de líneas, adicionalmente se puede configurar dando clic derecho en el margen izquierdo del editor, seleccionar show line numbers.

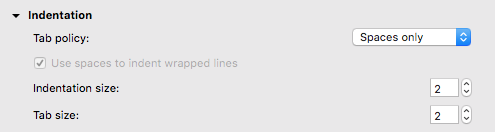
## Formato de código

* Dentro de eclipse seleccionar la opción Window-> Preferences del menu
* Seleccionar Java->Code Style->Formatter ->Botón Edit

Aplicar las configuraciones siguientes.

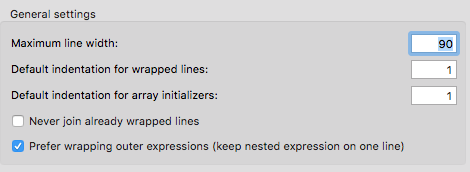
### Indentación

* Seleccionar la pestaña “Indentation”, configurar como se muestra a continuación:



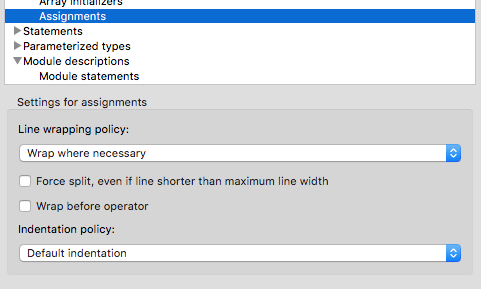
### Longitud de la línea

* Seleccionar la pestaña “Line wrapping” configurar como se muestra a continuación:



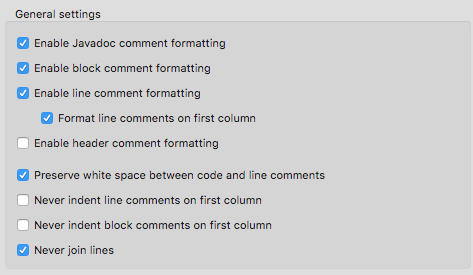
* En las opciones de la parte inferior, configurar como se muestra a continuación:

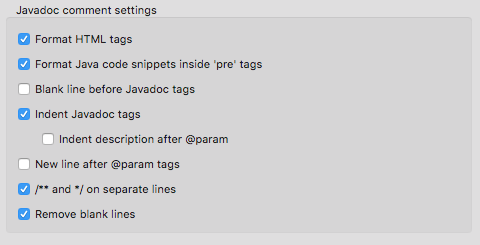
Assignments:

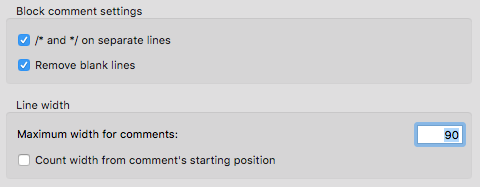


### Formato de comentarios.

* Seleccionar la pestaña “Comments”, configurar como se muestra a continuación:







# Comentarios y documentación de código.

La documentación del código fuente deberá realizarse en inglés. Seguir los siguientes pasos para realizar la configuración.

* Seleccionar la opción Window-> Preferences del menú
* Seleccionar Java -> Code Styles –> Code templates

Aplicar las configuraciones siguientes.

## Comentarios por default.

* Al final de la ventana seleccionar la siguiente opción:



## Encabezado de la clase.

* Seleccionar Comments->Files, presionar “Edit”. Capturar el siguiente texto:

/\*\*

\* ${file\_name}

\* Creation Date: ${date}, ${time}

\*

\* Copyright (C) The Project \*${project\_name}\* Authors.

\*

\* This software was created for didactic and academic purposes.

\* It can be used and even modified by referring to the author

\* or project on GitHub. If the file is modified, add a note

\* after this paragraph saying that this file is a modified version.

\*

\* The above copyright notice and this permission notice shall be

\* included in all copies or substantial portions of the Software.

\*/

* Seleccionar Comments->Types, presionar “Edit”. Capturar el siguiente texto:

/\*\*

\* TODO [Add class documentation]

\* ${tags}

\*/

* Seleccionar Comments->Fields, presionar “Edit”. Capturar el siguiente texto:

/\*\*

\* TODO [Add attribute documentation]

\*/

* Seleccionar Comments->Constructors, presionar “Edit”. Capturar el siguiente texto:

/\*\*

\* TODO [Add constructor documentation]

\* ${tags}

\*/

* Seleccionar Comments->Methods, presionar “Edit”. Capturar el siguiente texto:

/\*\*

\* TODO [Add method documentation]

\* ${tags}

\*/

***Importante:***  Todos los métodos y constructores deben tener este encabezado. Si el código es demasiado simple y no requiere de alguna documentación adicional, se puede omitir la descripción, pero el encabezado se debe conservar.

* Seleccionar Comments->Overriding methods, presionar “Edit”. Capturar el siguiente texto:

/\* See the original documentation of the method declaration.

\* ${see\_to\_overridden}

\*/

***Importante:*** Todos los métodos que implementen a los métodos de una interfaz, clase abstracta o que sobrescriban un método, deberán tener este encabezado. Si se desea agregar documentación adicional, el encabezado se modifica de la siguiente manera:

/\*\*

\* See the original documentatio of the method declaration.

\* **<Add here additional documentation>**

\* ${see\_to\_overridden}

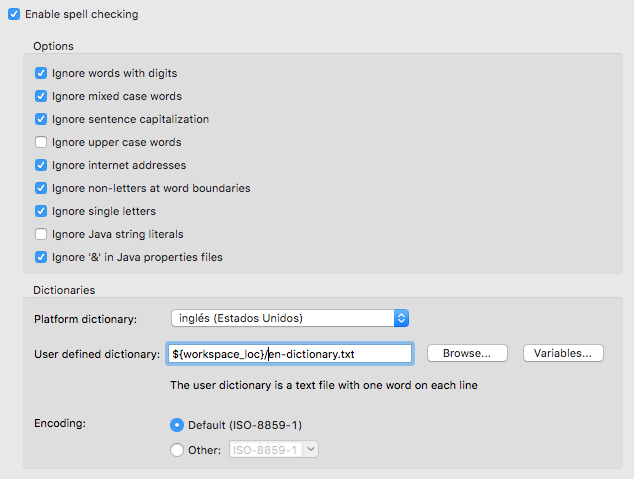
\*/

Observar que se agrega un “\*”

## Configuración del diccionario.

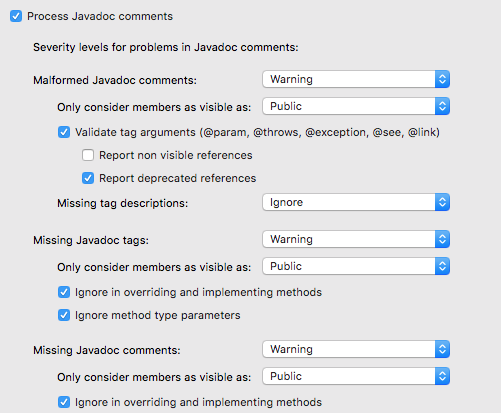
Con la finalidad de reducir faltas de ortografía en los comentarios en inglés, habilitar y configurar el diccionario como se muestra a continuación.

* Seleccionar la opción Window-> Preferences del menú
* Seleccionar General -> Editors –> Text editors ->Spelling



## Configuración de detección de comentarios incompletos o faltantes.

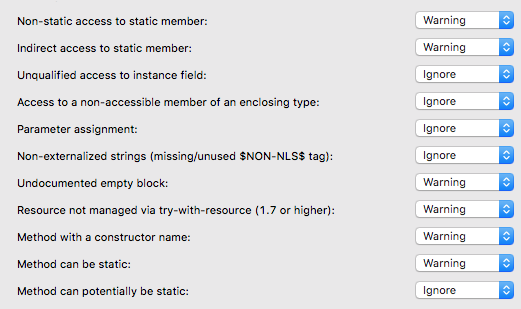
* Seleccionar Window-> Preferences del menu
* Seleccionar Java->Compiler->Javadoc , aplicar la configuración que se muestra a continuación.



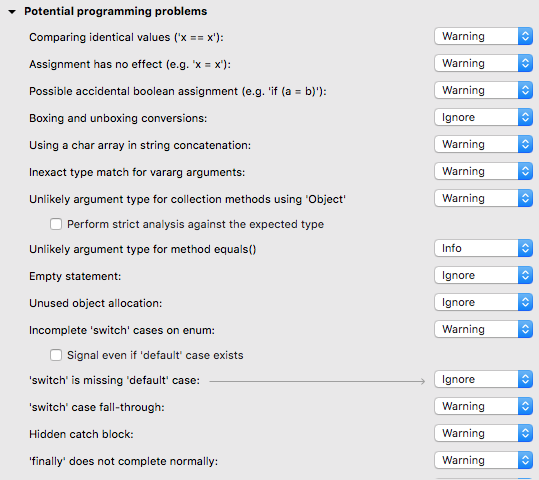
***Tip:***  Para agilizar la documentación de elementos, basta con situarse en el elemento que se desea documentar, presionar Alt + Shift + J para agregar un bloque de comentarios.

# Aspectos generales del estilo de programación

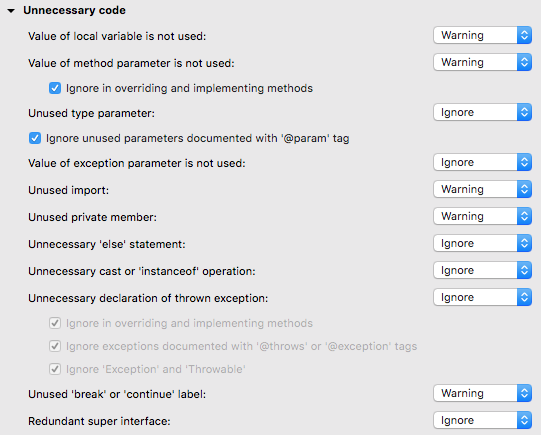
* Seleccionar Window-> Preferences del menu
* Seleccionar Java->Compiler->Errors/Warnings ->Code Style , aplicar la configuración que se muestra a continuación:



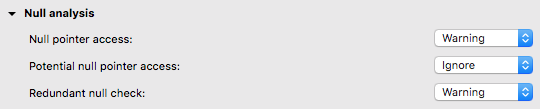
* Seleccionar la opción Potential Programming Problems, configurar como se muestra a continuación.



* Seleccionar la opción Unnecessary Code, configurar como se muestra a continuación



* Seleccionar la opción Null Analysis, configurar como se muestra a continuación



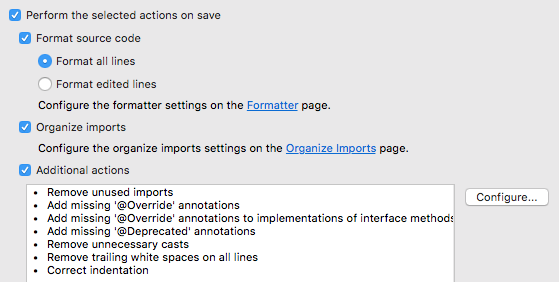
# Acciones al salvar un archivo

Las siguientes acciones serán ejecutadas al salvar un archivo Java:

* Formateo automático de la clase
* Organización de imports
* Adición de llaves para sentencias if/do/while/for que hayan sido omitidas (sentencias con una sola instrucción).

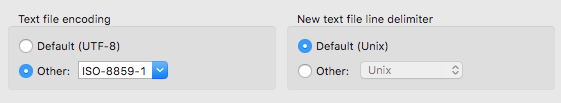
Para realizar esta configuración realizar los siguientes pasos:

* Seleccionar la opción Window-> Preferences del menu
* Seleccionar Java->Editor ->Save Actions,
* Aplicar la configuración que se muestra a continuación, hacer clic en ´Configure’.



# Configuración del juego de caracteres

* Seleccionar la opción Window-> Preferences del menu
* Seleccionar General -> Workspace, configurar el juego de caracteres como se muestra a continuación.



# Recomendaciones generales.

Antes de enviar una revisión de código o de realizar un commit, se deberá revisar lo siguiente:

* El código modificado debe estar probado, agregar pruebas en caso de ser necesario.
* El código no debe contener “Warnings”. Se deberán eliminar antes de hacer commit.

# Plugins instalados

Los siguientes plugins serán instalados en el workspace.

## Object AId

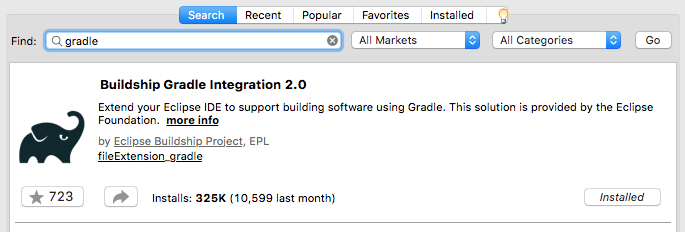
Empleado para realizar diagramas UML: <http://www.objectaid.com/>

Seguir las instrucciones que aparecen en la siguiente página para realizar su instalación: <http://www.objectaid.com/installation>

## Gradle plugin

Seguir las siguientes instrucciones para instalar el plugin de Gradle, empleado para realizar la integración de Gradle .

* Seleccionar del menú Help-> Eclipse Marketplace
* En el campo de búsqueda escribir ‘gradle’, presionar ‘go’.
* Instalar el plugin que aparece en la siguiente figura: *Buildship Gradle Integration X.x*



* Reiniciar la IDE para que los cambios tomen efecto.

Este documento estará en mejora continua conforme se detecten durante el proceso de desarrollo.