

# Preguntas de Examen:

**1.-Explicar el diagrama: Se adjunta imagen.**

**2.-Aplicar el patrón composite con pruebas unitarias**

**3.-Libro de scrum leer apéndice y explicarlo:**

R: Para aplicar Scrum de una forma adecuada y obtener resultados podríamos decir que se debe elegir primero a un responsable el cual se hará cargo de la visión del proyecto, sobre lo que se hará o se va a producir, así como los pros y contras que este conlleva.

Después se debe de seleccionar un equipo, pero siempre con la idea de quienes lo harán efectivamente ya que debe de hacerse realidad la visión del responsable, teniendo en cuenta que se debe de conformar un equipo pequeño de máximo 7 personas, no uno grande ya que se volvería deficiente esta metodología.

Aplicando esto se elige al que será el Scrum Master quien es la persona que capacitará al equipo seleccionado en Scrum y este mismo eliminará todos los retrasos y contratiempos que se lleguen a presentar.

Por cuarto se hará y se tendrá que priorizar una bitácora del producto, en estas enlista todo lo que debe de hacerse para cumplir con la visión del responsable y se deberá de hacer que los integrantes del equipo se hagan cargo de los elementos en esta misma y examinar cada elemento y ver si es viable.

Después se realizará la primera reunión con el equipo, Scrum Master y el responsable para planear los Sprints, los cuales se deberán de llevar a cabo una o dos veces por semana, en estas reuniones es donde se confirmará que todos están llevando a cabo los elementos asignados, así como la comprensión a la perfección de los elementos para llevar a cabo la visión.

Se creará una tabla de Scrum con 3 columnas las cuales serán: Pendiente, En Proceso y Terminado, junto con unas notas adhesivas que llevarán los elementos a realizar, con lo cual el equipo podrá avanzar conforme se concluya cada uno de estas notas.

Se tendrá una parada diaria o un Scrum diario cada día de la semana, a la misma hora, y no más de 15 minutos de duración donde solo se reunirá el equipo y el Scrum Master y se contestarán 3 simples preguntas y al terminar se concluirá la sesión.

Por último, se demostrará en una reunión con todos los interesados en el producto lo que el equipo pudo llevar a terminado, dando a entender por terminado lo total y completamente concluido y al finalizar se realizará una retrospectiva del sprint. Se pensará en que marchó bien, como pudo marchar mejor y que se puede mejorar.

Terminado con todo esto, comenzará inmediatamente otro ciclo del siguiente sprint tomando en cuenta la experiencia del equipo con impedimentos y mejoras.

Con esto se podría resumir todo el ciclo que debe realizarse al aplicar Scrum en un equipo de trabajo para eficientar de mejor manera el tiempo y recursos que se disponen para la visión o proyecto en cuestión.

#### 4.-Define el tema de las excepciones:

R: Las excepciones son el medio que se nos ofrecen para tratar situaciones anómalas que pueden suceder cuando ejecutamos un programa, una de las situaciones que se pueden presentar, son, por ejemplo, invocar a un método sobre un objeto “null”, intentar dividir un número por “0”, intentar abrir un fichero que no existe para leerlo, por este motivo, tenemos que ser capaces de crear, lanzar y gestionar excepciones de una forma adecuada, esto es lo que se conoce como manejo o gestión de la excepciones.

También se le puede denominar como una forma de intentar conseguir que, si un código fuente no se ejecuta de la forma prevista inicialmente, el programador sea capaz de controlar esa situación y decir cómo ha de responder el programa, pero esto no significa que con las excepciones vamos a corregir errores de programación.

Todas las excepciones que podamos usar o crear propias, deben heredar de la clase Exception propia de la librería de Java, existe también una superclase de Exception, llamada Throwable que sirve para representar la clase de la que deben heredar todas las clases que queramos lanzar, por medio del comando throw.

Sólo existen dos subclases directas de Throwable en la API de Java, una es la clase Error que se utiliza para representar algunos errores poco comunes que pueden tener lugar cuando ejecutamos una aplicación en Java la otra subclase directa es Exception.

De Exception se van a derivar lo que son **RuntimeException** que representa el conjunto de las excepciones que pueden tener lugar durante el proceso de ejecución de un programa sobre la JVM, con la peculiaridad de que el usuario no tiene que prestar atención al hecho de capturarlas (Unchecked).

**EJ: -IndexOutOfBoundsException, -NumberFormatException, -NullPointerException**

Y las que vienen de **other exception** que heredan de la clase Exception en Java, pero no de **RunTimeException**, y que por tanto los usuarios deben gestionar, al hacer uso de los métodos que las lancen.

**EJ: -IOException, -FileNotFoundException**

**Ejemplo de manejo de excepciones:**

**--Con try catch:**

```
public static void main (String [] args){
```

```
    double x = 15.0;
```

```
    double y = 3.0;
```

```

try{
    System.out.println ("El resultado de la division real de " + x + " entre " + y + " es "
+ division_real (x, y));
}catch (Exception mi_excepcion){
    System.out.println ("Has intentado dividir por 0.0;");
    System.out.println ("El objeto excepcion lanzado: " +
mi_excepcion.toString());
}
}

```

--Lanzando la excepcion para que quien la invoque haga el manejo de esta:

```

public static double acceso_por_indice (double [] v, int indice) throws Exception{
    if ((0<=indice) && (indice <v.length)){
        return v [indice];
    }else {
        //Caso excepcional:
        throw new Exception ("El indice " + indice + " no es una posicion valida");
    }
}

```

**5.- Del Proyecto web tener los métodos listar, insert y update.**