

Enunciado A:

Se quiere desarrollar una aplicación multiusuario, a la que se pueda acceder desde múltiples dispositivos, y que proporcione acceso a vídeos y fotografías. Tanto los vídeos como las fotografías estarán disponibles en varios formatos dependiendo del dispositivo en el que se tenga que visualizar su contenido. La aplicación deberá mostrar un catálogo general con la siguiente información de los vídeos y las fotografías: formatos disponibles, compresión de vídeo y/o audio (en el caso de vídeo), resolución (en el caso de las fotografías), espacio de almacenamiento en kb y precio de descarga. Inicialmente, la aplicación soportará conexiones a través de PCs (bajo Windows, Linux u OS X), e Iphones, pero esto se ampliará en un futuro cercano (sólo si la aplicación tiene éxito). Así mismo, inicialmente la aplicación contará con un repositorio local, pero está previsto ampliar el sistema para utilizar vídeos y fotografías de repositorios externos (a medida que crezca la demanda), haciendo transparente al usuario su búsqueda y localización. El sistema dispondrá además de una utilidad de búsqueda por palabra clave, que inicialmente tendrá ámbito local (mientras no se incluyan repositorios externos, aunque la localidad de los repositorios siempre debe ser transparente para el usuario). El sistema deberá mantener cuentas de usuario para poder realizar los cobros de las descargas, que deberán hacerse de forma segura.

1. Indica claramente qué patrón/es arquitectónicos utilizarías para desarrollar esta aplicación. Muestra la vista de módulos detallada del sistema. Es imprescindible que justifiques claramente todas tus decisiones basándote únicamente en la información del texto. No es necesario seguir el algoritmo visto en clase. (2 puntos)
2. Indica qué modelo de proceso (o combinación de modelos) seguirías para desarrollar la aplicación anterior y muestra una agenda que sirva como base para hacer estimaciones iniciales. (1 punto)
3. Sobre la agenda anterior, enumera los pasos a seguir para realizar las estimaciones iniciales, indicando claramente qué estamos estimando (y en qué unidades) y cómo hacerlo en cada uno de dichos pasos. Es imprescindible que indiques claramente las entradas y las salidas de cada uno de los pasos. (1.5 puntos)
4. Dado el enunciado A, y asumiendo un desarrollo UP, realiza un plan detallado de las dos primeras iteraciones. (1 punto)
5. Teniendo en cuenta el enunciado A, indica cuáles serían los elementos de configuración y estructura de directorios iniciales de tu proyecto, así como el conjunto de órdenes CVS que necesitarías para empezar el desarrollo de dicho proyecto. (1.5 puntos)
6. Diseña una serie de casos de prueba para el siguiente enunciado utilizando el método de pruebas que consideres adecuado. (1.5 puntos)

```
public boolean validaNif (String nif) {  
    if (nif.length() != 9 || nif == null) {  
        return false;  
    }  
}
```

```

String dni = nif.subString (0, 8);
char letra = nif.charAt (8);
Pattern pattern = Pattern.compile ("[0 - 9] {8, 8}");
Matcher matcher = pattern.matcher (dni);
String letras = "TRWAGMYFPDXBNJZSQVHLCKE";
long ldni = Long.parseLong (dni);
int indice = (int) (ldni % 23);
char letraEsperada = letras.charAt (indice);

return matcher.matches() && letra == letraEsperada;
}

```

7. Implementa con JUnit pruebas para el método calculaConsumo. Indicar todo el código necesario para automatizar dichas pruebas, de forma que se compruebe la aplicación de las dos tarifas. Suponemos que TARIFA_NOCTURNA y TARIFA_DIURNA son constantes definidas de forma global en la aplicación. (1.5 puntos)

```

public class GestorLlamadas {
    public int getHoraActual () {
        Calendar c = Calendar.getInstance ();
        int hora = c.get (Calendar.HOUR);

        return hora;
    }
    public calculaConsumo (int minutos) {
        int hora = getHoraActual ();

        if (hora < 8 || hora > 20) {
            return minutos * TARIFA_NOCTURNA;
        } else {
            return minutos * TARIFA_DIURNA;
        }
    }
}

```