19/04/2024

DOCUMENTACION CODIGO C#

Aplicativo Web To-do list

Raul Hoyos

rhoyosemp@gmail.com

**TABLA DE CONTENIDO**

[**1.** **OBJETIVO** 2](#_Toc164427907)

[**2.** **DOCUMENTACION** 2](#_Toc164427908)

[**2.1 Contexto** 2](#_Toc164427909)

[**2.2 Herramientas usadas** 2](#_Toc164427910)

[**2.3 Procesos** 3](#_Toc164427911)

[2.3.1 P1 Creación de Base de datos 3](#_Toc164427912)

[2.3.2 P2 Creación de Proyecto MVC con DevExpress 4](#_Toc164427913)

[2.3.3 P3 Iniciar control de versionado Git 4](#_Toc164427914)

[2.3.4 P4 Creación de clases o modelo de BD 5](#_Toc164427915)

[2.3.5 P5 Creación de controlador API 6](#_Toc164427916)

[2.3.6 P6 Creación de controlador principal 8](#_Toc164427917)

[2.3.7 P7 Creación de vistas Index y agregar 9](#_Toc164427918)

[2.3.8 P8 Creación de método agregarTarea 12](#_Toc164427919)

[2.3.9 P9 Creación de método eliminarTarea 13](#_Toc164427920)

[2.3.10 P10 Creación de método marcar 14](#_Toc164427921)

[2.3.11 P11 Creación de vista editar 15](#_Toc164427922)

[2.3.12 P12 Creación de método actualizarTarea 16](#_Toc164427923)

[Filtros 17](#_Toc164427924)

|  |
| --- |
| **OBJETIVO** |

Documentar todos los procesos y herramientas usados en la creación del aplicativo web To-do list.

|  |
| --- |
| **DOCUMENTACION** |

## **2.1 Contexto**

Una aplicación de lista de tareas (to-do list) utilizando tecnologías para el frontend y el backend. Se debe cumplir con los requerimientos de interfaz, funcionalidad y estilo. Además de llevar un control de versiones.

## **2.2 Herramientas usadas**

De acuerdo con las tecnologías por elección propia este aplicativo fue creado con las siguientes herramientas:

* Frontend:
  + Para la creación de la interfaz se usó el lenguaje CSHTML.
  + Para los estilos CSS se usó la biblioteca Bootstrap.
  + Para la vista o lista de tareas se usó el Framework DevExpress-DevExtreme MVC.
  + Para la vista de agregar y editar se hizo uso de una plantilla de formulario Bootstrap.
* Backend:
  + Controladores: Se creo 2 controladores. Un controlador para las vistas y el manejo de los datos, con las funciones de crear(agregar), editar(actualizar) y eliminar. Un controlador Api encargado de cargar los datos mediante sus propiedades DataSourceLoad a la vista principal.
  + Modelo o clase: Se creo una clase haciendo el modelado de la base de datos respectiva con sus correspondientes campos.
* Modelo:
  + La arquitectura del aplicativo está basada en el Modelo MVC (Modelo-Controlado-Vista).
* Base de datos:
  + Para almacenar las tareas se hizo uso de una base de datos creada en MS SQL Server modelo relacional.
* Manejo de versión:
  + Para el versionado del proyecto se uso la terminal Git Bash. Realizando diferentes comandos git para el estado de los ficheros, la adicción al “stage” y su respectivo “commit”. Además de trabajar en una rama separada del (main) o (master).
  + Uso de GitHub para el almacenamiento del repositorio.

## **2.3 Procesos**

### 2.3.1 P1 Creación de Base de datos

Creacion de la base de datos llamada “listadoDeTareas”, creación de la tabla “ListadoTareas” con los campos “IdTarea”, “Tarea” (hace referencia a un nombre), “DescripcionTarea”, “Estado” y “CreacionTarea” (hace referencia a la fecha de creación de la tarea).

***Código:***

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

#### Imagen P1 Código SQL

***Visualización:***

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

#### Imagen P1 Vista SQL

### 2.3.2 P2 Creación de Proyecto MVC con DevExpress

Creación del proyecto e instalación de los Framework necesarios.

***Visualización:***

Captura de pantalla de computadora

Descripción generada automáticamente

#### Imagen P2 Vista Proyecto MVC

### 2.3.3 P3 Iniciar control de versionado Git

Creación del archivo o fichero “.git”, añadir los nuevos ficheros al “stage”, añadir su respectivo “commit” y crear un tag para referenciar la versión y que esté totalmente funcional.

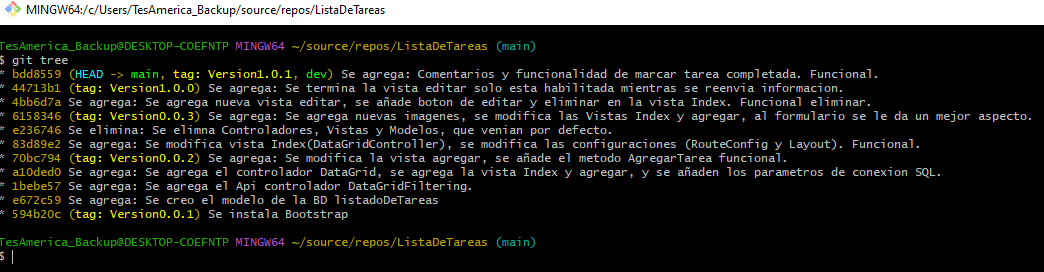
***Código:***

Texto

Descripción generada automáticamente

#### Imagen P3 Código. git

***Visualización:***



#### Imagen P3 Vista control de versión

### 2.3.4 P4 Creación de clases o modelo de BD

Modelado de la base de datos creada con SQL Server. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

#### Imagen P4 Código Models ListadoTareas

***Visualización:***

Pantalla de computadora con letras

Descripción generada automáticamente

#### Imagen P4 Vista Models

### 2.3.5 P5 Creación de controlador API

Creación del controlador API, con la conexión de la base de datos y su respectiva lista para el proceso de carga. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

#### Imagen P5 Código Controller

***Visualización:***

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

#### Imagen P5 Vista DataGridFilteringController API

### 2.3.6 P6 Creación de controlador principal

Creación del controlador principal, con conexión a la base de datos y su respectiva lista para el manejo de datos en lo métodos. Se hace uso del control de versión.

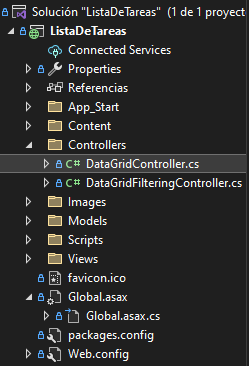
***Código:***

Texto

Descripción generada automáticamente

#### Imagen P6 Código Controller

***Visualización:***



#### Imagen P6 Vista DataGridController

### 2.3.7 P7 Creación de vistas Index y agregar

Creación de la vista principal con el listado de tareas y la vista agregar para insertar en la BD. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

#### Imagen P7 Código Vista Index

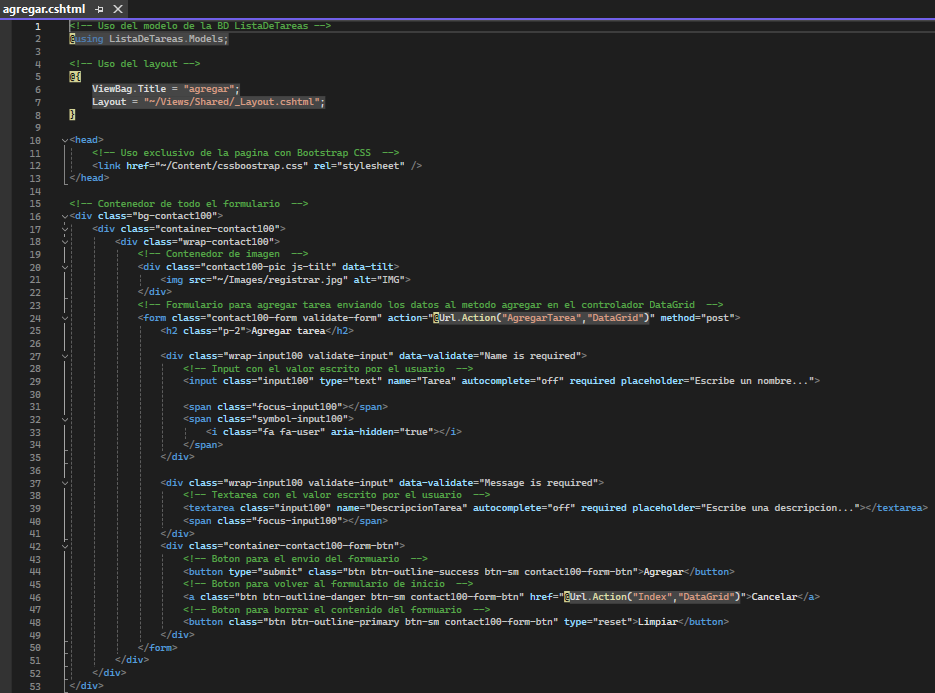
***Visualización:***

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Imagen P7 Vista Index

***Código:***



#### Imagen P7 Código Vista agregar

***Visualización:***

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

#### Imagen P7 Vista agregar

### 2.3.8 P8 Creación de método agregarTarea

Creación del método agregarTarea, con su respectiva conexión y envió de datos obtenidos desde el formulario. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

#### Imagen P8 Código método agregar

### 2.3.9 P9 Creación de método eliminarTarea

Creación del método eliminarTarea, con su respectiva conexión y envió de datos obtenidos desde el DataGrid. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

#### Imagen P9 Código método eliminar

***Visualización:***

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Imagen P9 Vista método eliminar

### 2.3.10 P10 Creación de método marcar

Creación del método marcar “Completado” o “Pendiente”, dependiendo del estado actual obtenido desde el DataGrid. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

#### Imagen P10 Código método marcar

***Visualización:***

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

#### Imagen P10 Vista método marcar

### 2.3.11 P11 Creación de vista editar

Creación de la vista editar, con los datos obtenidos desde el DataGrid, se redirecciona a la vista con estos datos. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

#### Imagen P11 Código vista editar

***Visualización:***

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Imagen P11 Vista editar

### 2.3.12 P12 Creación de método actualizarTarea

Creación del método actualizarTarea, con su respectiva conexión y envió de datos obtenidos desde el Formulario editar. Se hace uso del control de versión.

***Código:***

Texto

Descripción generada automáticamente

#### Imagen P12 Código método actualizar

***Visualización:***

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Imagen P12 Vista método actualizar

### Filtros

Filtros correspondientes al Framework de DevExpress MVC. Funcionales.

***Visualización:***

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Imagen Filtro Opcion1

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

#### Imagen Resultado Filtro Opcion1

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

#### Imagen Filtro Opcion2

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

#### Imagen Filtro Opcion3

|  |  |  |
| --- | --- | --- |
| Creado por: | Fecha de creación: | No.: |
| Raul Hoyos | 19/04/2024 | 1 |

Copyright © 2024. Todos los derechos reservados.