

# Inyección de dependencias

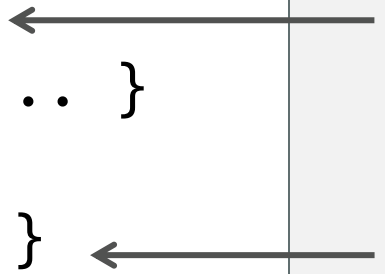
# Inyección de dependencias

El código específico de la plataforma "inyecte" la dependencia pasándola como un parámetro o configurando una propiedad.

```
public class DataAccessLayer
{
    public DataAccessLayer(
        IDbRespository db,
        IAlertService alerts) { ... }


    public ILogger Logger { get; set; }
    ...
}
```

Los servicios de los que depende esta clase deben proporcionarse ("inyectarse") a través de parámetros de constructor, propiedades o parámetros de método



# Contenedor Inversion of Control (IoC)

Un contenedor IoC es un **administrador de dependencias** que se utiliza para crear y controlar la vida útil de las dependencias en su aplicación; tiene dos propósitos:

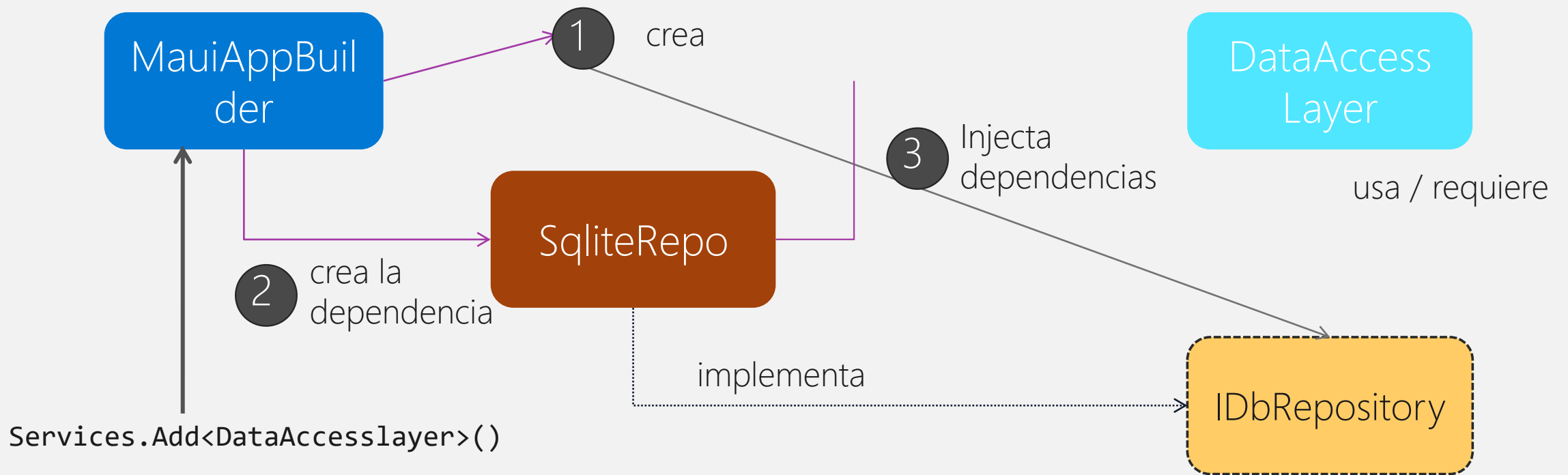


Registro de  
dependencias  
conocidas

Crea objetos e "inyecta"  
las dependencias  
requeridas

# Contenedor IoC

Puede automatizar DI con un contenedor en el que se registran las dependencias con las que luego se crean tipos, proporcionando automáticamente las dependencias



# Ejemplo

El MauiApp.CreateBuilder método crea un MauiAppBuilder objeto que podemos usar para registrar nuestras dependencias.

```
mauiAppBuilder.Services.AddSingleton<ViewModels.MainViewModel>();  
mauiAppBuilder.Services.AddSingleton<ViewModels.LoginViewModel>();
```

- **AddSingleton<T>** Creará una única instancia del objeto que se producirá durante la vigencia de la aplicación.
- **AddTransient<T>** Creará una nueva instancia del objeto cuando se solicite durante la resolución. Los objetos transitorios no tienen una duración predefinida, pero normalmente seguirán la duración de su host.

# Pros & Cons

## Pros

- El cliente solo necesita las **dependencias**, no se necesita una referencia de contenedor
- Más fácil de identificar las dependencias que se utilizan, ya que a menudo se pasan a los constructores o se completan las propiedades.

## Cons

- Implica un poco de magia (!), en general puede ser más difícil de entender (qué depende de qué)
- En determinados casos requiere alguna forma de reflexión; generalmente no es un problema de rendimiento, pero podría ser

# Conclusion

La inserción de dependencias permite **desacoplar** tipos concretos del código que depende de estos tipos. Normalmente usa un contenedor que contiene una lista de registros y asignaciones entre interfaces y tipos abstractos, y los tipos concretos que implementan o extienden estos tipos.

**Microsoft.Extensions.DependencyInjection** facilita la creación de un acoplamiento flexible y proporciona todas las características que se encuentran habitualmente en contenedores de inserción de dependencias, incluidos métodos para registrar asignaciones de tipos e instancias de objetos, resolver objetos, administrar la duración de los objetos e insertar objetos dependientes en constructores de objetos que resuelve.