Universidad Veracruzana

Facultad de Estadística e Informática

AssignU

ESTÁNDAR DE CODIFICACIÓN

Integrantes del equipo:

Saúl Barragán Torres

Juan David Carmona Ávila

Experiencias Educativas:

Desarrollo de Sistemas en Red

Desarrollo de Aplicaciones

Fecha de entrega:

Sin definir

Contenido

[Propósito y alcance 3](#_Toc198644741)

[Convención de nombres 4](#_Toc198644742)

[Clases 4](#_Toc198644743)

[Métodos/Funciones 4](#_Toc198644744)

[Variables locales 4](#_Toc198644745)

[Contantes 4](#_Toc198644746)

[Interfaces 4](#_Toc198644747)

[Archivos 5](#_Toc198644748)

[Propiedades 5](#_Toc198644749)

[Estilo de código 6](#_Toc198644750)

[Condicionales 6](#_Toc198644751)

[Bloques de control 6](#_Toc198644752)

# Propósito y alcance

Para garantizar que el código del sistema AssignU sea de calidad y legible, en este documento se definen las reglas y convenciones de codificación aplicables a:

* Back-end: ASP.NET Core (.NET 9+) y Node.js 2+
* Bases de datos MySQL / MongoDB
* Arquitectura: Microservicios y RESTful.

# Convención de nombres

## Clases

|  |  |
| --- | --- |
|  | **C#** |
| **Regla** | PascalCase |
| **Ejemplo correcto** | public class ClaseEjemplo |
| **Ejemplo incorrecto** | * public class claseEjemplo * public class clase\_ejemplo |

## Métodos/Funciones

|  |  |  |
| --- | --- | --- |
|  | **C#** | **JavaScript** |
| **Regla** | PascalCase | camelCase |
| **Ejemplo correcto** | public void MetodoEjemplo(int x, int y) | conts ejemploMetodo = (x, y) => |
| **Ejemplo incorecto** | * public void metodoEjemplo() * public void método\_ejemplo() | * const EjemploMetodo = () => * cont ejemplo\_metodo = () => |

## Variables locales

|  |  |  |
| --- | --- | --- |
|  | **C#** | **JavaScript** |
| **Regla** | camelCase | camelCase |
| **Ejemplo correcto** | var ejemploVariable | var ejemploVariable |
| **Ejemplo incorecto** | * var EjemploVariable * var ejemplo\_variable | * var EjemploVariable * var ejemplo\_variable |

## Contantes

|  |  |  |
| --- | --- | --- |
|  | **C#** | **JavaScript** |
| **Regla** | ALL\_CAPS | camelCase |
| **Ejemplo correcto** | const EJEMPLO\_VARIABLE | const ejemploVariable |
| **Ejemplo incorecto** | * const ejemploVariable * const EjemploVariable * const ejemplo\_variable | * const ejemploVariable * const EjemploVariable * const ejemplo\_variable |

## Interfaces

|  |  |
| --- | --- |
|  | **C#** |
| **Regla** | Prefijo I + PascalCase |
| **Ejemplo correcto** | public interface IInterfazEjemplo |
| **Ejemplo incorrecto** | * public class iInterfazEjemplo * public class i\_interfaz\_ejemplo |

## Archivos

|  |  |  |
| --- | --- | --- |
|  | **C#** | **JavaScript** |
| **Regla** | PascalCase | kebab-case |
| **Ejemplo correcto** | EjemploArchivo.cs | ejemplo-archivo.js |
| **Ejemplo incorecto** | * ejemploArchivo.cs * ejemplo\_archivo.cs | * EjemploArchivo.js * ejemploArchivo.js |

## Propiedades

|  |  |
| --- | --- |
|  | **C#** |
| **Regla** | PascalCase |
| **Ejemplo correcto** | var PropiedadEjemplo |
| **Ejemplo incorrecto** | * public class iInterfazEjemplo * public class i\_interfaz\_ejemplo |

# Estilo de código

|  |  |
| --- | --- |
| **C#** | **JavaScript** |
| Sangría de 4 espacios | Sangría de 2 espacios |
| Se deja un espacio entre la función y el paréntesis | |
| Luego de los paréntesis, la llave que abre se coloca en el siguiente renglón. | La llave que abre se coloca en el mismo renglón que la cabecera de la función. |
| Dentro del paréntesis, se deja un espacio entre una variable y un operador. | |
| Primera línea del código se escribe en el siguiente renglón después de la llave. | |
| Llave que cierra se coloca después de la última línea de código. | |

## Condicionales

|  |  |  |
| --- | --- | --- |
|  | **C#** | **JavaScript** |
| **AND** | && | && |
| **OR** | || | || |

## Bloques de control

|  |  |  |
| --- | --- | --- |
|  | **C#** | **JavaScript** |
| **If** | if (día = 10)  {  //*código*  } | if (mes != null) {  *//código*  } |
| **for** | for (int i = 0; i < PALABRAS; i++)  {  *//código*  } | for (int i = 0; i > nivel; i--) {  //*código*  } |
| **while** | while (nombreUsuario == ‘Test’)  {  //*código*  } | while (error) {  //*código*  } |
| **do** | do  {  //*código*  } while (numeroTareas > minimo) | do {  *//código*  } while (peso != TOTAL\_KILOS) |
| **switch** | switch (opcion)  {  case 1:  *//código*  break;  default:  *//código*  break;  } | switch (opcion) {  case 1:  *//código*  break;  default:  //código  break;  } |
| **try** | try  {  //*código*  } catch (Exception ex)  {  //*código*  } | try {  //código  } catch (error) {  //*código*  } |

|  |  |
| --- | --- |
| **Sangría** | 4 espacios. |
| **Uso de llaves** | Un renglón después de la cabecera del método, función o bloque de control. |
| **if** |  |
| **for** |  |
| **while** | While (nombreUsuario == ) |