Terminal Terrestre

Documento de Estandarización de Código

Versión 1.0

HISTORIAL DE REVISIONES

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 02/05/07 | 1.0 | Creación de documento | EGP, MAC |
|  |  |  |  |

**ESTANDARIZACION DE CODIFICACION**

1. **Creación de Librería de Clases**

TerminalTerrestre.AppUser

TerminalTerrestre.Public

1. **Creación de Clases**

App.cs

Contador.cs

1. **Organización de la Clase**
   * **Encabezado de la Clase**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** Clase: App.cs \*

*\** Espacio de nombres: TerminalTerrestre.Render \*

*\** Autor: Edilson Garcia \*

\*------------------------------------------------------------------------------------------------------------\*

\* Descripción: El render es quien enviara los cambios \*

\* que iremos implementado y los enviara por el \*

\* nodo raíz \*

\*------------------------------------------------------------------------------------------------------------\*

\* Fecha Autor Razón \*

\* 20/12/20 Edilson Garcia Creación \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

* + **Sentencias using**

*If..else*

*Try… catch*

*breack*

* + **Nombre de Espacio o Namespace**

Namespace TerminalTerrestre.Render

* + **Denotación de la Clase**

Class App

* + **Constantes**
  + **Declaración de variables**

**Const user**

**Let state**

* + **Métodos**

**Limpiar**

**Agregar**

**Borrar**

**Actualizar**

1. **Especificaciones de las Clases**
   * **Constantes**

Const CONTADOR=0

* + **Variables**

let

Las variables temporales se las denota con caracteres.

*Ejemplo:*

let i, j;

Las variables de instancia deben ser declaradas con el mismo nombre de la clase en minúsculas.

*Ejemplo:*

App app;

* + **Métodos**

render () {

try {

return ( )

}

catch( error) {

console.log(error);

}

}

1. **Sentencias**
   * **Sentencias compuestas**

try

{

for(..;..;..)

{

…….

}

}

catch( error)

{

console.log(error);

}

* + **Sentencias de retorno**

return ();

1. **Comentarios**

Cada método deber tener su respectivo comentario

* + **Comentarios de Línea**

*Ejemplo:*

// Este es un comentario de una sola línea

* + **Comentarios de Bloque**

*Ejemplo:*

/\*

Este es un comentario

de varias líneas

\*/

1. **Regiones**

Permite especificar un bloque de código que se puede expandir o contraer cuando se utiliza la característica de esquematización del editor de código de Visual Studio.

Ejemplo:

*var ticket = {*

*ruta: 'Quito-Guayaquil',*

*codigo: '41G78SW1',*

*compra: true,*

*fecha: '12-25-20',*

*hora: '12:00 a.m'*

*}*

*var contador = 0*

*while (ticket.compra == true) {*

*contador += 1*

*console.log( `Ticket: ${contador}.` )*

*ticket.compra = false*

*}*

*console.log( `El ticket de ${ticket.ruta} con codigo ${ticket.codigo} , ${ticket.fecha} ` )*