

### Arquitectura de referencia: Capa de dominio en C#



# Hola:

La capa del dominio es el núcleo central de una aplicación, en ella no solamente se encuentran las entidades que identifican los conceptos principales del problema, sino que también alberga muchos de los principales algoritmos de la aplicación.

La trazabilidad en este nivel es la codificación del lenguaje de programación C# de la capa del dominio y de los componentes que se encuentran en ella. La trazabilidad de la capa del dominio se hace utilizando la palabra reservada namespace o espacio de nombres, que permite una organización jerárquica de los elementos del programa fuente. En este caso se indica la aplicación hospital en casa, que está en la capa del dominio.



Una búsqueda en *Google* arroja que un namespace está diseñado para proporcionar una forma de mantener un conjunto de identificadores/nombres de las clases separados unos de otros, esto va a evitar conflictos entre nombres de clases pero que están en diferentes espacios de nombres.

Con la realización de una clase de entidad, la sintaxis de C# es muy similar a la de un lenguaje Java. Sin embargo, existen algunas diferencias importantes. Primero, escribir que la clase es pública para que pueda ser utilizada desde otras capas de la arquitectura. Luego está el identificador de la clase que en este caso es persona. Aquí se debe abrir y cerrar corchetes para agrupar los componentes de la clase que pueden ser atributos y métodos, entre otros. Los atributos de la clase deberían ser privados, pero en C# existe el concepto de propiedad, que encapsula un atributo privado para que pueda ser accedido externamente, por ejemplo, si se escribe la propiedad ID lo que se dice es que internamente hay un atributo privado llamado id con minúscula. Los comandos get y set permiten configurar la propiedad de lectura y/o escritura. Por convención las propiedades van con notación Pascal que es la que se observa en la propiedad número teléfono. La propiedad final que está resaltada se analizará con más detalle al momento de implementar las clases en C#.

#### Public class Persona

```
public int Id { get;set; }
public string Nombre { get;set; }
public string Apellidos { get;set; }
public string NumeroTelefono { get;set; }
public Genero Genero { get;set; }
```

Propiedades

Otra cosa que se debe implementar son las relaciones entre clase, en el ejemplo se tiene la herencia y el diagrama muestra que un paciente hereda de la clase de persona y en C# esto se expresa poniendo dos puntos al final de la clase hija seguida de la clase padre o base.

```
Public class Paciente : Persona
{
}
```

El último por implementar es la relación de asociación que se implementa definiendo una propiedad en la clase origen de la relación, es decir, paciente del tipo de la clase que se está asociando, en este caso FamiliarDesignado. En esta implementación la propiedad tiene el mismo nombre del tipo, es decir, familiar designado. Esta no siempre es una muy buena práctica porque puede generar confusiones y podría cambiar simplemente por familiar o utilizar guión bajo aunque ahí se estaría violando el estándar de notación Pascal.

#### Public class Paciente

```
{
    Public FamiliarDesignado FamiliarDesignado {get;set;}
}
```

En conclusión, se ha procurado establecer la capa del dominio. Primero, cómo se realiza/implementa la capa. Luego, cómo se hacen las clases. Además de analizar el concepto de propiedades que es bastante utilizado en C# y, por último, la implementación de las relaciones.

## AAISION TIC2022



