

DA2 Teórico - Clase 1

16/08/2021

Diseño de aplicaciones II - Nicolás Fierro -
fierrodilorenzo@gmail.com

Hoy:

- Introducción al curso
- Temario
- Eventos del semestre
- Repaso de Diseño de aplicaciones I

Temario

Teórico:

- Elementos de arquitecturas web y distribuidas (REST y diseño de APIs)
- Arquitecturas y modelo 4 + 1 (UML avanzado)
- Aplicación de patrones de diseño
- Aplicación de principios de diseño (clase y paquetes)
- Métricas de diseño
- Diagramas y maquinas de estado
- Introducción a frameworks

Tecnología:

- Web APIs en [ASP.NET](#) Core
 - Unit Tests y Mocking
 - Inyección de dependencias
 - Filters
 - Etc.
- Reflection
- Despliegue de Web API
- SPA en Angular

Eventos del semestre:

- 30/08 _ Lectura Obligatorio 1 (min: 0 | max: 20)

- 07/10 _ Entrega Obligatorio 1
- 07/10 _ Lectura Obligatorio 2 (min: 10 | max: 30)
- 18/11 _ Entrega Obligatorio 2
- 29/11 _ Parcial (min: 15 | max: 35)
- Pruebas de actuación en clase (15 pts)
 - 5 pruebas (5 pts c/u)
 - Se toman las mejores 3.
 - 3 de teórico y 2 de tecnología

Repaso de Diseño de aplicaciones 1:

- TDD
- Clean Code
- Patrones de diseño y principios de diseño
- Entity Framework (ORM)
- UML (secuencia, clases, paquetes, comunicacion)
- Conceptos Programacion orientada a objetos (POO)
- Git (Git Flow)
- Windows Forms
- Solución: archivo .sln, es la forma que tenemos de agrupar varios proyectos. Hace alusión al concepto aplicación.
- Proyecto: (biblioteca de clases, windowsforms, tests): me permite clases y paquetes.
- Paquetes: En NET son los namespaces.
- Ensablados: es una unidad de despliegue

```
namespace DA1.DataAccess.context
{
  ...
}
```

Conceptos de Objetos y Clases:

Clase: Especificacion para objetos, que puede tener atributos y métodos.

Objetos:

- Desde el punto de diseño: entidad tangible del problema.

- Desde el punto de vista de la POO: tiene identidad, estado y comportamiento.

Los atributos y metodos de las clases tienen diferentes visibilidades:

- + public: se puede acceder desde cualquier lado.
- - private: se acceder solo desde la misma clase.
- # protected: se puede acceder a esos metodos o atributos las clases que heredan.
- ~ internal: solo sera visible dentro del mismo ensamblado.



