**Estándar de codificación**

|  |  |
| --- | --- |
| Propósito | Guiar el desarrollo de los programas de BPUS. |
| Estándar de conteo | * Contar cada línea física como un LOC. * No contar los comentarios ni los espacios en blanco. |
| Nombre de los componentes | * Los nombres de los componentes deben ser descriptivos. * Deben ir en lowerCamelCase. * No puede ser de un solo carácter. * Deben ser claramente diferenciables. * No deben llevar caracteres especiales. |
| Ejemplo de nombres de clase | Aa //incorrecto  hojadecalculo // incorrecto  HojaDeCálculo // incorrecto  HojaDeCalculo //correcto |
| Encabezado del programa | * Empezar todas las clases con un encabezado descriptivo. * Según el lenguaje así mismo va el símbolo de los comentarios. |
| Formato de encabezado  (TypeScript o CSS) | /\*  \* Proyecto: Nombre del proyecto  \* Componente: Nombre del componente  \* Desarrollador: Nombre del desarrollador(es)  \* Descripción: Descripción corta del función y propósito  \* Última modificación: Fecha de la última modificación  \*/ |
| Formato de encabezado  (HTML) | <!--  - Proyecto: Nombre del proyecto  - Componente: Nombre del componente  - Desarrollador: Nombre del desarrollador(es)  - Descripción: Descripción corta del función y propósito  -Última modificación: Fecha de la última modificación  -- > |
| Importaciones | * Todas las importaciones van debajo del encabezado dejando un salto de línea. * Cada importación va en un LOC diferente. |
| nombres de variables y métodos | * Los nombres deben ser descriptivos. * Los nombres deben ir en lowerCamelCase * Los nombres deben ser claramente diferenciables. * Los nombres no deben llevar caracteres especiales. * Los nombres no pueden ser de un solo carácter. * Los nombres no poder ser iguales a las palabras reservadas. |
| Variables estáticas  y atributos de clase | * Si hay variables estáticas se deben declarar antes que los atributos de clase. * Cada variable o atributo debe llevar su respectivo modificador de acceso (public, private, protected). * Cada variable va en un LOC diferente. |
| Uso de las llaves | * La llave de apertura van en el mismo LOC que el nombre o el control de flujo. * La llave de cierre va en un LOC individual para esta. * En la llave de cierre del do…while, en ese mismo LOC puede ir la expresión del while. * En la llave de cierre de un “if” se puede colocar en ese mismo LOC el “else” o el “else if”. |
| Ejemplo de uso de llaves | public void calcularPromedio(){  }  while(numeroDatos < 20){  }  do{  }while(numeroDatos<20);  if(promedio < 3){  }else{  } |
| Comentarios de uso | * Documente el código para que el lector pueda entender su funcionamiento. * Los comentarios deben explicar tanto el propósito como el comportamiento del código. * Antes de cada método agregar un comentario del funcionamiento de este. El comentario de los getters y setters no es necesario. * Comentar declaraciones de variables para indicar su finalidad. * Proporcionar advertencias de valores ilegales, condiciones de desbordamiento u otras condiciones que podrían resultar en una operación incorrecta. |
| Espacios en blanco | * En las importaciones no se debe dejar espacio. * En la declaración de las variables no se debe dejar espacio. * Antes de abrir un control de flujo debe haber un espacio. * Entre el final de las instrucciones y la llave de cierre no debe ir un espacio. Sin embargo, la llave va en un LOC aparte. * Entre la llave de cierre y las instrucciones siguientes debe haber un espacio. * Se debe dejar un espacio entre cada método. |
| Indentación | * Cada instrucción luego de una apertura de llaves debe llevar un TAB de indentación. * Cada instrucción dentro de una etiqueta de HTML debe llevar un TAB de indentación. * El cierre de la llave debe estar en el mismo nivel que el de apertura. * El cierre de la etiqueta debe estar en el mismo nivel que el de apertura. |
| Ejemplo de indentación | <div style="text-align:center">  <h1>  Welcome to {{title}}!  </h1>  </div>  @Component({  selector: 'app-root',  templateUrl: './app.component.html',  styleUrls: ['./app.component.css']  })  export class AppComponent {  title = 'MyfistApp';  } |