

Pruebas unitarias JUnit5

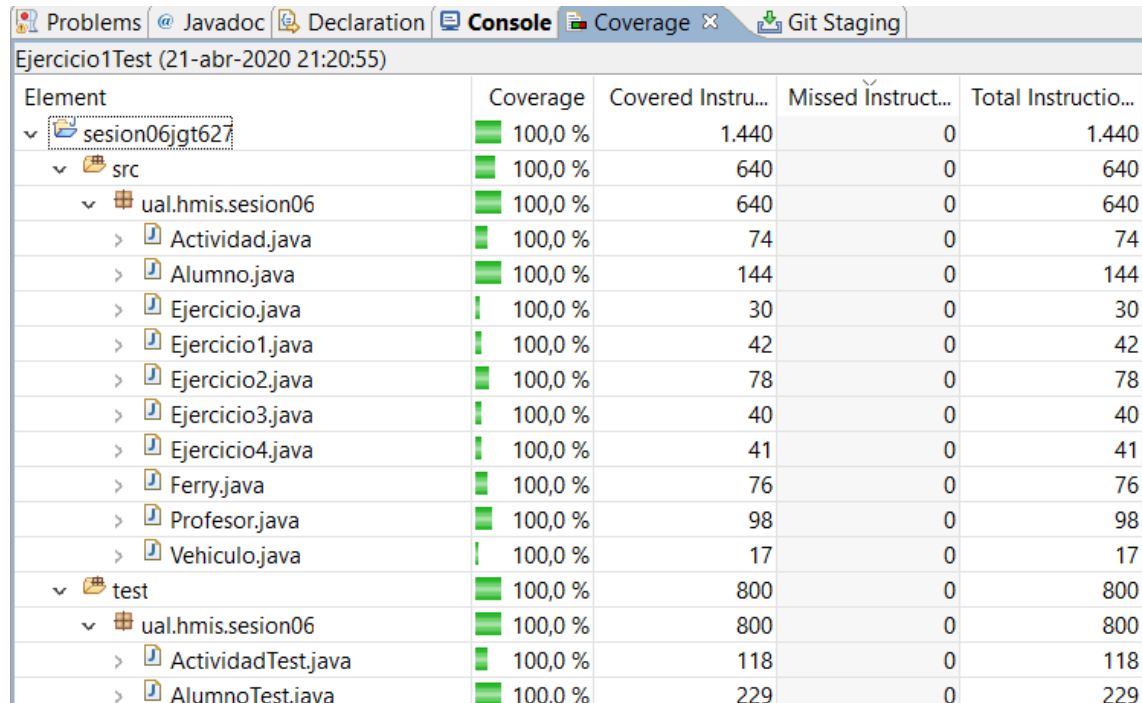
Sesión 06

JUAN PABLO GROSSO TARAZAGA

Las URLs de los repositorios son:

Donde están las pruebas (publico) → <https://github.com/jgt627/clasehmis2020-jgt627>

Forkeado (privado) → <https://github.com/jgt627/pruebas-unitarias>



Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼ sesion06jgt627	100,0 %	1.440	0	1.440
▼ src	100,0 %	640	0	640
▼ ual.hmis.sesion06	100,0 %	640	0	640
> Actividad.java	100,0 %	74	0	74
> Alumno.java	100,0 %	144	0	144
> Ejercicio.java	100,0 %	30	0	30
> Ejercicio1.java	100,0 %	42	0	42
> Ejercicio2.java	100,0 %	78	0	78
> Ejercicio3.java	100,0 %	40	0	40
> Ejercicio4.java	100,0 %	41	0	41
> Ferry.java	100,0 %	76	0	76
> Profesor.java	100,0 %	98	0	98
> Vehiculo.java	100,0 %	17	0	17
▼ test	100,0 %	800	0	800
▼ ual.hmis.sesion06	100,0 %	800	0	800
> ActividadTest.java	100,0 %	118	0	118
> AlumnoTest.java	100,0 %	229	0	229

Coverage de todo.

Ejercicio 1)

```
package ual.hmis.sesion06;

public class Ejercicio1 {
    public int transformar (int x) {
        int resultado = 0;
        if (x % 2 == 0) // % Resto de una división entre enteros (mod)
            resultado = transformar (x/2);
        else if (x % 3 == 0)
            resultado = transformar (x/3);
        else if (x % 5 == 0)
            resultado = transformar (x/5);
        else return x;
        return resultado;
    }
}
```

Ejercicio 2)

```
package ual.hmis.sesion06;

public class Ejercicio2 {
    public boolean login (String username, String password) {
        int cont1 = 0;
        int cont2 = 0;
        int cont3 = 0;
        //Comprobar que sean distintos de vacío
        if (username.isEmpty() || password.isEmpty()) return false;
        //Comprobar que la longitud sea < 30
        if(username.length() >= 30 || password.length() >= 30) return false;
        //Comprobar que password tenga mayúsculas, minúsculas y dígitos
        for(int i=0;i<password.length();i++) {
            if(Character.isUpperCase(password.charAt(i))) {
                cont1 += 1;
            }
            if(Character.isLowerCase(password.charAt(i))) {
                cont2 += 1;
            }
            if(Character.isDigit(password.charAt(i))) {
                cont3 += 1;
            }
        }
        if(cont1 == 0 || cont2 == 0 || cont3 == 0) return false;
        //Llamar al método de la bbdd
        return compruebaLoginEnBD(username, password);
    }

    public boolean compruebaLoginEnBD(String username, String password) {
        //Método mock (simulado)
        if(username.equals("user") && password.equals("Pass1"))
            return true;
        else
            return false;
    }
}
```

Ejercicio 3)

```
package ual.hmis.sesion06;

public class Ejercicio3 {

    public String EscrituraDeAsteriscos(int numAsteriscos) {
        String asteriscos = "";
        if (numAsteriscos < 0)
            asteriscos = "número erróneo";
        else if(numAsteriscos < 5)
            asteriscos = "*****";
        else if(numAsteriscos > 12)
            asteriscos = "*****";
        else
            for(int i = 0; i < numAsteriscos;i++)
                asteriscos += "*";

        return asteriscos;
    }
}
```

Ejercicio 4)

```
package ual.hmis.sesion06;

public class Ejercicio4 {
    public String caracteresIguales(String P1, String P2) {
        String resultado = "";
        for(int i=0;i<P1.length();i++) {
            for(int j=0;j<P2.length();j++) {
                if(P1.charAt(i) == P2.charAt(j)) {
                    resultado+=P1.charAt(i);
                }
            }
        }
        return resultado;
    }
}
```

Ejercicios 5, 6 y 7)

```
package ual.hmis.sesion06;

import java.util.ArrayList;

public class Alumno {
    private String nombre;
    private ArrayList<Actividad> actividadesAsignadas = new ArrayList<Actividad>();

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public ArrayList<Actividad> getActividadesAsignadas() {
        return actividadesAsignadas;
    }

    public void setActividadesAsignadas(ArrayList<Actividad> actividadesAsignadas) {
        this.actividadesAsignadas = actividadesAsignadas;
    }

    public String toString() {
        return this.nombre;
    }

    public double calcularNotaActividad(String nombreActividad) {
        double suma = 0.0;
        for(Actividad a : this.actividadesAsignadas) {
            if(a.getNombre().equals(nombreActividad)) {
                for(Ejercicio e : a.getEjercicios()) {
                    suma+=e.getPuntuacion();
                }
            }
        }
        a.setPuntuacionTotal(suma);
        return suma;
    }
}
```

```

    public String calificacion(double nota) {
        if(nota >= 0.0 && nota < 5.0) return "Suspenso";
        if(nota >= 5.0 && nota < 7.0) return "Aprobado";
        if(nota >= 7.0 && nota < 9.0) return "Notable";
        if(nota >= 9.0 && nota < 10.0) return "Sobresaliente";
        if(nota == 10.0) return "Matrícula";
        return "Error en la nota";
    }

    public String cursoSegunEdad (int anyoNacimiento) {
        switch(anyoNacimiento) {
            case 2017:
                return "1º Educación Infantil";
            case 2016:
                return "2º Educación Infantil";
            case 2015:
                return "3º Educación Infantil";
            case 2014:
                return "1º Educación Primaria";
            case 2013:
                return "2º Educación Primaria";
            case 2012:
                return "3º Educación Primaria";
            case 2011:
                return "4º Educación Primaria";
            case 2010:
                return "5º Educación Primaria";
            case 2009:
                return "6º Educación Primaria";
            case 2008:
                return "1º Educación Secundaria";
            case 2007:
                return "2º Educación Secundaria";
            case 2006:
                return "3º Educación Secundaria";
            case 2005:
                return "4º Educación Secundaria";
            default: return "";
        }
    }
}

```

Coverage de Alumno.

```

package ual.hmis.sesion06;

import java.util.ArrayList;

public class Profesor {
    private ArrayList<Alumno> alumnosDePracticas;

    public ArrayList<Alumno> getAlumnosDePracticas(){
        return alumnosDePracticas;
    }

    public void setAlumnosDePracticas(ArrayList<Alumno> alumnosDePracticas) {
        this.alumnosDePracticas = alumnosDePracticas;
    }

    public void calificarAlumnos() {
        for(int i=0; i<alumnosDePracticas.size(); i++) {
            for(int j=0; j<alumnosDePracticas.get(i).getActividadesAsignadas().size(); j++) {
                double suma = 0.0;
                for(int k=0; k<alumnosDePracticas.get(i).getActividadesAsignadas().get(j).getEjercicios().size(); k++) {
                    double puntuacion = alumnosDePracticas.get(i).getActividadesAsignadas().get(j).getEjercicios().get(k).getPuntuacion();
                    suma += puntuacion;
                }

                if(suma >= 5)
                    alumnosDePracticas.get(i).getActividadesAsignadas().get(j).setApta(true);
                else
                    alumnosDePracticas.get(i).getActividadesAsignadas().get(j).setApta(false);
            }
        }
    }
}

```

Coverage de Profesor.

```

package ual.hmis.sesion06;

public class Ejercicio {
    private String nombre;
    private double puntuacion;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public double getPuntuacion() {
        return puntuacion;
    }

    public void setPuntuacion(double puntuacion) {
        this.puntuacion = puntuacion;
    }

    public String toString() {
        return this.nombre + " "+this.puntuacion;
    }
}

```

Coverage de Ejercicio.

```

package ual.hmis.sesion06;

import java.util.ArrayList;

public class Actividad {
    private String nombre; // nombre de la actividad
    private ArrayList<Ejercicio> ejercicios = new ArrayList<Ejercicio>();
    private boolean apta;
    private double puntuacionTotal;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public ArrayList<Ejercicio> getEjercicios() {
        return ejercicios;
    }

    public void setEjercicios(ArrayList<Ejercicio> ejercicios) {
        this.ejercicios = ejercicios;
    }

    public boolean isApta() {
        return apta;
    }

    public void setApta(boolean apta) {
        this.apta = apta;
    }

    public double getPuntuacionTotal() {
        return this.puntuacionTotal;
    }

    public void setPuntuacionTotal(double value) {
        this.puntuacionTotal = value;
    }
}

```

```

    public void agregarEjercicioCalificado(String nombreEjercicio, double puntuacion) {
        Ejercicio nuevoEjercicio = new Ejercicio();
        nuevoEjercicio.setNombre(nombreEjercicio);
        nuevoEjercicio.setPuntuacion(puntuacion);
        ejercicios.add(nuevoEjercicio);
    }

    public String toString() {
        return this.nombre + " " + this.puntuacionTotal + " " + (apta == true ? "apta" : "no apta");
    }
}

```

Coverage de Actividad.

```

package ual.hmis.sesion06;

import java.util.ArrayList;

public class Actividad {
    private String nombre; // nombre de la actividad
    private ArrayList<Ejercicio> ejercicios = new ArrayList<Ejercicio>();
    private boolean apta;
    private double puntuacionTotal;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public ArrayList<Ejercicio> getEjercicios() {
        return ejercicios;
    }

    public void setEjercicios(ArrayList<Ejercicio> ejercicios) {
        this.ejercicios = ejercicios;
    }

    public boolean isApta() {
        return apta;
    }

    public void setApta(boolean apta) {
        this.apta = apta;
    }

    public double getPuntuacionTotal() {
        return this.puntuacionTotal;
    }

    public void setPuntuacionTotal(double value) {
        this.puntuacionTotal = value;
    }
}

```

```

    public void agregarEjercicioCalificado(String nombreEjercicio, double puntuacion) {
        Ejercicio nuevoEjercicio = new Ejercicio();
        nuevoEjercicio.setNombre(nombreEjercicio);
        nuevoEjercicio.setPuntuacion(puntuacion);
        ejercicios.add(nuevoEjercicio);
    }

    public String toString() {
        return this.nombre + " " + this.puntuacionTotal + " " + (apta == true ? "apta" : "no apta");
    }
}

```

Coverage de Actividad.

Ejercicio 8)

```

import java.util.ArrayList;
public class Ferry {
    private int maxVehiculos;
    private int maxPasajeros;
    private double maxPeso;
    private int totalPasajeros;
    private int totalVehiculos;
    private double totalPeso;
    private ArrayList<Vehiculo> listaVehiculos = new ArrayList<Vehiculo>();

    public void setMaxVehiculos(int maxVehiculos) {
        this.maxVehiculos = maxVehiculos;
    }
    public void setMaxPeso(double peso) {
        this.maxPeso = peso;
    }
    public boolean embarcarVehiculo(Vehiculo v) {
        // Embarca un vehículo añadiéndolo a la lista
        if (v.getPeso() == 0.0)
            return false;
        listaVehiculos.add(v);
        totalVehiculos++;
        totalPasajeros += v.getPasajeros();
        totalPeso += v.getPeso();
        return true;
    }
    public int totalVehiculos() { // Devuelve el número total de vehículos embarcados
        return totalVehiculos;
    }
    public boolean vacio() { // Verdadero si no hay ningún vehículo
        return listaVehiculos.isEmpty();
    }
    public boolean superadoMaximoVehiculos() { // Verdadero si el número total de vehículos supera el máximo
        if (totalVehiculos > maxVehiculos)
            return true;
        return false;
    }
    public boolean superadoMaximoPeso() { // Verdadero si el peso total de los vehículos supera el máximo
        if (totalPeso > maxPeso)
            return true;
        return false;
    }
}

```

Coverage de Ferry.


```

package ual.hmis.sesion06;

public class Vehiculo {
    private int numPasajeros;
    private int numRuedas;
    private double peso;

    public int getPasajeros() {
        return numPasajeros;
    }
    public void setPasajeros(int pasajeros) {
        this.numPasajeros = pasajeros;
    }
    public double getPeso() {
        return peso;
    }
    public void setPeso(double peso) {
        this.peso = peso;
    }
}

```

Coverage de Vehiculo.