



Estándar de Codificación

Proyecto de Prácticas Profesionales

Elaborado por:
Martínez Caixba Miguel Ángel
Mongeote Tlachy Daniel
Torres Osorio Alesis de Jesús

14 de Diciembre del 2023

Universidad Veracruzana

Propósito	El propósito de establecer este estándar de codificación es promover prácticas de programación que resulten en un software seguro, fiable, verificable y de fácil mantenimiento.
Encabezado para los archivos FXML	<p>El encabezado se encuentra al principio del archivo, justo antes de las bibliotecas importadas y/o paquetes para proporcionar información sobre su creación.</p> <p>Esta deberá contener los siguientes puntos:</p> <ul style="list-style-type: none"> • Nombre del autor(es) • Fecha de creación • Última fecha de modificación • Descripción breve del propósito del archivo <p>Para el campo del nombre del autor, cada palabra se empezará con letra mayúscula y el resto en minúscula.</p> <p>Para el campo de descripción, la primera letra de la palabra que da inicio al texto se empezará con letra mayúscula y el resto en minúscula.</p> <p>Para el dato de fecha de creación y modificación, se deberá escribir en base al siguiente formato: dd/mm/aaaa.</p>
Ejemplo de Encabezado	<pre>/* *Autor: Mongeote Tlachy Daniel *Fecha de creación: 06/06/2023 *Fecha de modificación: 06/06/2023 *Descripción: Controlador de la vista para la creación de una experiencia educativa */</pre>
Nombrado	<p>El nombrado de los elementos que utilizamos al codificar es importante para poder hacer nuestro código más legible y fácil de entender.</p> <p>Se debe usar nombres descriptivos, significativos y relacionados al dominio de la solución para todas las variables, nombres de funciones, constantes y otros identificadores.</p> <p>El nombre de una variable, método o clase, etc. debería responder a todas las posibles preguntas.</p>
Nombrado de Clases	<p>Las Clases deben tener como nombre un sustantivo o una frase que haga alusión al mismo (persona, cosa, etc.) y empezar con letra mayúscula. Se debe evitar nombre con verbos.</p> <p>Se ocupará como formato el UpperCamelCase para el nombrado de clases.</p>
Ejemplo de nombrado de clases	<pre>private class Estudiante { ... } private class Director { ... } private class EdificioPrincipal { ... }</pre>

Nombrado de Componentes de Interfaz	de de	<p>Los componentes de una interfaz gráfica tales como botones, campos de texto, etiquetas, etc. deben de empezar con una abreviación del componente.</p> <p>Para abreviar un componente de una sola palabra, se deben quitar sus vocales y ocupar las tres primeras consonantes.</p> <p>Para abreviar un componente formado de dos palabras: de la primera palabra, se deben quitar sus vocales y ocupar las tres primeras consonantes, y juntar esto con la segunda palabra, empezando por mayúscula.</p> <p>Además, se ocupará como formato el lowerCamelCase.</p>
Ejemplo nombrado de componentes de interfaz	de de	<pre>private Button btnEliminarEstudiante; private Label lblFechaInicio; private TextField txtFieldBuscador; private MenuItem mnItem; private Estudiante usuarioEstudiante; private INotificacionActividad notificacion; private VBox vboxListaActividades; private ScrollPane scrPaneContenedorAnteproyectos; private Pane pnEstudiante; private ComboBox<Estudiante> cmbBoxEstudiante; private CheckBox chcBoxesCodirector;</pre>
Nombrado de método listeners	de	<p>Para declarar el nombre de los métodos <i>listeners</i> se debe empezar con la palabra “clic”, seguido de esto una palabra describiendo brevemente la acción que realiza.</p> <p>Se ocupará como formato el lowerCamelCase.</p>
Ejemplo nombrado de listeners	de de	<pre>private void clicIngresarSistema (ActionEvent event){..} private void clicBuscarAlumno (ActionEvent event){..} private void clicConfirmarSolicitud (ActionEvent event){..} private void clicDescargarDocumento (ActionEvent event){..}</pre>
Nombrado de Variables	de	<p>Las variables del programa deben tener como nombre un sustantivo que haga alusión a el mismo y empezar con una letra minúscula, se ocupara como formato el lowerCamelCase.</p>
Ejemplo		<pre>private double totalCuenta = 0.00; private int totalPersonas = 0;</pre>
Nombrado de Constantes	de	<p>Las constantes del programa deben tener como nombre un sustantivo que haga alusión a el mismo y se escribirán con mayúsculas, además serán nombradas con el formato SCREAMING_SNAKE_CASE.</p>
Ejemplo		<pre>public static final float GRAVEDAD_TIERRA = 9.81; private static final int TOTAL_JUGADORES = 2; private static final double PESO_MEXICANO = 18.50;</pre>
Nombrado de Métodos	de	<p>Los métodos de deben ser verbos o frases que hagan alusiones a verbos y se ocuparan prefijos en caso de ser necesario, además se utilizara el formato lowerCamelCase.</p>
Ejemplo		<pre>private void setName () { ... } private void getName () { ... } private boolean verificacion () { ... }</pre>

Comentarios	<p>Los comentarios en programación se utilizan para poner aclaraciones del código, y así es más fácil de entender lo que hace.</p> <ul style="list-style-type: none"> • Explica el por qué, no el cómo. • Documentar si existe algo inusual o inesperado dentro del código. • Evitar palabras altisonantes, chistes, bromas o cualquier comentario innecesario.
Buenos Comentarios	Un buen comentario describe el que hace cierta parte del código y no el cómo. Todos los comentarios deben ser claros y consistentes, utilizando mismo estilo.
Ejemplo	<pre>/* Esto es mucho más legible como comentario en bloque en medio de un montón de código */</pre>
Malos Comentarios	En un mal comentario se invierte mucho tiempo en generar cajas de comentarios “bonitas”, esto hace que se pierda mucho tiempo intentando acomodar cada caja cada vez que el comentario sea editado.
Ejemplo	<pre>***** * * * Este método suma dos números * * * *****</pre>
Indentación	<p>La indentación (o mejor conocido como <i>sangría</i>) es utilizado para una mejor lectura y comprensión del código. Esta debería seguir las siguientes características:</p> <ul style="list-style-type: none"> ▪ No hay un salto de línea antes de abrir una llave. ▪ Hay un salto de línea después de la llave de cierre, solo si esa llave termina una declaración o el cuerpo de un método, constructor o clase con nombre. Por ejemplo, no hay salto de línea después de la llave si va seguido de un <i>else</i> o “,”. ▪ Poner una sola instrucción por línea. ▪ Dejar un espacio entre cada uno de los elementos de una expresión. ▪ Agregar una tabulación en las instrucciones que están dentro de estructuras de control, métodos o bloques específicos de código.
Ejemplo	<pre>public class MiClase { public void method() { if(condition()) { try{ something(); }catch (ProblemException e) { recover(); } }else if (otherCondition()) { somethingElse(); }else { lastThing(); } } };</pre>

Capitalización	<p>lowerCamelCase: La primera letra no está en mayúsculas, pero las siguientes palabras ocupan una letra mayúscula para diferenciar unas palabras de otras.</p> <p>UpperCamelCase/Pascal Case: La primera letra está en mayúsculas y las primeras letras de las subsecuentes palabras se encuentran también en mayúsculas.</p> <p>SCREAMING_SNAKE_CASE: Las palabras se escriben en mayúsculas separadas por un “_”.</p>
Ejemplo	<p>lowerCamelCase:</p> <ul style="list-style-type: none"> • private double totalCuenta; <p>UpperCamelCase/Pascal Case:</p> <ul style="list-style-type: none"> • private class ProfesorMatematicas{...} <p>SCREAMING_SNAKE_CASE:</p> <ul style="list-style-type: none"> • private double PESO_MEXICANO = 18.50;