

**Teoría ISW**

# **Evaluación Teoría Recuperación**

**09-02-2024**

**3h 30 min**

## **Normas a seguir:**

- No abrir el enunciado hasta que lo indique el/la profesor/a del aula.
- Puede contestar a las preguntas en las hojas proporcionadas.
- Entrega:
  - El/la alumno/a se debe identificar ante el profesor en el momento de la entrega.
  - El enunciado y las respuestas se depositarán en una caja habilitada para ello.
  - El procedimiento de entrega se realizará de forma ordenada. En cada momento solo podrá estar un/a alumno/a entregando.

**Ingeniería del Software**

ETS Ingeniería Informática

DSIC – UPV

**Curso 2023-2024**

Apellidos y nombre: \_\_\_\_\_

Grupo de teoría: \_\_\_\_\_

**Tiempo: 3 horas 30 minutos**

**Cuestiones** (1,75 puntos)

1. (1 punto) En un producto software que sigue una arquitectura de tres capas cerradas, ¿qué estrategia de diseño/implementación seguirías para la comunicación entre capas? Ilústralo con un ejemplo ¿Qué beneficios aporta tu solución?
2. (0,75 puntos) Indica al menos 4 diferencias entre metodologías ágiles y tradicionales.

**Problemas (8.25 puntos)**

(4 puntos) Obtenga el diagrama de clases UML (2 puntos) y el modelo de casos de uso (diagramas de contexto y estructurado) (2 puntos) para el sistema descrito. (Diagrama de clases: ponga nombre a todas las relaciones y no defina métodos, ni tipos en los atributos).

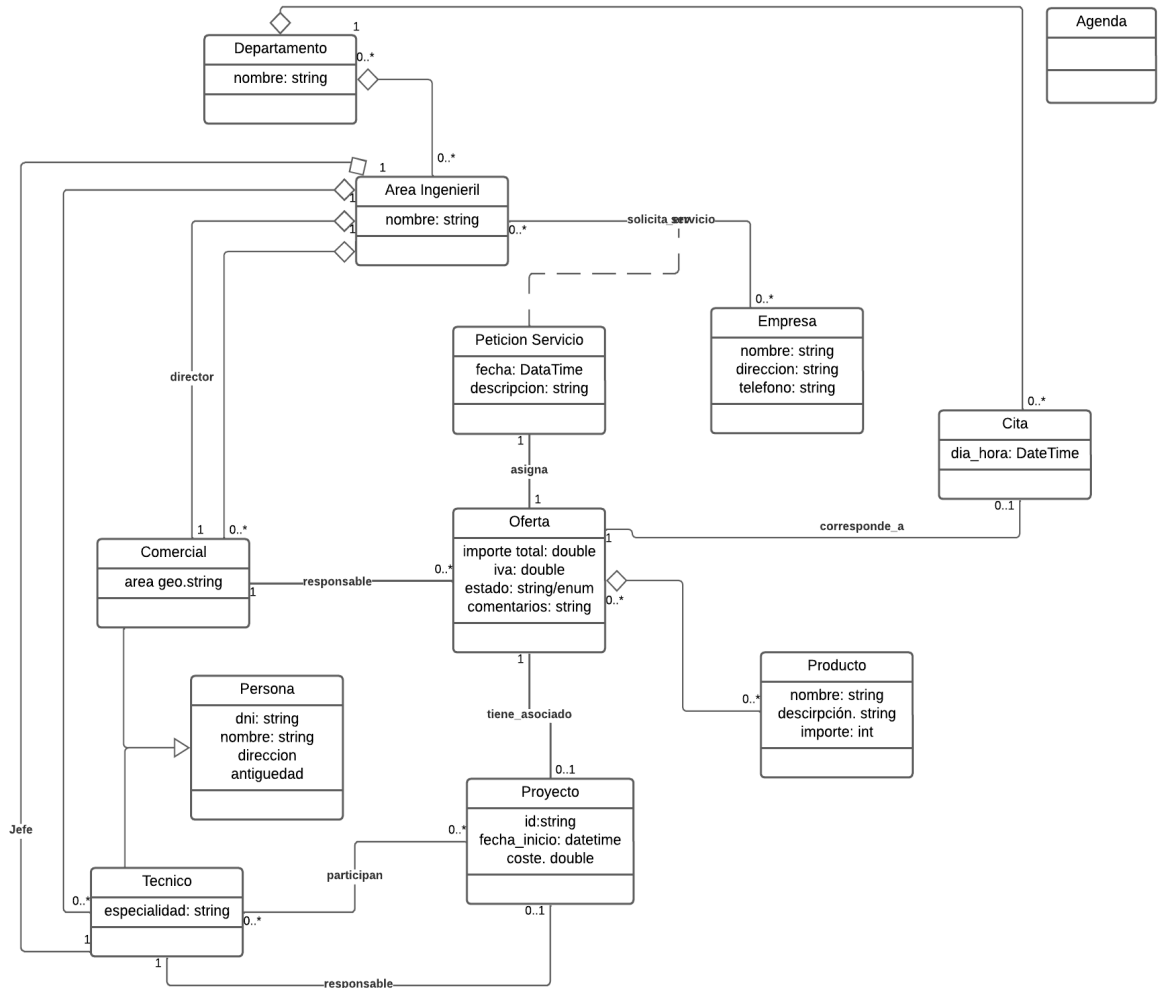
La empresa IProyectos dedicada a proyectos de ingeniería industrial ha encargado a ISW Soft el desarrollo de una aplicación web para la gestión de ofertas y proyectos a sus clientes. IProyectos está especializada en distintas áreas ingenieriles, como son el tratamiento de superficies (pintura industrial), protección térmica, protección radiológica, etc. Para todas ellas, el departamento de coordinación de ofertas es el encargado de la gestión de las ofertas-proyectos. Dicho departamento cuenta con comerciales especializados en cada una de las áreas que preparan las ofertas a los potenciales clientes, uno de los cuales actúa como director de coordinación de ofertas, siendo el que finalmente autorice la oferta. Para los proyectos, el departamento cuenta con técnicos especializados por áreas, e igualmente, uno de ellos actúa de jefe. De todos ellos se registra, dni, nombre, dirección y antigüedad. De los comerciales se registra el área geográfica asignada, y de los técnicos, su especialidad.

Una empresa (cliente potencial), a través de la aplicación web, podrá registrarse con su nombre, dirección y teléfono, y solicitar los servicios de varias áreas ingenieriles. Por cada solicitud realizada a un área, se genera una petición de servicio, que queda registrada en el sistema con una fecha y una descripción. A partir de dichas peticiones, que los comerciales podrán consultar en cualquier momento, se generan las ofertas. Para cada petición, un comercial registrará una oferta, pasando a ser el responsable de dicha oferta. Dicha oferta podrá ser registrada tras la consulta de la petición (si lo desea), o bien en cualquier momento. La oferta será por un servicio concreto de un área, y en ella se desglosarán tanto el conjunto de productos a utilizar, con su nombre, descripción e importe, como el importe total de la oferta y el iva aplicado. Todas las ofertas deben ser aprobadas por el director de coordinación de ofertas. Para facilitar esta labor, al final de cada día, el sistema debe enviar una notificación al director de coordinación de ofertas con el conjunto de nuevas ofertas registradas y pendientes de aprobación. En cualquier caso, el director de coordinación puede consultar todas las ofertas cuando lo desee. Cuando la oferta ya está aprobada, el comercial concertará una cita con la empresa, la cual quedará registrada en la agenda del departamento. Se registrará día y hora. Cada comercial tendrá una vista parcial de la agenda que incluye sólo sus citas con empresas, y que podrá actualizar cuando lo desee.

Por otra parte, cuando el director de coordinación consulta una oferta siempre emite una valoración en el momento, pudiendo aprobarla, rechazarla, o bien puede dejarla en estado de revisión para que sea modificada de acuerdo con los comentarios que anote. El comercial deberá consultar cada día las ofertas de las que es responsable para modificar aquellas que estén en estado de revisión. Si la oferta no está aprobada, no podrá realizarse la visita.

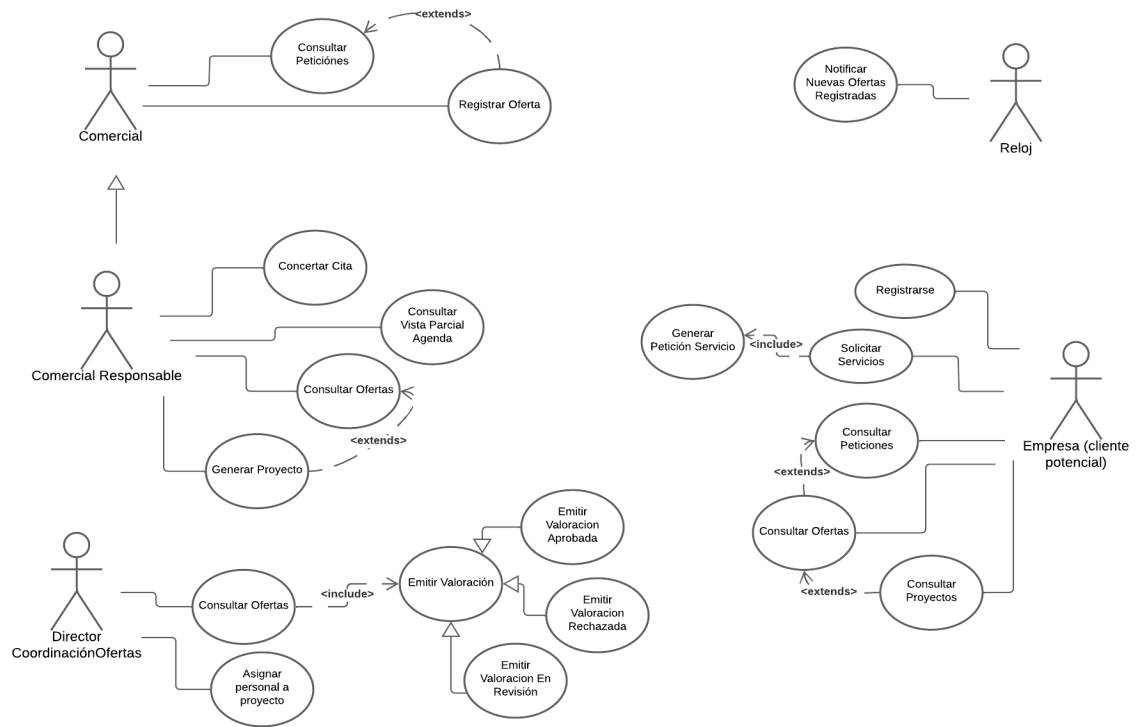
La empresa (cliente potencial) podrá consultar sus peticiones. Cuando el comercial visita a la empresa, ésta podrá aceptar o no la oferta que se le presenta en la reunión. Si la oferta es aceptada, el comercial generará un proyecto asociado a dicha oferta, con un identificador, la fecha de inicio y el coste total. El director de coordinación de ofertas asigna al proyecto creado, un conjunto de técnicos participantes y un responsable técnico. A partir de ese momento, la empresa podrá consultar en la web, las peticiones realizadas, la oferta que se le ha presentado para cada petición, y los proyectos (si se ha generado para la oferta).

**Ingeniería del Software (ISW)**  
Evaluación Teoría - Recuperación  
09-02-2024  
ETSInf-UPV



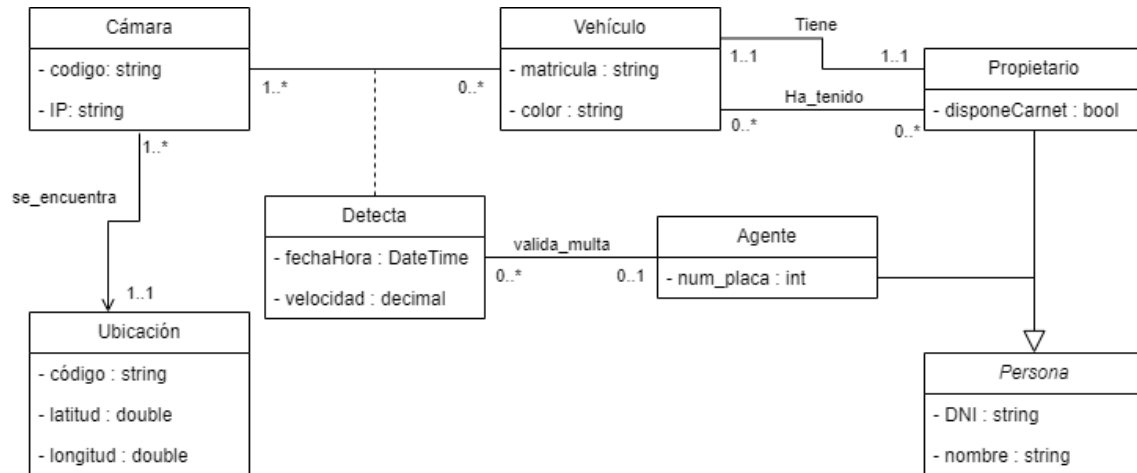
// Puede aparecer la clase Agenda o no.

**Ingeniería del Software (ISW)**  
Evaluación Teoría - Recuperación  
09-02-2024  
ETSInf-UPV



// Emitir valoración puede ser más simple, sin herencia (o con extends)

5. (2.25 puntos) Dado el siguiente diagrama de clases



- a) (1 punto) Realice el diseño en C# de las propiedades de cada clase siguiendo las guías de diseño conocidas. Se deben declarar todos los atributos que se deducen del modelo, pero **no se pide ningún método**.

```

public abstract class Persona
{
    public string DNI { get; set; }
    public string Nombre { get; set; }
}

public class Propietario : Persona
{
    public bool DisponeCanet { get; set; }
    public Vehiculo Tiene { get; set; }
    public ICollection<Vehiculo> Ha_tenido { get; set; }
}

public class Agente : Persona
{
    public int Num_placa { get; set; }
    public ICollection<Detecta> Valida_multa { get; set; }
}

public class Vehiculo
{
    public string Matricula { get; set; }
    public string Color { get; set; }
    public ICollection<Propietario> Ha_tenido { get; set; }
    public Propietario Tiene { get; set; }
    public ICollection<Detecta> Detectados { get; set; }
}

public class Detecta
{
    public DateTime FechaHora { get; set; }
    public Decimal Velocidad { get; set; }
    public Vehiculo detectado { get; set; }
    public Camara Que_detecta { get; set; }
    public Agente Valida { get; set; }
}
    
```

```
public class Ubicación
{
    public String codigo { get; set; }
    public double latitud { get; set; }

    public double longitud { get; set; }

    //public Camara Se_encuentra { get; set; }
}

public class Camara
{
    public String Codigo { get; set; }
    public String Ip { get; set; }
    public Ubicación Se_encuentra { get; set; }
    public ICollection<Detecta> Detectados { get; set; }
}
```

- b) (0.5 puntos) Realice el diseño en C# del constructor con parámetros de cada clase (sólo la cabecera, no los implemente).

```
public Persona(string dNI, string nombre)

//Se relaja
public Propietario(string dNI, string nombre, bool disponeCanet) : base(dNI,
nombre)//, Vehiculo tiene)

public Agente(string dNI, string nombre, int num_placa) : base(dNI, nombre)

//Relajado
public Vehiculo(string matricula, string color, Propietario tiene)//, Detecta
etectado)

public Detecta(DateTime fechaHora, decimal velocidad, Vehiculo detectado,
Camara que_detecta)

public Ubicación(string codigo, double latitud, double longitud)

public Camara(string codigo, string ip, Ubicación se_encuentra)
```

- c) (0.75 puntos) Usando los constructores del apartado anterior cree un objeto de cada clase, si es posible. Los objetos creados deben respetar el modelo dado.

```
static void Main(string[] args)
{
    Propietario propietario = new Propietario("123456789E", "JJ", true);
    Agente agente = new Agente("234567890A", "TT", 123);

    Vehiculo coche = new Vehiculo("1234ABC", "rojo", propietario);
    propietario.Tiene=coche;

    Ubicación lugar = new Ubicación("U01", 12, 23);
    Camara cam = new Camara("C01", "123.456", lugar);

    Detecta deteccion = new Detecta(DateTime.Now, (decimal)123.45, coche,
cam);
```

## Ingeniería del Software (ISW)

Evaluación Teoría - Recuperación

09-02-2024

ETSInf-UPV

```
cam.AddDeteccion(deteccion);  
coche.AddDeteccion(deteccion);  
}
```



3. (1.25 puntos) Diseñe los casos de prueba para el siguiente fragmento de código siguiendo la técnica del camino básico de caja blanca, dibuje el grafo de flujo, calcule la complejidad ciclomática, especifique los caminos independientes y los casos de prueba asociados a cada camino.

El método `SelectJoke` pertenece a la clase `Elf`, que entre otros, tiene como atributos:

- `MAX_CHILD`: edad máxima de las bromas que tiene el elfo.
- `jokes`: una lista de objetos `Joke`, que siempre está inicializada.

El método `Next(maxValue)` de la clase `Random` ofrece un número aleatorio positivo menor a `MaxValue`.

Para diseñar los casos de prueba, puede considerarse que para un `MAX_CHILD` de 17, un `Joke` válido sería {description: "Wrapping the stair rail with toilet paper", age:8}

```
public Joke SelectJoke(int childAge)
{
    if (jokes.Count == 0)
        throw new InvalidOperationException("No jokes available.");

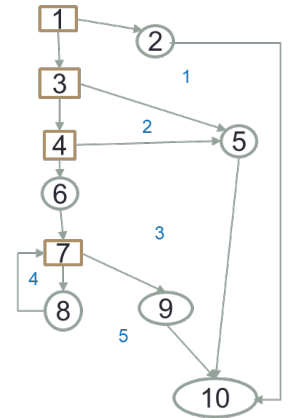
    if (childAge <= 0 || childAge > MAX_CHILD)
        throw new ArgumentOutOfRangeException("Incorrect child age.");

    Joke selectedJoke = null;
    bool selected = false;
    Random rnd = new Random();
    int jokePos;
    while (!selected)
    {
        jokePos = rnd.Next(jokes.Count);
        selectedJoke = jokes.ElementAt(jokePos);
        selected = true;
        jokes.Remove(selectedJoke);
    }
    return selectedJoke;
}
```

**Solución:**

```

public Joke SelectJoke(int childAge)
{
    if (jokes.Count == 0) 1
        throw new InvalidOperationException("No jokes available."); 2
    if (childAge <= 0 || childAge > MAX_CHILD 3
        throw new ArgumentOutOfRangeException("Incorrect child age."); 4
    Joke selectedJoke = null;
    bool selected = false;
    Random rnd = new Random(); 6
    int jokePos;
    while (!selected) 7
    {
        jokePos = rnd.Next(jokes.Count);
        selectedJoke = jokes.ElementAt(jokePos); 8
        selected = true;
        jokes.Remove(selectedJoke);
    }
    return selectedJoke; 9
} 10
    
```



$V(G) = 5$   
 $\text{Áreas} = 7$   
 $\text{Nodos predicado} = 4 \rightarrow 4 + 1 = 5$   
 $\text{Nodos} = 10 \rightarrow 13 - 10 + 2 = 5$   
 $\text{Aristas} = 13$

Camino	Entradas			Salidas		
	MAX_CHILD	childAge	Jokes[ {Description, age}]	Joke Returned{Description, age}	Jokes	Exception
1-2-10	17	10	[]	null	[]	InvalidOperationException.Message: "No jokes available"
1-3-5-10	17	-1	[{"Wrapping the stair rail with tolite paper", 8}]	null	[{"Wrapping the stair rail with tolite paper", 8}]	ArgumentOutOfRangeException.Message: "Incorrect child age"
1-3-4-5-10	17	18	[{"Wrapping the stair rail with tolite paper", 8}]	null	[{"Wrapping the stair rail with tolite paper", 8}]	ArgumentOutOfRangeException.Message: "Incorrect child age"
1-3-4-6-7-9-10			No puede probarse			
1-3-4-6-7-8-7-9-10	17	8	[{"Wrapping the stair rail with tolite paper", 8}]	{"Wrapping the stair rail with tolite paper", 8}	[]	---

## Ingeniería del Software (ISW)

Evaluación Teoría - Recuperación

09-02-2024

ETSIInf-UPV

4. (0.75 puntos) Aplique la **técnica de la partición equivalente** de caja negra para obtener una **tabla con las clases de equivalencia** numeradas con las siguientes cuatro columnas: entrada bajo consideración, clases válidas, clases inválidas, heurística aplicada. Además, debe **indicar el número de casos de prueba válidos e inválidos** que se obtienen, **sin diseñarlos**.

Se quiere probar el funcionamiento de un módulo de búsqueda de contenidos subidos a la plataforma UPVTube. El módulo permitirá filtrar los contenidos por diversos criterios de búsqueda:

- Nick del creador: es una cadena de tamaño máximo 8 que puede contener letras y dígitos.
- Fecha de subida: fecha anterior a la fecha actual con formato DD/MM/YYYY.
- Puntuación de los suscriptores: cadena que representa un valor numérico entero entre 1 y 5.

El módulo devolverá una lista con los ID de los contenidos que cumplan los criterios de búsqueda: creador indicado, fecha igual o posterior a la indicada y puntuación igual o superior a la indicada. Ninguno de los criterios de búsqueda puede ser vacío. Además devolverá el siguiente valor de retorno:

- S1: Número de contenidos encontrados
- S2: El formato del nick es incorrecto
- S3: El nick no corresponde a ningún creador de contenidos
- S4: Fecha inexistente o de formato incorrecto
- S5: La fecha de subida debe ser igual o anterior a la fecha actual
- S6: La puntuación debe ser un entero entre 1 y 5

### Solución:

Condición de entrada	Clases válidas	Clases no válidas	Heurística
Nick	1<=Longitud <=8 (1) Sólo dígitos o letras Existe en BD	Longitud<1 (2) Longitud>8 (3) Carácter incorrecto (4) No existe en BD (5)	Número finito valores Booleana Booleana
Fecha de subida	Fecha DD/MM/YYYY existente, igual o anterior a la actual (6)	Formato incorrecto (7) Fecha inexistente (8) Fecha posterior (9)	Booleana Booleana Booleana
Puntuación	Entero entre 1 y 5 (10)	Puntuación <1 (11) Puntuación >5 (12) No entero (13)	Rango Booleana

1 caso válido y 10 no válidos.

**Ingeniería del Software (ISW)**  
Evaluación Teoría - Recuperación  
09-02-2024  
ETSInf-UPV

Clases Válidas	Nick	Fecha	Puntuación	Lista Contenidos	Valor Retorno
(1) – (6) – (10)	"Bart123"	"09/10/2023"	1	["1111"]	S1

Clases No Válidas	Nick	Fecha	Puntuación	Lista Contenidos	Valor Retorno
(2) – (6) – (10)	""	"09/10/2023"	1	[]	S2
(3) – (6) – (10)	"Bart12345"	"09/10/2023"	1	[]	S2
(4) – (6) – (10)	"Bart%%3"	"09/10/2023"	1	[]	S2
(5) – (6) – (10)	"Homer"	"09/10/2023"	1	[]	S3
(1) – (7) – (10)	"Bart123"	"09-10-2023"	1	[]	S4
(1) – (8) – (10)	"Bart123"	"30/02/2020"	1	[]	S4
(1) – (9) – (10)	"Bart123"	"30/01/2024"	1	[]	S5
(1) – (6) – (11)	"Bart123"	"09/10/2023"	0	[]	S6
(1) – (6) – (12)	"Bart123"	"09/10/2023"	6	[]	S6
(1) – (6) – (13)	"Bart123"	"09/10/2023"	A	[]	S6