**Proyecto Restaurantes**

**Universidad de Caldas**

**Registro Histórico de Cambios en el Documento**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Código** | **Fecha** | **Elaboró** | **Revisó** | **Descripción Cambio** | **Versión** |
| **D-VIS-IS-UC** | **26/04/2021** | ***José Daniel Cruz Ávila*** |  | **Documento Inicial** | **0.1** |
| **D-VIS-IS-UC** | **28/04/2021** | ***José Daniel Cruz Ávila*** |  | **Se completa documento** | **0.2** |

***Informe área desarrollo***

# Estándares

* **Implementación**

Para el Back-End de la aplicación usaremos el leguaje .NET bajo los siguientes estándares de codificación:

* Generales:
  + Vida de variables corta (Mayoría de variables de método o locales en lugar globales).
  + Evitar procedimientos con más de un propósito (Si se tiene más de uno se debe dividir).
  + Evitar variables públicas.
  + Usar sentencias más optimas como Select-Case o Switch en lugar de sentencias if repetitivas.
  + Toda la codificación, nombres de variables y documentación se realizará en español.
  + No tener más de una clase en un mismo archivo.
* Nomenclatura
  + No usar caracteres especiales.
  + Los nombres deben ser lo más cortos y descriptivos posibles (Que definan lo que hace la variable o el método sin dar demasiado información).
  + Los namespaces llevaran la siguiente estructura:
    - BlastCode.Proyecto.Restaurante.[Funcionalidad]
  + El nombre de un archivo debe ser igual al de la clase.
  + Los nombres de clase deben estar en formato UpperCamelCase al igual que los nombres de los métodos y funciones.
  + Los métodos o funciones se nombraran siguiendo la estructura VerboSustantivo().
  + Las interfaces se nombrarán anteponiendo como prefijo la letra ‘I’ mayúscula y seguido de un nombre descriptivo y corto en UpperCamelCase así:
    - IEjemplo.
  + Las enumeraciones se nombrarán igual que las interfaces cambiando el prefijo por la letra ‘E’ mayúscula así:
    - EEjemplo.
  + Los nombres de las variables no deben llevar el nombre de la clase.
  + Las variables deben escribirse en formato camelCase lo más descriptivo y corto posible evitando abreviaturas.
    - Correcto: tablaMesa
    - Incorrecto: tablaSobreMesas, tabla\_mesa, tblMesa
  + **Bases de datos:**
    - Las tablas se definen en UpperCamelCase y en singular:
      * Correcto: Mesa
      * Incorrecto: Mesas, mesas, mesa
    - Los campos de las tablas no deben llevar el nombre de la tabla y estarán en formato UpperCamelCase.
* **Documentación**
  + Antes de cada definición de namespace debe ir una descripción de los tipos de clases que se agruparan en el namespace así:
    - /// <summary>

/// Comentario descriptivo del namespace

/// </summary>

* + Antes de cada definición de clase debe ir una descripción de la función principal o general de la clase además del autor de la clase y la versión así:
    - /// <summary>

/// Clase encargada de gestionar X

/// autor: Pepito Perez

/// version: 1.0

/// </summary>

* + - La versión aumenta en su parte decimal cada que se añade, modifica o elimina una nueva funcionalidad.
  + Antes de la definición de cada método o función debe añadir la descripción, los parámetros (si los hay) y los retornos (si los hay). Adicionalmente se deben añadir excepciones (Si las hay) y referencias a métodos relacionados (Si es necesario) de la siguiente manera:
    - /// <summary>

/// Desripción del método

/// </summary>

/// <param name=”NombreParametro”>

/// Descripción del parámetro.

/// </param>

/// <returns>

/// Descripción del retorno

/// </returns>

/// <exception cref=”Referencia a la excepción”>

/// Descripción de la excepción

/// </exception>

/// See <see cref=”MetodoReferencia(int, int)”/> descripción referencia