




# MANUAL Y LINEAMIENTOS PARA CREACIÓN DE PROYECTOS C# Y ANGULAR

---


Versión: 01

Unidad de Planeación de Infraestructura de Transporte  
Bogotá D.C., noviembre de 2024

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

## TABLA DE CONTENIDO

1. Introducción .....	3
2. ¿Para qué debo aplicar el documento? .....	3
3. ¿Cuál es la aplicación del documento? .....	3
4. ¿Qué conceptos debo tener claros para comprender el documento? .....	4
5. ¿Qué normatividad afecta el documento? .....	4
6. ¿Qué documentos externos requiero en la ejecución? (obligatorio) .....	5
7. ¿Qué documentos internos requiero en la ejecución .....	5
8. Desarrollo del documento (obligatorio) .....	5
8.1. Creación de proyectos para Backend con C# .....	5
8.2. Creación de proyectos para Frontend con Angular .....	12
9. ¿Qué cambios ha tenido el documento? .....	17

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
	<b>Fecha:</b> 01/11/2024	

## 1. Introducción

En la era moderna del desarrollo web, la combinación de tecnologías para Back-end y Front-end permite crear aplicaciones completas y optimizadas que cubren tanto el procesamiento de datos como la interacción de usuario. C# y Angular representan una sinergia poderosa en este contexto: C# ofrece un entorno robusto y seguro para el desarrollo de servicios de Back-end, mientras que Angular proporciona un framework estructurado y flexible para el desarrollo de interfaces de usuario dinámicas y reactivas.


Este documento presenta una guía integral para la creación de proyectos que integren el Back-end con C# y el Front-end con Angular, que son los lenguajes adoptados por el equipo de desarrollo de la UPIT, abarcando desde la configuración inicial de ambos entornos hasta la implementación de arquitectura limpia que facilite la escalabilidad y mantenibilidad de los mismos.

## 2. ¿Para qué debo aplicar el documento?

El documento está dirigido al personal técnico del proceso de Gestión de la Información, el grupo de GIT de tecnologías de la información y las comunicaciones cuyo objetivo es orientar al profesional – Desarrollador de aplicaciones, sobre el proceso para la construcción de proyectos de una manera organizada y estructurada manteniendo los lineamientos de la UPIT para la construcción de software, los cuales están alineados a la implementación de la arquitectura limpia tanto en el front-end como el back-end.

## 3. ¿Cuál es la aplicación del documento?

Este documento aplica para el proceso de construcción de proyectos en el front-end y el back-end usando arquitectura limpia, le indica al desarrollador las librerías básicas usadas para cumplir con los procesos de flujos de datos o CRUDS, si desea ampliar el conocimiento sobre mejores prácticas y lineamientos, debe revisar el documento que se encuentra en el repositorio de SharePoint, destinado a los desarrollos de sistemas de información, llamado "Documento de lineamientos y estándares para el desarrollo de sistemas de información en la UPIT".

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
	<b>Fecha:</b> 01/11/2024	

#### 4. ¿Qué conceptos debo tener claros para comprender el documento?

**Angular:** Es un framework de desarrollo web de código abierto, creado por Google, que permite construir aplicaciones web dinámicas y de una sola página (SPA, por sus siglas en inglés). Angular está basado en TypeScript, un superconjunto de JavaScript, y proporciona una arquitectura estructurada para desarrollar aplicaciones de gran escala.

**C#: (C-Sharp):** Es un lenguaje de programación moderno, orientado a objetos, desarrollado por Microsoft como parte de su plataforma .NET. Fue creado a principios de los años 2000 y está diseñado para ser simple, robusto y flexible, permitiendo el desarrollo de aplicaciones tanto de escritorio como web, móviles y servicios en la nube.

**IDE:** Un IDE (Integrated Development Environment) es un entorno de desarrollo integrado que proporciona un conjunto de herramientas diseñadas para facilitar la escritura, compilación, depuración y ejecución de código. Un IDE típicamente incluye:


- **Editor de código:** Un lugar donde los desarrolladores escriben su código.
- **Compilador o intérprete:** Para traducir el código a un formato que la máquina pueda entender y ejecutar.
- **Depurador:** Herramienta que permite detectar y corregir errores en el código.
- **Autocompletado y resaltado de sintaxis:** Facilita la escritura de código proporcionando sugerencias y destacando errores.
- **Integración con control de versiones:** Permite a los desarrolladores gestionar su código con herramientas como Git.

**Visual Studio Code:** Es un IDE o entorno de desarrollo integrado, normalmente usado para proyectos que usan JS en este caso Angular.

**Visual Studio:** Es un IDE o entorno de desarrollo integrado, normalmente usado para proyectos con C# o tecnologías Microsoft.

#### 5. ¿Qué normatividad afecta el documento?

La Normatividad que regula el presente manual o las citas normativas que se enuncian en las actividades, se encuentra definidas en el Normograma de la UPIT disponible para consulta en el siguiente link: <https://upit.gov.co/transparencia-y-acceso-a-la-informacion/>.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

## 6. ¿Qué documentos externos requiero en la ejecución? (obligatorio)

No aplica

## 7. ¿Qué documentos internos requiero en la ejecución

La documentación interna del equipo de desarrollo que hace parte de este documento y que se enuncian en las actividades, se encuentra almacenados en el repositorio de SharePoint, sitio de la Oficina de gestión de la Información, del siguiente link: [Manual y lineamientos para la construcción de software a la medida](#)

## 8. Desarrollo del documento (obligatorio)

### 8.1. Creación de proyectos para Backend con C#

Los pasos y lineamientos a tener en cuenta para la creación de proyectos C# tanto en el Back-end y Front-end, se relacionan a continuación:

8.1.1 Crear solución en blanco, en donde se agregará los diferentes módulos, al momento de crear este documento se usó la versión .Net Core 8.

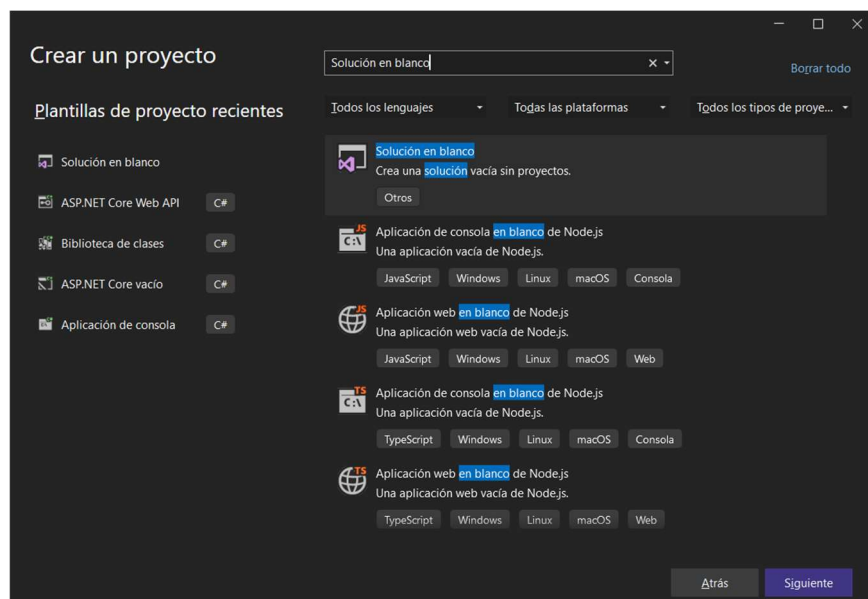



Ilustración 1 - GIT de Tecnología de la Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	<b>Manual y Lineamientos para Creación de Proyectos C# y Angular</b>	
	<b>Código:</b> M-GTI-012	<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

8.1.2 Iniciar la adición de proyectos, se crearán Bibliotecas de clases y como proyecto para la capa de presentación un API con OpenApi

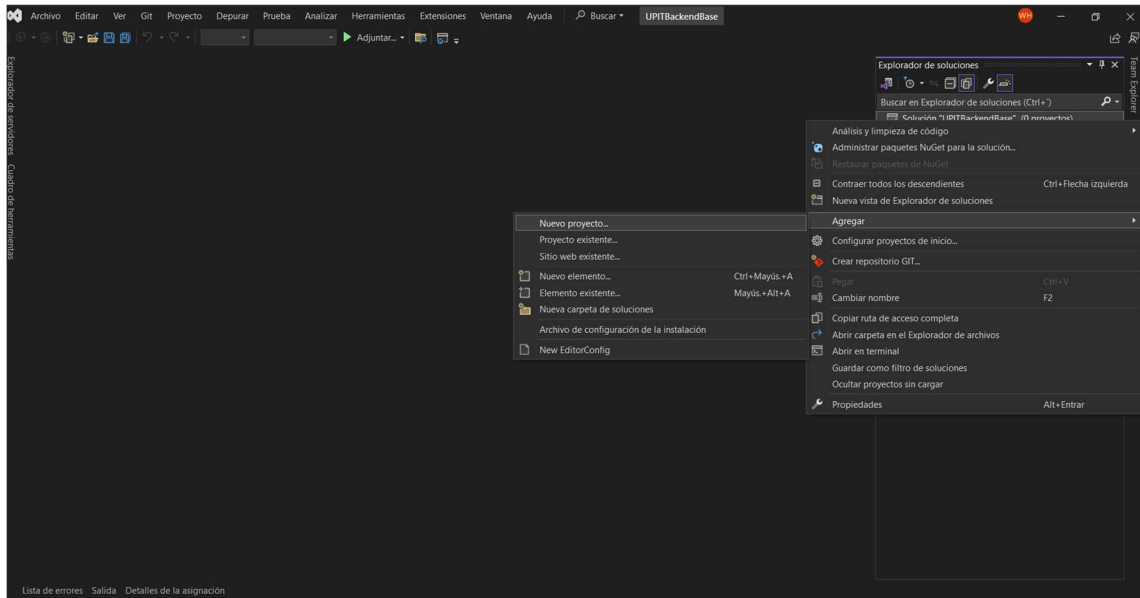


Ilustración 2 - GIT de Tecnología de la Información y las Comunicaciones

8.1.3 Crear bibliotecas de clases para las capas respectivas (Domain, Infraestructure, Application (Opcional))

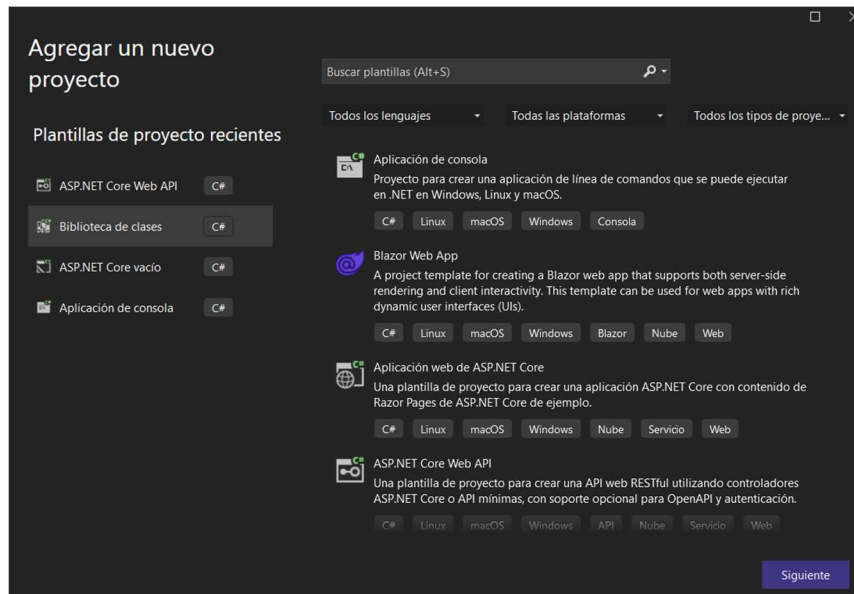

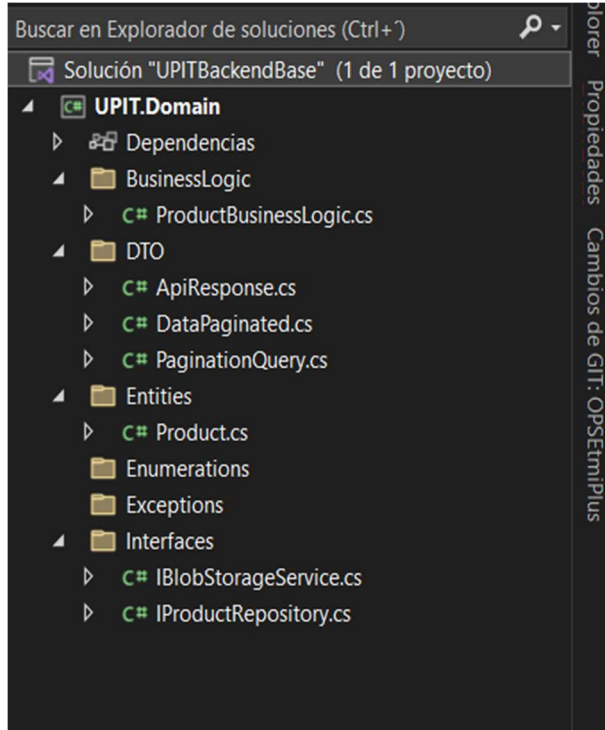


Ilustración 3 - GIT de Tecnología de la Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación validada la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

## Capa de Dominio



## Capa de Infraestructura

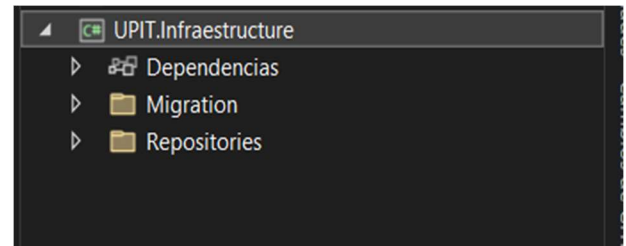


Ilustración 4 - GIT de Tecnología de la Información y las Comunicaciones

8.1.4 Por último, creamos la capa de presentación en esta capa se pueden colocar APIs, Aplicaciones de formulario como WPF, Aplicaciones de consola, Azure Functions, etc. Esto depende de nuestra necesidad, también es necesario colocar esta capa como proyecto de inicio.

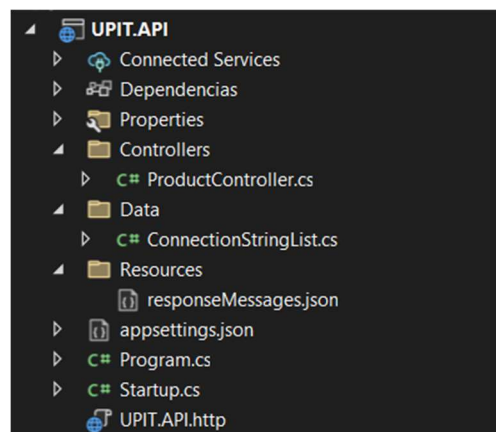

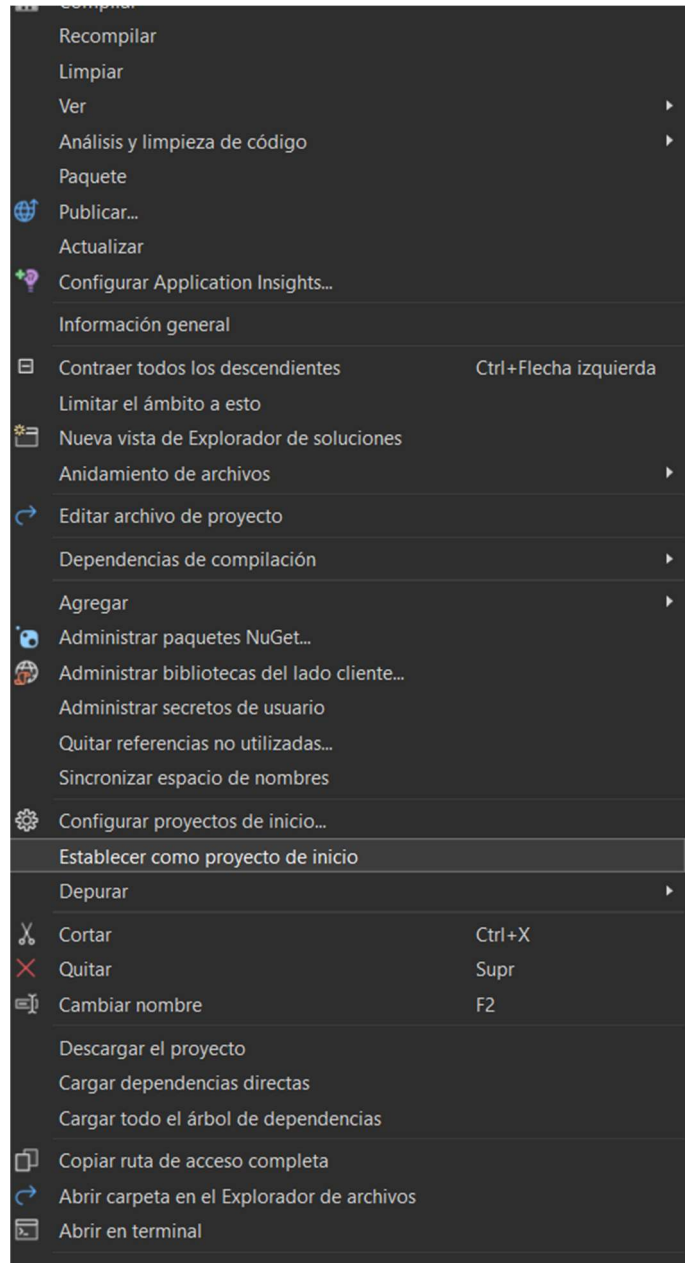


Ilustración 5 - GIT de Tecnología de la Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024


El resultado final del proyecto base debe quedar de la siguiente manera:



*Ilustración 6 - GIT de Tecnología de la Información y las Comunicaciones*

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.



	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

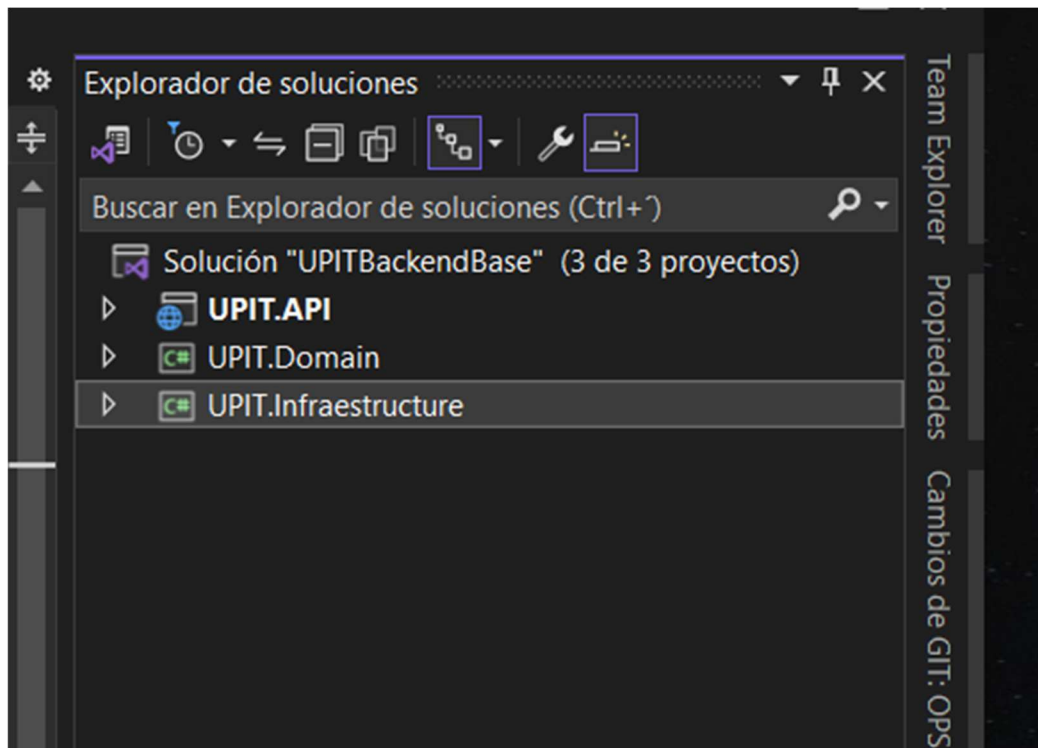



Ilustración 7 - GIT de Tecnología de la Información y las Comunicaciones

## Instalación de librerías básicas

Sobre la capa de Infraestructura damos click derecho Administrar paquetes Nuget, como esta capa es la encargada de manejar la persistencia colocamos los paquetes requeridos como Dapper y SqlClient para los casos generales.

	<b>Gestión de Tecnologías de la Información</b>	<b>Código:</b> M-GTI-012
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

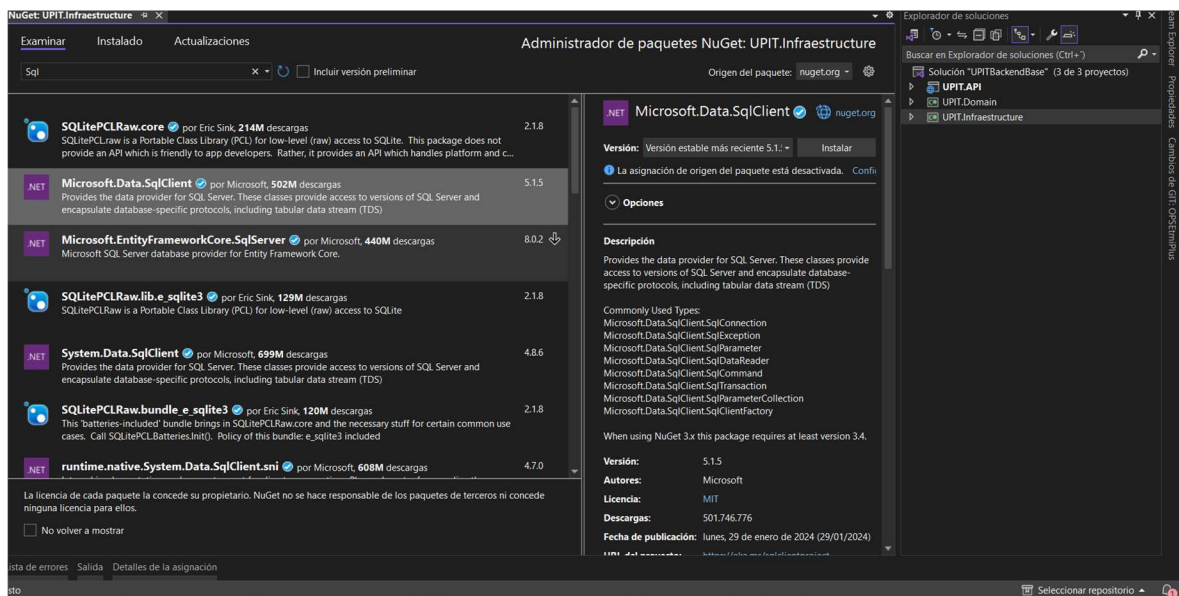
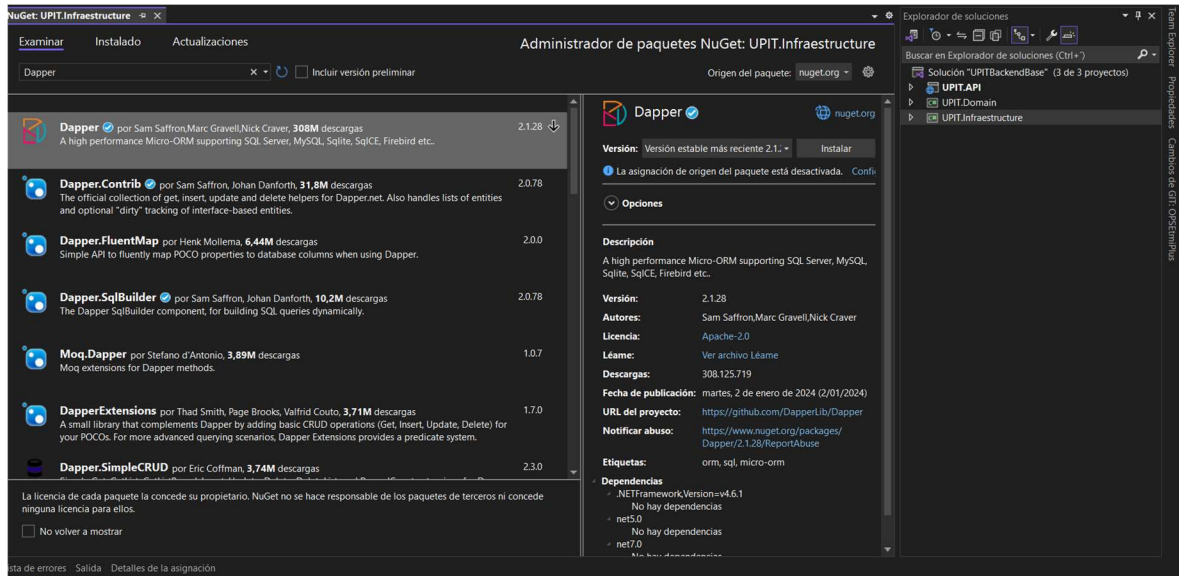



Ilustración 8 - GIT de Tecnología de la Información y las Comunicaciones

Para el manejo de la seguridad en los servicios web, instalamos las librerías necesarias para manejar la seguridad con Token (Microsoft.AspNetCore.Authentication.JwtBearer, System.IdentityModel.Tokens.Jwt)

La unidad de Planeación de Infraestructura de Transporte declara como única documentación validada la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

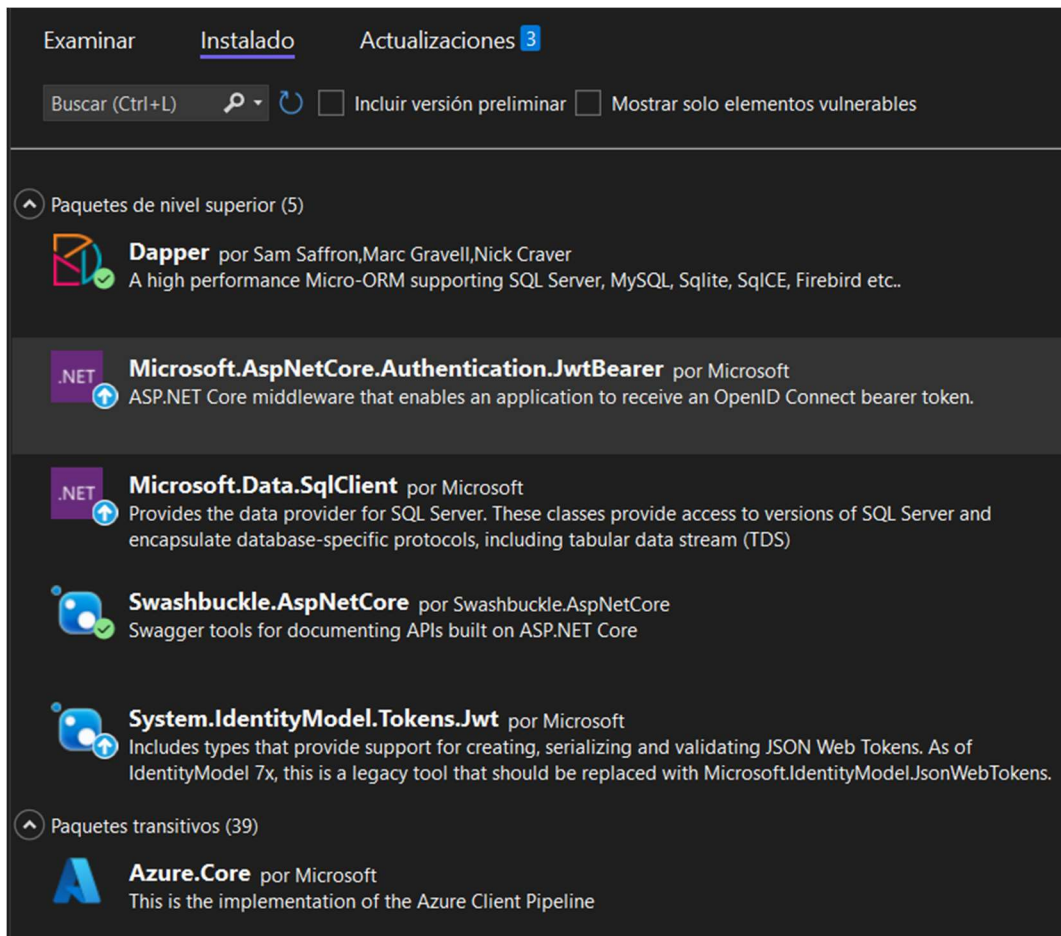



Ilustración 9 - GIT de Tecnología de la Información y las Comunicaciones

Con esta configuración ya tendríamos listo el proyecto base para trabajar con BD SqlServer y Dapper, además agregando seguridad a los servicios, a continuación, se identificarán que librerías se pueden usar para usar el almacenamiento del BlobStorage de Azure.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

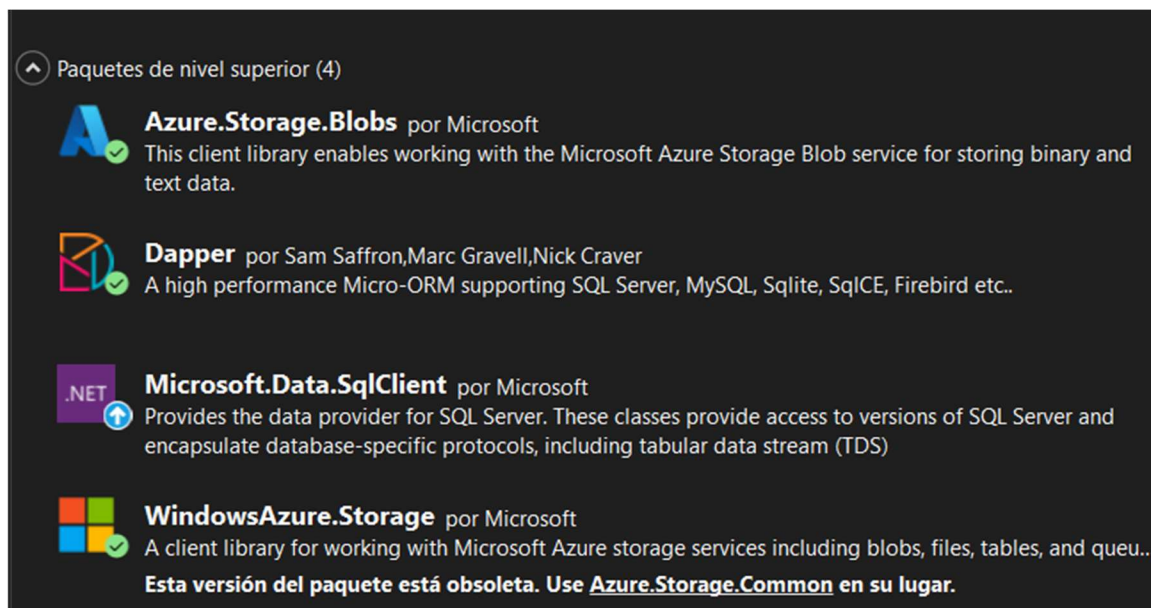


Ilustración 10 - GIT de Tecnología de la Información y las Comunicaciones

Los Nugets que se deben instalar son los siguientes: **Azure.Storage.Blobs**, **WindowsAzure.Storage**. estos Nugets se instalan en la capa de Infraestructura ya que es esta quien se encarga de acceder a repositorios.

## 8.2 Creación de proyectos para Frontend con Angular

8.2.1 Se debe ubicar en el directorio donde se quiere crear el proyecto y ejecutar el siguiente comando: `ng new {nombre_proyecto}`


```

C:\Users\FABIAN HUERFANO\Desktop\Backup\Documentos\Backup Lenovo Viejo\UPIT\Desarrollo>ng new UPITFrontendBase
? Which stylesheet format would you like to use?
  CSS
  > SCSS [ https://sass-lang.com/documentation/syntax#scss ]
  Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]

```

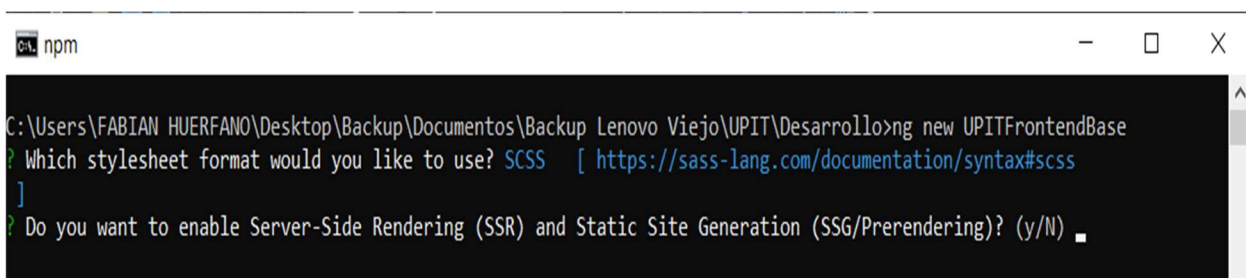
Ilustración 11 - GIT de Tecnología de la Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

Esta instrucción ejecutará unas tareas en la terminal y le hará algunas preguntas como que formato de estilos desea usar, en este caso vamos a seleccionar SCSS como se muestra en la imagen, ya que vamos usar un preprocesador de CSS pero usando indentación ya que de la otra manera es muy propenso a generar errores si no se tiene unas buenas prácticas de desarrollo y la idea de esta guía es ayudar al desarrollador.

8.2.2 La segunda pregunta es si desea hacer Server Side rendering (SSR) para hacer un prerrenderizado de la aplicación antes de visualizarse en el navegador esto es altamente útil si lo que queremos es mejorar la lectura de los robots de los navegadores, mejorando el SEO de la página, de lo contrario ponemos N (No)




```

C:\Users\FABIAN HUERFANO\Desktop\Backup\Documentos\Backup Lenovo Viejo\UPIT\Desarrollo>ng new UPITFrontendBase
? Which stylesheet format would you like to use? SCSS [ https://sass-lang.com/documentation/syntax#scss
]
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? (y/N)

```

Ilustración 12 - GIT de Tecnología de la Información y las Comunicaciones

8.2.3 A continuación, se creará el proyecto y se instalarán los paquetes requeridos para iniciar nuestra aplicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	
	<b>Código:</b> M-GTI-012	<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

```

C:\Windows\system32\cmd.exe
C:\Users\FABIAN HUERFANO\Desktop\Backup\Documentos\Backup Lenovo Viejo\UPIT\Desarrollo>ng new UPITFrontendBase
? Which stylesheet format would you like to use? SCSS [ https://sass-lang.com/documentation/syntax#scss
]
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? No
CREATE UPITFrontendBase/angular.json (2817 bytes)
CREATE UPITFrontendBase/package.json (1048 bytes)
CREATE UPITFrontendBase/README.md (1070 bytes)
CREATE UPITFrontendBase/tsconfig.json (909 bytes)
CREATE UPITFrontendBase/.editorconfig (274 bytes)
CREATE UPITFrontendBase/.gitignore (548 bytes)
CREATE UPITFrontendBase/tsconfig.app.json (263 bytes)
CREATE UPITFrontendBase/tsconfig.spec.json (273 bytes)
CREATE UPITFrontendBase/.vscode/extensions.json (130 bytes)
CREATE UPITFrontendBase/.vscode/launch.json (470 bytes)
CREATE UPITFrontendBase/.vscode/tasks.json (938 bytes)
CREATE UPITFrontendBase/src/main.ts (250 bytes)
CREATE UPITFrontendBase/src/favicon.ico (15086 bytes)
CREATE UPITFrontendBase/src/index.html (302 bytes)
CREATE UPITFrontendBase/src/styles.scss (80 bytes)
CREATE UPITFrontendBase/src/app/app.component.html (20887 bytes)
CREATE UPITFrontendBase/src/app/app.component.spec.ts (946 bytes)
CREATE UPITFrontendBase/src/app/app.component.ts (378 bytes)
CREATE UPITFrontendBase/src/app/app.component.scss (0 bytes)
CREATE UPITFrontendBase/src/app/app.config.ts (227 bytes)
CREATE UPITFrontendBase/src/app/app.routes.ts (77 bytes)
CREATE UPITFrontendBase/src/assets/.gitkeep (0 bytes)
✓ Packages installed successfully.
warning: LF will be replaced by CRLF in .editorconfig.
The file will have its original line endings in your working directory

```

Ilustración 13 - GIT de Tecnología de la Información y las Comunicaciones

8.2.4 Una vez tenemos el proyecto procederemos a instalar las dependencias para realizar las pantallas responsive y tener los componentes de angular material disponibles, ejecutamos las siguientes instrucciones:

**ng add @angular/material** (Librería que contiene los componentes de material tambien responsivos para crear aplicaciones más rápidas y más bonitas al usuario, manteniendo buenas prácticas de usabilidad)

**npm i ngx-flexible-layout --save** (Librería que nos da las directivas necesarias para hacer aplicaciones responsive)

```


C:\Users\FABIAN HUERFANO\Desktop\Backup\Documentos\Backup Lenovo Viejo\UPIT\Desarrollo\UPITFrontendBase>ng add @angular/material
Using package manager: npm
✓ Found compatible package version: @angular/material@17.2.1.
✓ Package information loaded.

The package @angular/material@17.2.1 will be installed and executed.
Would you like to proceed? Yes
✓ Packages successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo/Pink [ Preview: https://material.angular.io?theme=indigo-pink ]
? Set up global Angular Material typography styles? Yes
? Include the Angular animations module? Include and enable animations
UPDATE package.json (1114 bytes)
✓ Packages installed successfully.
UPDATE src/app/app.config.ts (1035 bytes)
UPDATE angular.json (2951 bytes)
UPDATE src/index.html (525 bytes)
UPDATE src/styles.scss (325 bytes)

```

Ilustración 14 - GIT de Tecnología de la Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación validada la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

Al instalar material nos preguntara por la fuente a usar, el tema y la habilitación del paquete de animation de Angular.

## Estructuración

Una vez configurado el proyecto inicial podemos iniciar con la estructuración de carpetas, igual que en el backend necesitamos usar las mismas capas para implementar arquitectura limpia, este es un ejemplo de la estructuración para un proyecto:

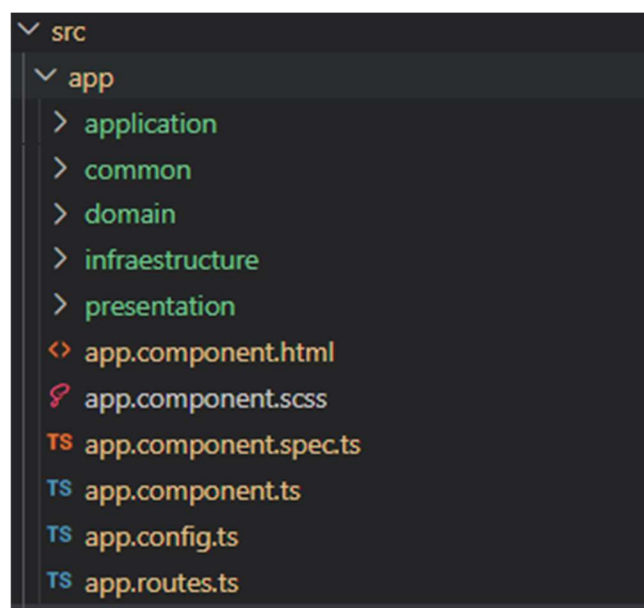



Ilustración 15 - GIT de Tecnología de la Información y las Comunicaciones

En el documento de arquitectura se encuentra una explicación más en profundidad sobre cada capa, esta estructura puede ser implementada si es una aplicación sencilla o si es una aplicación macro se debe considerar crear una estructura por dominio para mantener el orden y asegurar el mantenimiento y escalabilidad de la aplicación, es decir que cada dominio (subcarpeta) ej: Usuarios, Nomina., Deben contener la estructura anterior de capas como Application, Domain, Infraestructure y Presentation. La capa common pasaría a un nivel más alto la reutilización de código desde diferentes dominios.

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.

	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
	<b>Fecha:</b> 01/11/2024	

A continuación, se explica un poco la capa de presentación, esta carpeta está compuesta por un contenedor se encargará de hacer el routing interno del módulo, los demás componentes irán sobre la carpeta pages.

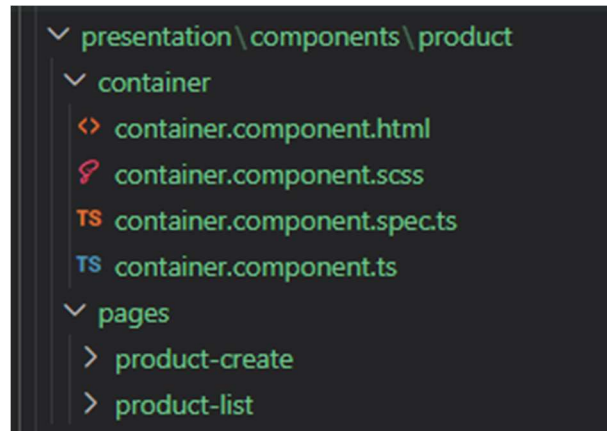


Ilustración 16 - GIT de Tecnología de la Información y las Comunicaciones

El routing es una parte fundamental de las aplicaciones por eso es indispensable hacer una buena configuración en este archivo, se recomienda el uso de Lazy Loading (carga perezosa) para la carga tanto de módulos como de componentes ya que reducirá la carga de proceso en el navegador, cargando solo lo necesario por demanda, haciendo las aplicaciones más fluidas.

```


//Lazy loading components
{
  path: 'products',
  loadComponent: () => import('./presentation/components/product/container/container.component'),
  children: [
    {
      path: 'list',
      title: 'List Products',
      loadComponent: () => import('./presentation/components/product/pages/product-list/product-list.component')
    },
    {
      path: 'create',
      title: 'Create Product',
      loadComponent: () => import('./presentation/components/product/pages/product-create/product-create.component')
    },
    {
      path: '',
      redirectTo: 'list',
      pathMatch: 'full'
    }
  ]
}

```

Ilustración 17 - GIT de Tecnología de la Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.



	<b>Gestión de Tecnologías de la Información</b>	
	Manual y Lineamientos para Creación de Proyectos C# y Angular	<b>Código:</b> M-GTI-012
		<b>Versión:</b> 001
		<b>Fecha:</b> 01/11/2024

## 9. ¿Qué cambios ha tenido el documento?

Versión Generada	Fecha	Descripción del Cambio o Modificación
1	01/11/2024	Documento Inicial

Elaboró	Revisó	Aprobó
Wilson Fabian Huerfano Hurtado  Contratista OGI -TI	Ricardo Diaz Cifuentes Contratista GIT de Planeación	Bismark Buenaños Coordinador del GIT de Tecnologías de Información y las Comunicaciones

La unidad de Planeación de Infraestructura de Transporte declara como única documentación válida la ubicada en el Banco de Documentos, y entra en vigencia a partir de la publicación.