

Clase 6 - Teórico

20/09/21

Hoy:

- Resolver primera prueba de actuación en clase Teórico
- Dudas obligatorio
- Modelo 4+1

Prueba de actuación en clase de Teórico del 11 de octubre se cambia para el **4 de octubre**.

```
[
  {
    Project: {
      id: 1,
      name: Pepito,
    },
    Count: 10
  }
]
```

```
public class ProjectBugsCount
{
    public Project Project { get; set }
    public int count {get; set;}

    public ProjectBugsCount(Dictionary<Project,
int> values) {}
}

return Dictionary<Project, int>() {
    project 1, 10}
```

```
En el controller
return new List<ProjectBugsCount>();
```

Modelo 4 + 1

Es un modelo para representar diferentes partes de un sistema por Philippe Kruthen.

Me permite ver un sistema desde diferentes puntos de vista o partes del mismo.

Vista lógica

- Se interesa mas el analista de requerimientos
- Describe como se soportan los requerimientos funcionales del sistema, esta enfocada en describir la estructura y funcionalidad del sistema
 - Diagramas de clases
 - Diagramas de objetos (nuevo)
 - Diagramas de interacción
 - Diagramas de estructura compuesta (nueva)
 - Diagrama de estado (nuevo)

Vista de componentes

- Se interesa mas los desarrolladores
- Muestra el sistema desde la perspectiva del programador y esta enfocada en los componentes y/o artefactos del software.
 - Diagramas de paquetes
 - Diagramas de componentes

Vista de procesos

- Se interesa mas los arquitectos
- Trata los aspectos dinámicos del sistema, se enfoca en el comportamiento del sistema en tiempo de ejecución. La vista considera los atributos de calidad (disponibilidad, seguridad, etc).
 - Diagrama de actividad

Vista de despliegue

- Se interesa mas quien vaya a desplegar o publicar el sistema

(devops, infraestructura)

- Describe la relación entre el software y hardware, esta relacionada con los componentes del software en la capas física y las conexiones físicas entre estos componentes.

Vista de casos de uso

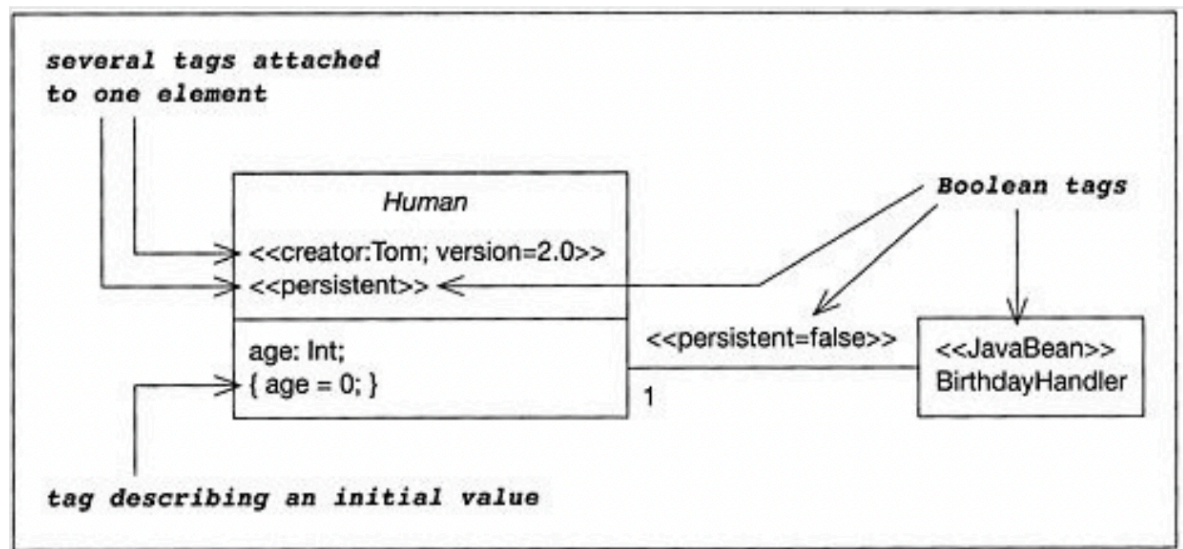
- Describe el comportamiento del sistema en escenario, desde la perspectiva de los usuarios, analistas, etc.

Conceptos de UML 2 Avanzado

Mecanismos de extensibilidad:

Stereotypes: `<<nombre_de_estereotipo>>`

Tagged values: { clave = valor }



Interfaz

En UML 2 existe una nueva notación de interfaz denominada "interfaz provista" e "interfaz requerida". Se basa en la notación lollipop/hook (chupetín/gancho).

- Interfaz provista: hacemos referencia a la implementación de cierta interfaz. Cuando decimos que una clase provee una interfaz, lo que estamos diciendo es que la clase implementa dicha interfaz (lollipop) **—○**
- Interfaz requerida: hacemos referencia la uso o dependendencia de una clase sobre una interfaz existente.

Cuando decimos que una clase requiere una interfaz,
estamos diciendo que la clase precisa de dicha interfaz.

