

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

Implementación de plataforma Crowdsourcing - Sustain Partners

**Documento de Estándares de Programación
Versión 1.0**

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

Historia de Revisión

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	03/12/2023	1.0	Versión Final.	AJX

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

Tabla de Contenidos

1.	OBJETIVO	4
2.	DECLARACIÓN DE VARIABLES	5
3.	Definición de Controles	6
3.1	Tipo de datos	6
3.2	Prefijo para el Control	6
3.3	Nombre descriptivo del Control	6
3.4	Declaración de variables, atributos y objetos	6
3.5	Declaración de clases	7
3.6	Declaración de métodos	7
4.	Clases.	8
5.	Métodos, Procedimientos y Funciones definidos por el Usuario.	8
6.	Beneficios	8
7.	Conclusiones	10

Estándares de Programación

1. OBJETIVO

Reglamentar la forma en que se implementará el código fuente del proyecto web y del proyecto de escritorio, pasando, por las variables, controles, clases, métodos, archivos y todo aquello que esté implicado en el código.

Mejorar y uniformizar a través de las reglas que se proponen, el estilo de programación que tiene cada programador.

- Los nombres de variables serán mnemotécnicos con lo que se podrá saber el tipo de dato de cada variable con sólo ver el nombre de la variable.
- Los nombres de variables serán sugestivos, de tal forma que se podrá saber el uso y finalidad de dicha variable o función fácilmente con solo ver el nombre de la variable.
- La decisión de poner un nombre a una variable o función será mecánica y automática, puesto que seguirá las reglas definidas por nuestro estándar.

Por tanto, se seguirán dichos patrones para un entendimiento legible del código y para facilitar el mantenimiento del mismo.

2. DECLARACION DE VARIABLES

JAVA SERVER PAGE Y API:

Se propone que la declaración de las variables, se ajusten al motivo para la que se requieran. El mnemotécnico definido se establece tomando en consideración principalmente lo siguiente:

- La longitud debe ser lo más recomendable posible. No debe ser tan grande de tal forma que el programador tenga la facilidad de manejo sobre la variable y ni tan corta que no pueda describirse claramente. Para el caso establecemos una longitud máxima de variable de 16 caracteres.
- El tipo de dato al que pertenece la variable.

Por lo tanto la estructura de la variable es como sigue:

Estructura	Descripción de la Variable
LONGITUD. MAX.	<input type="checkbox"/> 1 <input type="checkbox"/> <input type="checkbox"/> 16 <input type="checkbox"/>
FORMATO	<i>Minúscula la primera parte y luego la segunda con Mayúsculas</i>
EJEMPLO	isCaptchaValid

Siendo el nombre que identifica a la variable: **isCaptchaValid**

2.1 Descripción de la Variable.

Nombre que se le asignará a la variable para que se le identifique y deberá de estar asociada al motivo para la cual se le declara.

Ejemplos: pagoDAO, fkUser, monto

JAVA APPLICATION:

Se propone que la declaración de las variables, se ajusten al motivo para la que se requieran. El mnemotécnico definido se establece tomando en consideración principalmente lo siguiente:

- La longitud debe ser lo más recomendable posible. No debe ser tan grande de tal forma que el programador tenga la facilidad de manejo sobre la variable y ni tan corta que no pueda describirse claramente. Para el caso establecemos una longitud máxima de variable de 16 caracteres.
- El tipo de dato al que pertenece la variable.

Por lo tanto la estructura de la variable es como sigue:

Estructura	Descripción de la Variable
LONGITUD. MAX.	<input type="checkbox"/> 1 <input type="checkbox"/> <input type="checkbox"/> 16 <input type="checkbox"/>
FORMATO	<i>Minúscula la primera parte y luego la segunda con Mayúsculas</i>
EJEMPLO	nuevoEstado

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

Siendo el nombre que identifica a la variable: nuevoEstado

2.2 Descripción de la Variable.

Nombre que se le asignará a la variable para que se le identifique y deberá de estar asociada al motivo para la cual se le declara.

Ejemplos: columnaAux, rs, fila, nuevoEstado.

3. Definición de Controles

Para poder determinar el nombre de un control dentro de cualquier aplicación de tipo visual, se procede a identificar el tipo al cual pertenece y la función que cumple dentro de la aplicación.

3.1 Prefijo para el Control

El prefijo del control será determinado mediante tres caracteres que estarán conformados por las consonantes más representativas del control, es así, por ejemplo; el control Button, estará asociado al prefijo btn.

3.2 Nombre descriptivo del Control

Formado por la descripción de la función que lleva a cabo el control, esta debe ser descrita en forma específica y clara.

Tipo de control	Prefijo	Ejemplo
TextBox	txt	txtApellido
Button	btn	btnIngresar
Table	tbl	tblUsuarios
ComboBox	cmb	cmbOds

3.3 Declaración de variables, atributos y objetos

- Se debe declarar una variable por línea.

Título	Descripción
Sintaxis	[TipoVariable] [Nombre de la Variable]
Descripción	Todas las variables o atributos tendrán una longitud máxima de 30 caracteres. El nombre de la variable puede incluir más de un sustantivo los cuales se escribirán juntos. Si se tuvieran variables que puedan tomar nombres iguales, se le agrega un número asociado.
Observaciones	En la declaración de variables o atributos no se deberá utilizar caracteres como: <ul style="list-style-type: none"> Letra Ñ o ñ. Caracteres especiales ¡, ^, #, \$, %, &, /, (,), ¿, ‘, +, -, *, {, }, [,]. Caracteres tildados: á, é, í, ó, ú.
Ejemplo	boolean nuevoEstado Indica una variable o atributo que guardará un estado.

3.4 Declaración de clases

Título	Descripción
Sintaxis	[Tipo] [Clase] [Nombre de Clase]
Descripción	El nombre de las clases tendrá una longitud máxima de 30 caracteres y las primeras letras de todas las palabras estarán en minúsculas. Tipo se refiere a si la clase será: Private, Public o Protected. Nombre de Clase se refiere al mismo nombre de la clase, iniciado con las siglas cls en minúscula y luego la letra del paquete al que pertenece, si es Entidad (E) o Negocio (N) seguido del nombre de la clase.
Observaciones	En la declaración de clases no se deberá utilizar caracteres como: <ul style="list-style-type: none"> • Letra Ñ o ñ. • Caracteres especiales ¡, ^, #, \$, %, &, /, (,), ¿, ', +, -, *, {, }, [,], _. • Caracteres tildados: á, é, í, ó, ú.
Ejemplo	public class clsEcategoria. Indica la declaración de la clase Entidad categoría.

3.5 Declaración de métodos

Título	Descripción
Sintaxis	[alcance][tipo]nombreProcedim[(ListaParámetros)]
Descripción	El nombre del método consta hasta de 25 caracteres. La primera letra de la primera palabra del nombre será escrita en minúscula y las siguientes palabras empezarán con letra mayúscula.
Observaciones	En la declaración de métodos no se deberá utilizar caracteres como: <ul style="list-style-type: none"> • Letra Ñ o ñ. • Caracteres especiales ¡, ^, #, \$, %, &, /, (,), ¿, ', +, -, *, {, }, [,], _. • Caracteres tildados: á, é, í, ó, ú.
Ejemplo	private void MtdActualizarEstado(clsEusuario objEu) Indica una función que actualiza el campo estado del objeto usuario.

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

4. Clases.

JAVA SERVER PAGE Y API:

El estándar para nombres de clases es el PascalCase. Los nombre de las clases de la carpeta **Model** llevan el nombre de la tabla a la que instancian, quitando la tb del principio. Para las clases de la carpeta **DAO** se utilizará el nombre de la clase del modelo seguido de la palabra DAO en mayúscula.

- Ejemplos: Proyecto, Usuario. ProyectoDAO, UsuarioDAO.

El estándar de las clases conexion utiliza igualmente el PascalCase. Se iniciará con el nombre del tipo de conexion, seguido de la palabra **Connection**.

- Ejemplos: NeonConnection, RedisConnection.

El estándar de los objetos de las clases tienen el mismo nombre de las clases, sin embargo estas empiezan con minúscula.

- Ejemplos: proyecto, usuario, proyectoDAO, usuarioDAO.

JAVA APPLICATION:

El nombre de las clases debe ser auto descriptivo de manera que no se requiera, en lo posible, entrar al código de la función para saber qué es lo que realiza.

El estándar para nombres de clases es usar iniciar con las siglas **cls**, la cual debe estar escrita en minúscula seguido de una letra que identifique si es una clase del paquete **Entidad** o **Negocio**, luego finaliza con el nombre que identifica la clase en minúscula.

- Ejemplos: clsEmodalidad, clsEusuario, clsEmodalidad, clsNusuario.

El estándar de los objetos de las clases empiezan con las siglas **obj**, seguidas de la letra representativa de Negocio (N) O Entidad (E), seguida de la primera letra del nombre de la entidad o negocio que instancia.

- Ejemplos: objEm, objEu, objNm, objNu.

5. Métodos, Procedimientos y Funciones definidos por el Usuario.

JAVA SERVER PAGE Y API:

El nombre de las funciones y procedimientos debe ser auto descriptivo de manera que no se requiera, en lo posible, entrar al código de la función para saber qué es lo que realiza.
verbo-Sustantivo

El estándar para nombres de procedimiento es usar un Verbo que describa la acción realizada seguida por un sustantivo (objeto sobre el cual actúa). Se recomienda:

- Usar un nombre que represente una acción y un objeto. El nombre del procedimiento debe indicar qué hace el procedimiento a... o qué hace el procedimiento con....
- El verbo debe estar en infinitivo.

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

- Ser consistente en el orden de las palabras. Si se va a usar **verboNombre**, siempre usar **verboNombre**.
- Ser consistente en los verbos y sustantivos usados. Por ejemplo, si tiene un procedimiento **asignarNombre**, en vez de **colocarNombre**.
- Para la acción **enviar boleta al usuarios** se define:
enviarBoleta
Verbo: enviar
Sustantivo: Boleta

Nota:

- No se hará uso de los caracteres: Espacio en blanco " ", Caracter de subrayado "_".

JAVA APPLICATION:

El nombre de las funciones y procedimientos debe ser auto descriptivo de manera que no se requiera, en lo posible, entrar al código de la función para saber qué es lo que realiza.

El estándar para nombres de procedimiento es usar las siglas Mtd, seguidas de un Verbo que describa la acción realizada seguida por un Sustantivo (objeto sobre el cual actúa). Se recomienda:

- Usar un nombre que represente una acción y un objeto. El nombre del procedimiento debe indicar qué hace el procedimiento a... o qué hace el procedimiento con....
- El verbo debe estar en infinitivo.
- Ser consistente en el orden de las palabras. Si se va a usar **MtdVerboNombre**, siempre usar **MtdVerboNombre**.
- Ser consistente en los verbos y sustantivos usados. Por ejemplo, si tiene un procedimiento **MtdListarProyecto**, en vez de **MtdMostrarProyecto**.
- Para la acción **listar proyectos** se define:
MtdListarProyecto
Verbo: Listar
Sustantivo: Proyecto

Nota:

- No se hará uso de los caracteres: Espacio en blanco " ", Caracter de subrayado "_".
- En caso se tenga que especificar más a detalle, se colocará un sustantivo más.

Por ejemplo: **MtdGenerarGraficoParticipante**

6. Beneficios

- La documentación hace más legible un programa.
- Al documentar bien un programa desde un principio se evita que para cada modificación deba estudiarse profundamente el funcionamiento del programa, redescubriendo todo lo no documentado, con la ventaja adicional de que generalmente quién modifica el programa no es siempre quién lo escribió.
- Facilita la reutilización de módulos y rutinas desde cualquier otro programa o el mismo.
- Ayuda a determinar cuándo debe ser reescrito un código. Si existen problemas para explicar el código con un comentario, probablemente el código esté mal escrito.

Implementación de plataforma Crowdsourcing - Sustain Partners	Versión: 1.0
Documento de Estándares de Programación	

7. Conclusiones

- Una buena programación e implementación legible, sólo se logra usando y llevando de la mano un buen estándar o patrón de programación.
- Es muy importante que el programador tenga un conocimiento previo del estándar o en su defecto que lea el documento para prever diferencias.
- Al documentar se obtienen dos cosas fundamentales, un documento legible y segundo una buena base para los futuros desarrollos de mantenimiento del código.