

## Estándar de codificación

Estándar de <b>prácticas de codificación</b>	
<b>Recomendaciones Generales.</b>	<ul style="list-style-type: none"> <li>- Escribir cada componente de software en español incluyendo comentarios.</li> <li>- Usar paquetes o diccionarios para mostrar la información al usuario final en el lenguaje nativo.</li> <li>- Las líneas en blanco o comentarios no cuentan como líneas de código.</li> <li>- Separar en archivos la parte de lógica de negocios con la capa de presentación.</li> </ul>
<b>Formato de cabecera</b>	<p>Describe como debe organizar la información de cabecera en cada archivo que se realice con el fin de que cada colaborador lo tenga en cuenta al momento de codificar.</p> <p><i>@nombre:</i> Nombre del componente.</p> <p><i>@objetivo:</i> Detalla brevemente el objetivo general del componente.</p> <p><i>@descripción:</i> Detalla brevemente lo que hace el componente.</p> <p><i>@colaborador:</i> Nombre del colaborador que realizó el componente.</p> <p><i>@zona:</i> Identificador de la zona a la cual pertenece.</p> <p><i>@fecha:</i> Fecha de entrega del componente.</p> <p><i>@versión:</i> Número de entrega correspondiente al componente. (Incrementos de manera decimal)</p>
<b>Ejemplo Formato de cabecera.</b>	<p><i>@nombre:</i> Componente para el registro de usuarios a la plataforma.</p> <p><i>@objetivo:</i> Desarrollar un componente apto para facilitar el registro de usuarios al sistema de notas de la Universidad de Nariño.</p> <p><i>@descripción:</i> Componente para el registro de usuarios al sistema sea docentes tiempo completo u hora catedra de la Universidad de Nariño, solicitando la información básica del usuario.</p> <p><i>@colaborador:</i> Nelson Suarez</p> <p><i>@zona:</i> PAS-03</p> <p><i>@fecha:</i> 10 - febrero 2017</p> <p><i>@versión.</i> 1.0</p>
<b>Organización de Variables.</b>	<p>Especifica la manera en cómo las variables que puedan ser usadas, debe ser declaradas siguiendo un orden en específico y además escritas una por cada sentencia.</p>
	<p>Ejemplo:</p> <p><b>Correcto:</b></p> <pre>int contador = 1; string saludo = "bye";</pre> <p><b>Incorrecto:</b></p> <pre>string saludo = "bye"; int contador = 1;</pre>
	<p>- Ubicación de variables primero las de clase public, luego las protected y finalmente las private.</p>

<b>Organización de Variables.</b>	- Organización de acuerdo al tipo de dato (int, float, double), tipo carácter, tipo texto, tipo booleano, arreglos o matrices.
	<p>Ejemplo:</p> <p><b>Correcto:</b></p> <pre>public int contador = 0; public string nombre = "Maria"; public boolean bandera = true; public string idEstudiante = '212';</pre> <p><b>Incorrecto:</b></p> <pre>private int suma = 0; private string teléfono = "8383";</pre>
	- Nombre de variables compuesta con el formato LowerCamelCase.
	<p>Ejemplo:</p> <p><b>Correcto de la Forma LowerCamelCase:</b></p> <pre>int contarLlegadas = 0;</pre> <p><b>Incorrecto</b></p> <pre>int countllegadas = 0;</pre>
	-No permita que el nombre de las variables sea la combinación de más de dos idiomas.
	<p>Ejemplo:</p> <p><b>Correcto:</b></p> <pre>int contarSalidas = 0;</pre> <p><b>Incorrecto:</b></p> <pre>int countSalidas = 0;</pre>
<b>Inicialización de Variables</b>	<p>Procure inicializar las variables que va a usar al momento de su declaración.</p> <p>Ejemplo:</p> <p><b>Correcto</b></p> <pre>public int dinero = 1000; private String[] profesion = {"Sistemas", "Administración", "Psicología"};</pre>
<b>Constantes</b>	<p>- Se sugiere que el nombre de las constantes se escriba en mayúscula.</p> <p>Ejemplo:</p> <p><b>Correcto:</b></p> <pre>int AREA_MAXIMA; int EDAD_MINIMA;</pre>

<b>Indentación</b>	<p>Propiciar que el código este ordenado en base a las estructuras que se vayan planteando. Se propone una Indentación común de 4 espacios equivalente al uso del tabulador.</p>
<b>Operadores</b>	<p>Permite establecer la declaración de operadores dentro del código fuente.</p> <p>Agregar espacios entre:</p> <ul style="list-style-type: none"> <li>- Operandos de declaración de variables.</li> </ul> <p>Ejemplo:</p> <pre>int edad = 5;</pre> <ul style="list-style-type: none"> <li>- Entre valores de operaciones aritméticas</li> </ul> <p>Ejemplo:</p> <pre>m = 5 + 2;</pre> <ul style="list-style-type: none"> <li>- Entre operadores lógicos o de comparación como &lt;=, &gt;=, &lt;, &gt;, !=, &amp;&amp;,   .</li> </ul> <p>Ejemplo:</p> <pre>if(variable &gt; 4 &amp;&amp; variable &lt; 10){     cout&lt;&lt;"Correcto"; }</pre> <p>Use signos de agrupación para separar los operadores lógicos cuando se usan más de uno en una misma instrucción.</p> <p>Ejemplo:</p> <pre>if((edad &gt; 15 &amp;&amp; edad &lt; 20)    (genero == 'F')){     alert("comprobado"); }</pre>
<b>Operadores</b>	
<b>Comentarios</b>	<ul style="list-style-type: none"> <li>- Los comentarios pueden ser de una línea y de múltiples líneas.</li> <li>- Después de un comentario aplique un salto de línea.</li> <li>- Los comentarios deben ir con lo estrictamente necesario.</li> <li>- Los comentarios deben mantener que no ocupen más de 80 caracteres por línea.</li> </ul>
<b>Funciones, métodos internos o clases</b>	<ul style="list-style-type: none"> <li>-Definir las funciones internas que pueda contener el componente software con nombres pertinentes a lo que se está realizando y que no superen los 30 caracteres.</li> <li>-Evitar colocar espacios entre el nombre de un método o función y el paréntesis de apertura de esta.</li> </ul> <p>Ejemplo:</p> <p><b>Correcto:</b></p> <pre>public void verEstudiantes()</pre> <ul style="list-style-type: none"> <li>- Definir cada método o función empleada con el formato lowerCamelCase.</li> </ul> <p>Ejemplo:</p> <p><b>Correcto:</b></p> <pre>void consultarRegistro()</pre> <ul style="list-style-type: none"> <li>- Coloque espacios en un método o función entre cada coma que separa un parámetro de otro.</li> </ul>

<b>Funciones, métodos internos o clases</b>	<p>Ejemplo:</p> <pre>void consultarNotas(a, b, c);</pre> <ul style="list-style-type: none"> <li>- Separe cada método, función o clase con una línea en blanco.</li> <li>- Las funciones o métodos deberán nombrarse de acuerdo al listado de palabras ofrecidas en el diccionario de datos de cada componente.</li> </ul>
<b>División de líneas.</b>	<p>Cuando una expresión no alcance por su tamaño en una sola línea esta podrá continuar en la siguiente siempre y cuando:</p> <ul style="list-style-type: none"> <li>- Exista una coma</li> <li>- Exista un operador.</li> </ul> <p>Ejemplo:</p> <pre>int recibirCliente(nombre, apellido, edad, teléfono);  if ((condicion1 &amp;&amp; condicion2)        (condicion3 &amp;&amp; condicion4)        (condicion5 &amp;&amp; condicion6)) {     recibirCliente('maria', 'gomez',                   31, '722881'); }</pre>
<b>Reutilización de código</b>	<p>Para evitar que exista código repetido dentro del desarrollo del componente, haga uso de la reutilización de código fuente mediante el uso de funciones, economice tiempo y trabajo doble.</p>

***Estándar para plataformas de modelado, entornos de programación y gestión de código fuente***

<b>Plantilla de entornos de programación</b>	
<b>Recomendaciones Generales.</b>	A continuación, se listan las herramientas que va usar para trabajar en el desarrollo de software del componente.
<b>Lenguajes de programación.</b>	Java
<b>Frameworks para su uso.</b>	Ninguno
<b>Entornos de desarrollo</b>	Netbeans
<b>Editor de texto</b>	Sublime Text
<b>Gestor de base de datos</b>	NA
<b>Gestor de librerías o dependencias.</b>	Github
<b>Modelador de Software</b>	StarUML

***Estándar para herramientas de comunicación y gestión de trabajo***

<b>Plantilla de herramientas de gestión de trabajo</b>	
<b>Recomendaciones Generales.</b>	A continuación, se listan las herramientas que va usar para la gestión de trabajo y comunicación en el proyecto de desarrollo de software.
<b>Plataformas para la comunicación.</b>	
<b>Descripción general</b>	Herramientas de comunicación que van a ser empleadas dentro del proyecto como parte fundamental para la interacción entre los participantes.
<b>Herramientas a usar.</b>	<b>Skype.</b>
	<b>Correo electrónico.</b>
<b>Plataforma para la administración de proyectos.</b>	
<b>Descripción general</b>	Orientada al seguimiento de actividades en línea, de cada una de las tareas que cada colaborador esta efectuado y el estado real de ellas.
<b>Plataforma a usar</b>	<b>Trello.</b>

### **Estándar para la documentación del Sistema**

<b>Plantilla para la documentación del Sistema</b>	
<b>Arquitectura</b>	Aplicación Java
<b>Fuente</b>	<b>Para líneas de código:</b> <b>Tipo de Fuente:</b> Consolas
<b>Modelado</b>	Cada diagrama, modelado o diseño que se haga del componente deberá tener un título y si se desea una descripción sencilla (no supere las 20 palabras).
<b>Funciones o métodos</b>	<p>Antes de iniciar una función, clase, método o procedimiento según sea el caso haga una breve descripción de lo que esta realiza.</p> <p>Ejemplo</p> <pre>/**  * Representa un producto de la máquina expendedora.  */ public class Producto { }</pre>
<b>Interfaces</b>	<p>Interfaz en Java</p> <ul style="list-style-type: none"> <li>- Utilice herramientas Java para la realización de la interfaz.</li> <li>- Utilice como apoyo el diseño propuesto</li> </ul>
<b>Glosario de Herramientas</b>	<p><b>Correo electrónico.</b> De carácter formal para el envío de las plantillas de funcionalidades, procesos y componente entre los participantes, así como la asignación de actividades, tutoriales y estándares establecidos.</p> <p><b>Github.</b> Es una plataforma para desarrollo de software colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora.</p> <p><b>Netbeans.</b> Es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a tecnologías como Java, PHP, Groovy, C/C++, HTML5, entre otras.</p> <p><b>StarUML.</b> Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture). Permite crear Diagrama de casos de uso, Diagrama de clase, de secuencia, de colaboración, de estados, de actividad, Diagrama de componentes, de despliegue y de composición estructural.</p> <p><b>Skype.</b> Empleada para la realización de reuniones grupales de manera virtual cara a cara, entre colaboradores y jefes de zona.</p> <p><b>Trello.</b> De uso diario para la administrar el flujo del proyecto en su totalidad y de los procesos asignados por zona para tener un control de los avances con mayor facilidad, organizado y midiendo su progreso.</p>

<b><i>Tutorial de Herramientas.</i></b>	<p>Trello. <a href="https://www.youtube.com/watch?v=tOpBJnOifAc">https://www.youtube.com/watch?v=tOpBJnOifAc</a></p> <p>Skype <a href="https://www.youtube.com/watch?v=4gjinviTZzew">https://www.youtube.com/watch?v=4gjinviTZzew</a></p> <p>Github <a href="https://www.youtube.com/watch?v=mlt1t5zDtbs">https://www.youtube.com/watch?v=mlt1t5zDtbs</a></p>
---	---