**Blazor Inventory Management System in .NET 8**

1. Architecture  
   Imagen que contiene Diagrama

   El contenido generado por IA puede ser incorrecto.
2. Comparision
   1. Traditional Web App Work (stateless application)  
      Diagrama

      El contenido generado por IA puede ser incorrecto.
   2. Interactivity with Javascript (traditional)  
      Diagrama

      El contenido generado por IA puede ser incorrecto.
   3. Interactivity with SignalR channel  
      Diagrama

      El contenido generado por IA puede ser incorrecto.
   4. Interactivity with WebAssembly  
      Diagrama

      El contenido generado por IA puede ser incorrecto.
3. Blazor Application
   1. Blazor Web App
      1. Front  
         Interfaz de usuario gráfica, Aplicación, Word

         El contenido generado por IA puede ser incorrecto.
      2. Cmd  
         Texto

         El contenido generado por IA puede ser incorrecto.  
         Console app -> program.cs
      3. Blazor workflow (internal)
         1. Program.cs -> App.razor -> Routes.razor -> MainLayout.razor
4. Inventory Management System
   1. Workflow  
      Diagrama

      El contenido generado por IA puede ser incorrecto.
   2. Architecture (Clean Arquitecture - Onion)  
      General Clean Arquitecture  
      Diagrama

      El contenido generado por IA puede ser incorrecto.
   3. View inventories Use Case
      1. Mockup  
         Tabla

         El contenido generado por IA puede ser incorrecto.  
         Interfaz de usuario gráfica, Aplicación

         El contenido generado por IA puede ser incorrecto.
      2. Description  
         As a user, I can view all the inventories as well as search certain inventories by names, so that Ican manage them.
      3. Main Scenario
         1. Pre Conditions
            1. The user is on the inventories page.
         2. Post Conditions
            1. The user sees the inventories that meet the search condition.
         3. Detailed Steps
            1. The user enters a few letters in the search box
            2. The user clicks on the View button.
      4. Alternative Scenario A
         1. Post Conditions
            1. The user sees all the inventories.
         2. Detailed Steps
            1. The user makes the search box empty.
   4. Development the use case
      1. Open Visual Studio
      2. Create class library Project
         1. Group the use case by Concept
      3. IMS.UseCases
         1. Add new folder -> Inventories -> new class -> ViewInventoriesByNameUseCase
      4. Create new class library Project IMS.CoreBusiness -> class Inventory
      5. IMS.UseCases -> dependencias -> IMS.CoreBusiness
      6. Data (Interface)
         1. IMS.UseCases -> New folder PluginInterfaces
         2. IMS.UseCases -> Inventories -> ViewInventoriesByNameUseCase -> ctor
         3. PluginInterfaces -> Add new Interface -> IInventoryRepository  
            Diagrama

            El contenido generado por IA puede ser incorrecto.
         4. Implement Repository -> créate solution folder IMS.Plugins -> créate folder IMS.Plugins in file browser
         5. IMS.Plugins -> Create class Project IMSPluginInMemory
         6. IMS.Plugins.InMemory -> GetInventoriesByNameAsync -> ctor -> List<Inventory> (fixed, then we Will use EF)
         7. IMS.CoreBusiness -> Inventory.cs -> créate properties  
            Texto

            El contenido generado por IA puede ser incorrecto.
         8. IMS.Plugins.InMemory -> Fill Inventory properties  
            Texto

            El contenido generado por IA puede ser incorrecto.
         9. IMS.Plugins.InMemory-> InventoryRepository -> GetInventoriesByNameAsync  
            Captura de pantalla de computadora

            El contenido generado por IA puede ser incorrecto.
   5. IMS.WebApp (implementation IInventoryRepository,InventoryRepository)
      1. Program.cs  
         
      2. IMS.UseCases->Inventories-> ViewInventoriesByNameUseCase -> Extract interface  
         Imagen que contiene Texto

         El contenido generado por IA puede ser incorrecto.
      3. IMS.UseCases->Inventories -> Create folder Interfaces -> Move IViewInventoriesByNameUseCase.cs to this folder
      4. Program.cs -> register mapping -> 