

OBSERVACIONES DE CORRECCIÓN POR ASPIRANTE.

El tiempo total para la realización de la prueba es de 2 horas

Ejercicio 1: (Total 1 Punto)

La organización X dispone de un conjunto de aplicaciones Java que regulan diferentes aspectos de la lógica de negocio de dicha organización y que se comunican entre ellas a través de RMI. Con el fin de agilizar el desarrollo se prevé la creación de un mecanismo que facilite la inyección de los objetos remotos de una determinada aplicación en otra. Como parte del equipo de desarrollo se le encomiendan las siguientes tareas:

1.1.- Diseñe e implemente una anotación java para inyectar en un atributo un objeto remoto RMI. Se debe tener en cuenta que el objeto remoto puede estar en una ubicación remota diferente. En caso de que no se especifique una ubicación remota explícita, la anotación debe buscar en la máquina local y el puerto por defecto de RMI. La interfaz debe permitir especificar también el nombre del registro a mayores del nombre del objeto remoto. **(1 punto)**

Ejercicio 2: (Total 3.5 puntos)

Como miembro del equipo de desarrollo de la organización X, se le encomienda la tarea de tomar como partida esta clase Java de un software basado en la versión Java SE 6 (1.6) y realizar los cambios que se sugieren a continuación:

```
import java.util.Date;

public class Persona {

    private final String nombre;
    private final String primerApellido;
    private final String segundoApellido;
    private final Date fechaDeNacimiento;
    private final String dni;

    public Persona( String nombre, String primerApellido, String
                    segundoApellido, Date fechaDeNacimiento, String dni) {

        if (nombre == null || primerApellido == null || segundoApellido == null
            || fechaDeNacimiento == null || dni == null) {
            throw new IllegalArgumentException("No se admiten valores nulos");
        }

        this.nombre = nombre;
        this.primerApellido = primerApellido;
        this.segundoApellido = segundoApellido;
        this.fechaDeNacimiento = fechaDeNacimiento;
        this.dni = dni;
    }

    public String getNombre() {
        return nombre;
    }

    public String getPrimerApellido() {
        return primerApellido;
    }

    public String getSegundoApellido() {
        return segundoApellido;
    }

    public String getNombreCompleto() {        return nombre + " " + primerApellido+ " " +
segundoApellido;
    }

    public Date getFechaDeNacimiento() {
        return fechaDeNacimiento;
    }

    public String getDni() {
        return dni;
    }

    @Override
    public String toString() {
        return "Persona{" +
            "nombre='" + nombre + '\'' +
            ", primerApellido='" + primerApellido + '\'' +
            ", segundoApellido='" + segundoApellido + '\'' +
            ", fechaDeNacimiento='" + fechaDeNacimiento + '\'' +
            ", dni='" + dni + '\'' + '}';
    }
}
```

2.1.- Implementar un método compatible con la versión actual que en base a la fecha de nacimiento de la persona devuelva su edad. **(1 Punto)**

Como parte del proceso de renovación y mejora continua de cualquier organización además del cambio sugerido con anterioridad se propone su migración a la última versión LTS (Long-term support) de Java SE. Por ello también se le encomienda las siguientes tareas:

2.2.- Indicar cuál será la versión Java SE a utilizar. **(0,5 puntos)**

2.3.- Tratándose de una clase que almacena información inmutable y aprovechando las características de las últimas versiones de Java SE se decide migrar la clase Persona a un registro de java (java record). Escriba la implementación de un registro “Persona” equivalente a la definición de la clase propuesta en el enunciado, teniendo también en cuenta el método implementado en el punto 2.1. **(2 puntos)**

Ejercicio 3: (Total 3,5 puntos)

Una administración pública ha desarrollado un sistema de gestión de expedientes administrativos. La base de datos del sistema está compuesta por las siguientes tablas:

- **usuarios:** Contiene la información básica de los ciudadanos. Campos:
 - *id_usuario*: Identificador único del usuario.
 - *nombre*: Nombre completo del usuario.
 - *email*: Correo electrónico del usuario.
- **registros:** Almacena los registros de entrada o salida que dan origen a los expedientes administrativos. Campos:
 - *id_registro*: Identificador del registro.
 - *fecha_registro*: Fecha en la que se produjo el registro.
 - *id_usuario*: Usuario que realizó el registro.
- **expedientes:** Contiene los datos principales de cada expediente tramitado en la plataforma. Campos:
 - *id_expediente*: Identificador único del expediente.
 - *id_registro*: Registro a partir del cual se ha generado el expediente.
 - *fecha_creacion*: Fecha de creación del expediente.
- **interesados_expediente:** Relaciona expedientes con los usuarios que figuran como interesados en su tramitación. Un expediente puede tener N usuarios asociados. Campos:
 - *id_expediente*: Identificador del expediente.
 - *id_usuario*: Usuario interesado en el expediente.
- **documentos:** Almacena los documentos vinculados a los expedientes, incluyendo su tipo y tamaño. Un expediente puede tener N documentos asociados. Campos:
 - *id_documento*: Identificador del documento.
 - *id_expediente*: Expediente al que pertenece el documento.
 - *tipo_documento*: Tipo de documento (ej.: informe, solicitud, resolución, etc.).
 - *tamano_kb*: Tamaño del documento en kilobytes.
 - *fecha_subida*: Fecha en la que se subió el documento al sistema.

3.1.- Partiendo de las tablas anteriores, se solicita crear una vista que muestre para cada expediente lo siguiente:

- El ID del expediente
- El número total de documentos asociados
- La media de tamaño de los documentos
- El tipo de documento más frecuente
- La posición del expediente en función del tamaño total de documentos, ordenada de mayor a menor.

(1,5 puntos)

3.2.- Dicha administración ha incorporado a su CPD, un sistema de almacenamiento de 2 MB, con la intención de archivar la documentación de los expedientes más antiguos. Para ello se le pide que desarrolle una consulta que devuelva todos los expedientes ordenados de más antiguos a más actuales, cuya suma de tamaño de documentos no supere el tamaño del nuevo sistema de almacenamiento.

Aclaraciones, Solo se pueden archivar expedientes completos, es decir, con todos sus documentos. **(2 puntos)**

Ejercicio 4: (Total 2 puntos)

En España cada administración debe disponer de un Registro Electrónico General en el que se dejará constancia de todas las solicitudes y documentos que se presenten o reciban en dicha administración. Cada asiento que se practique en dicho registro ha de contar con un número de entrada de registro, un epígrafe expresivo de su naturaleza, fecha y hora de su presentación, la identificación del interesado que lo presenta, órgano administrativo remitente, si procede, y persona u órgano administrativo al que se envía, y, en su caso, referencia al contenido del documento que se registra. A fin de que un interesado pueda tener un documento que acredite la presentación de una solicitud en un registro, las administraciones deben emitir automáticamente un recibo en el que se incluya la fecha y hora de presentación, el número de entrada de registro, así como un recibo acreditativo de otros documentos que, en su caso, lo acompañen, que garantice la integridad y el no repudio de los mismos.

4.1.- Fruto de un análisis previo, la administración para la que usted trabaja cree que no cumple con los requisitos anteriormente citados en lo que respecta a la emisión del recibo o justificante. Como miembro del departamento de informática de dicha organización se le solicita que diseñe un nuevo justificante y que enumere y describa todos los mecanismos que permitan cumplir con las especificaciones anteriormente citadas. **(2 puntos)**