Contenido

[1. Introducción 2](#_Toc203848286)

[2. Objetivos 2](#_Toc203848287)

[3. Funcionalidades 2](#_Toc203848288)

[3.1 StringManager 2](#_Toc203848289)

[3.2 DoCmd 2](#_Toc203848290)

[4. Parámetros de Entrada y Salida 2](#_Toc203848291)

[5. Público Objetivo 3](#_Toc203848292)

[6. Anexos 3](#_Toc203848293)

Especificación Funcional - Solutions.Utility.Manager

# 1. Introducción

El presente documento describe la especificación funcional del proyecto Solutions.Utility.Manager. Este componente forma parte de una librería de utilidades destinada a simplificar y estandarizar operaciones comunes dentro de las aplicaciones .NET. Provee funciones para la manipulación de cadenas, tratamiento de valores nulos y formateo de fechas, horas y números.

# 2. Objetivos

Brindar un conjunto de métodos estáticos reutilizables que ayuden a:  
- Manipular cadenas de texto (Left, Right, Mid).  
- Convertir valores nulos a valores predeterminados.  
- Validar tipos de datos (fechas, numéricos, arrays, DBNull).  
- Formatear fechas, horas y números de forma estandarizada.

# 3. Funcionalidades

## 3.1 StringManager

Esta clase contiene extensiones para la manipulación de cadenas de texto:  
- Left: retorna los primeros N caracteres.  
- Right: retorna los últimos N caracteres.  
- Mid(start, end): retorna un segmento desde una posición y longitud.  
- Mid(start): retorna desde una posición hasta el final.

## 3.2 DoCmd

Contiene utilitarios diversos para:  
- Conversión segura de valores nulos a tipos comunes (string, int, long, double, bool, DateTime).  
- Funciones de formateo de fechas (FormatDateString) y horas (FormatTimeString).  
- Validaciones de tipo: IsArray, IsDate, IsDBNull, IsNumeric.  
- Funciones similares a VB (IIf, Choose).  
- Formateo de números con o sin símbolo monetario.

# 4. Parámetros de Entrada y Salida

Cada método de la biblioteca recibe uno o más parámetros simples (string, object, DateTime, etc.) y retorna valores procesados del mismo tipo o su representación formateada.  
  
No se manejan excepciones complejas ni flujos de negocio, ya que la funcionalidad es utilitaria.

# 5. Público Objetivo

Desarrolladores de aplicaciones en .NET que requieran métodos simplificados para manipulación de datos comunes.

# 6. Anexos

Se recomienda utilizar estas funciones en proyectos donde se requiera estandarizar la validación y formateo de datos sin depender de bibliotecas externas.