EXAMEN 3ª EV - DAM

Atento al formato de entrega de cada uno de los ejercicios. Cualquiera de ellos que no esté entregado en el formato requerido **no** será puntuado.

PARTE 2 – PRUEBAS UNITARIAS

EJERCICIO 1 (3 puntos)

A partir de la siguiente de la siguiente clase en Java, crea las pruebas unitarias para cada uno de los métodos, incluido el constructor, tal y como se detalla posteriormente:

```
public class Matricad {
    private java.util.ArrayList<String> cadenes;  // referencia a la lista
de cadenas, un campo

    /**
     * Constructor de Matricad.
     * @param dada matriu amb les cadenes per a la llista
     */
    public Matricad(String[] dada) {
        if ((dada == null) || (dada.length == 0)) {     // Verificamos que
la lista tenga valores
            throw new IllegalArgumentException();
        }
        this.cadenes = new java.util.ArrayList<>();
        for (String element : dada) {
            cadenes.add(element);
        }
    }

    /**
     * @return la cantidad de cadenas que empiezan con una letra específica.
     */
    public int getCantidadCadenasQueEmpiezanCon(char letra) {
        int contador = 0;
        for (String element : cadenes) {
            if (element.charAt(0) == letra) {
                contador++;
            }
        }
        return contador;
    }

        /**
     * @return la cantidad de cadenas que empiezan con una letra específica.
     */
    public boolean hayCadenasQueEmpiezanPor(char letra) {
        for (String element : cadenes) {
            if (element.charAt(0) == letra) {
                return true;
            }
```

```
        }
        return false;
    }

    /**
     * @return la concatenación de todas las cadenas en la lista, separadas
por un espacio.
     */
    public String getConcatenacionCadenas() {
        StringBuilder concatenacion = new StringBuilder();
        for (String element : cadenes) {
            concatenacion.append(element).append(" ");
        }
        return concatenacion.toString().trim();
    }

    /**
     * Busca todas las apariciones de una cadena y devuelve sus posiciones
como una lista de enteros.
     *
     * @param unaCadena Cadena buscada
     * @return Retorna una lista de enteros con las posiciones de la cadena
en la lista, o una lista vacía si no se encuentra.
     */
    public java.util.ArrayList<Integer> getPosicionesDe(String unaCadena)
{
        if (unaCadena == null) {      // Comprobamos que el argumento sea
válido
            throw new IllegalArgumentException();
        }
        java.util.ArrayList<Integer>        posiciones        =        new
java.util.ArrayList<>();
        for (int i = 0; i < cadenes.size(); i++) {
            String d = cadenes.get(i);
            if (d.equals(unaCadena)) {
                posiciones.add(i);
            }
        }
        return posiciones;
    }
}
```

Establece los siguientes casos de prueba:

1. **Constructor** (1,25 puntos)
   a. con un null
   b. con una matriz vacía
   c. con la matriz 'prueba', que tenga los siguientes strings:
   “hola”, “que”, “tal”, “cómo”, “estás”, “hola”

2. **getCantidadCadenasQueEmpiezanCon** (0,25 puntos)
   a. Comprueba que hay dos elementos que empiezan por “h”.

3. **hayCadenasQueEmpiezanPor** (0,25 puntos)
   a. Comprueba que verifica si hay una cadena que empieza por “t”.

No puedes emplear el método **assertEquals**

4. **getConcatenacionCadenas** (0,25 puntos)
   a. Comprueba que el método devuelve la cadena correcta.

5. **getPosicionesDe** (1 punto)
   a. Prueba que con el string “adios” devuelve una lista vacía.
   b. Prueba con el string “hola”, se devuelve una lista con las posiciones correctas.
   c. Prueba que con un null se devulve una excepción.

Recuerda que debe pasar todos los test para que las pruebas unitarias estén diseñadas de forma correcta, puesto que los métodos son correctos.

En el caso de tener que devolver una excepción, será correcto si se lanza esa excepción.

EN LA TAREA DEL EXAMEN TIENES QUE ENTREGAR EN ENLACE DEL REPOSITORIO DONDE HAYAS GUARDADO EL PROYECTO.