**Documentos de Referencia para el Desarrollo de la implementación de la User Story “Aceptar Cotización”**

**Backend: ASP.NET Core MVC**

Para la construcción del backend de la aplicación, seguimos la documentación oficial de **ASP.NET Core MVC**, la cual proporciona una base sólida para la estructuración del proyecto. En particular, nos enfocamos en los siguientes aspectos:

**1. Estructuración del Backend**

* Utilizamos la **información general de ASP.NET Core MVC** para estructurar el backend, aprovechando el patrón de diseño Model-View-Controller (MVC) para separar responsabilidades entre la lógica de negocio, la interfaz de usuario y el control de flujo de la aplicación.

**2. Inserción de Dependencias**

* Seguimos la documentación de **inserción de dependencias en ASP.NET Core**, lo que nos permitió implementar los servicios de manera desacoplada. Utilizamos el contenedor de inyección de dependencias proporcionado por el framework para gestionar la creación y el ciclo de vida de los servicios.
  + **Servicios**: Creamos servicios específicos para manejar la lógica de negocio, siguiendo las mejores prácticas de inyección de dependencias. Los servicios se registraron en el contenedor de servicios de ASP.NET Core particularmente mediante el metodo AddScoped()

**3. Controladores**

* Implementamos la lógica de control de solicitudes utilizando la documentación de **Control de solicitudes con controladores en ASP.NET Core MVC**. Los controladores se encargan de recibir y manejar las solicitudes HTTP, delegando la lógica de negocio a los servicios correspondientes y devolviendo las respuestas al cliente en los formatos adecuados (JSON).

Nos aseguramos de que estos controladores estén lo más desacoplados posibles y que no contengan el contexto de datos y la menor lógica de negocio posible utilizando los servicios que corresponden

**Frontend: Angular**

Para la parte del frontend, nos basamos en **Angular** y su **Style Guide oficial**. A continuación, detallamos los aspectos que seguimos para estructurar el frontend:

**1. Estructuración de la Aplicación**

* **Carpetas**: Organizamos el proyecto de manera que cada parte de la aplicación esté ubicada en su propia carpeta, facilitando la escalabilidad y el mantenimiento del código.
  + **Carpeta Services**: Aquí se ubicaron todos los servicios responsables de manejar la lógica de negocio del frontend, así como las llamadas HTTP a la API del backend.
  + **Carpeta Models**: En esta carpeta definimos todas las interfaces y tipos necesarios para tipar los datos utilizados en la aplicación.

**2. Inyección de Dependencias**

* Aplicamos el decorador @Injectable() en los servicios para permitir su inyección en los componentes y otros servicios, siguiendo las recomendaciones oficiales de Angular para manejar la inyección de dependencias.

**3. Modularización del Código**

* Organizamos la aplicación en **módulos por funcionalidad**. Por ejemplo, creamos módulos separados para manejar las funcionalidades de "Pedidos" y "Pagos", lo que permite mantener el código más limpio y modular.

# Bibliografía

## Backend:

* Información general de ASP.NET Core MVC: <https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-8.0>
* Inserción de dependencias en ASP.NET Core: <https://learn.microsoft.com/es-es/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-8.0>
* Controladores en ASP.NET Core MVC: <https://learn.microsoft.com/es-es/aspnet/core/mvc/controllers/actions?view=aspnetcore-8.0&source=recommendations>

## Frontend:

* Angular Style Guide (Oficial): <https://angular.dev/style-guide>