UNIVERSIDADE DA CORUÑA

Facultad de Informática
4º Grado en Ingeniería Informática. Especialidad en Ingeniería de Software
Validación y Verificación del Software
2016-2017

REPOSITORIO DEL PROXECTO:
https://github.com/AlejandroPeralTaboada/VVS

PARTICIPANTES EN EL PROXECTO:

- **Sergio Costa García**
- **Alejandro Peral Taboada**

VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE

INFORME DE PRÁCTICAS

Contenido

1. DESCRIPCIÓN DEL PROYECTO

En un moderno centro deportivo acuático orientado tanto al ocio como al deporte existen varias piscinas, dotadas de sensores electrónicos que miden diferentes parámetros del agua. Con esta aplicación se quiso gestionar la correcta gestión de las piscinas del centro, el mantenimiento de estas y la seguridad de los bañistas.

- Piscinas

Nuestra aplicación necesita reflejar el estado actual de las piscinas para poder ser consultado por el personal. Cada piscina tendrá un nombre, una ubicación y los parámetros medibles por los sensores. A mayores existirá una piscina dedicada exclusivamente al relax, en la cual es importante conocer el nivel de sales minerales del agua.

- Personas

Existe personal de cinco categorías: socorristas, encargados, mantenimiento, marketing y gestores. El personal de cada categoría realiza diversas tareas sobre las piscinas por lo que tienen que estar al tanto de lo que suceda con ellas. El personal se puede organizar en equipos y subequipos.

- Alarmas

Para tener informado al personal adecuado de qué sucede con las piscinas, se implementó un sistema de alarmas. Existen alarmas para controlar cambios en cada uno de los parámetros mensurables en las piscinas. Si un determinado parámetro sube por encima de un umbral o baja por debajo de otro, se mande un aviso al personal encargado de la piscina en cuestión.

- Fases de la piscina
  - Piscina activa: La piscina está en su régimen de funcionamiento normal y abierta al público.
  - Piscina en mantenimiento: Se están aplicado operaciones de mantenimiento a la piscina y por tanto no puede estar siendo utilizada por nadie.
  - Piscina cerrada: La piscina no está abierta al público.
  - Piscina en evacuación: Debido a algún tipo de alarma se ha puesto la piscina en fase de evacuación (durar unos minutos) para que la gente que se encuentra en ella salga para que puedan ser tomadas las medidas adecuadas.
- Registro e informes de las piscinas

El personal de categoría gestor debe extraer un informe de las piscinas. Para ello es necesario, para cada una de las piscinas, registrar cada evento que suceda. Los eventos pueden ser de dos tipos:

- Cambio en un parámetro.
- Cambio en la fase de una piscina.

- Prueba global

Además de las pruebas de unidad de cada una de las clases, se realizó una prueba global que permite comprobar el funcionamiento de la aplicación completa.

2. ESTADO ACTUAL

Los estados de las pruebas al comienzo del proyecto eran nulos, a día de hoy el proyecto se encuentra con todos sus componentes probados y con altos porcentajes de cobertura.

2.1. COMPONENTES EXAMINADOS

Resumen general de todos los componentes y las pruebas

- En pruebas de unidad:

- **Registro.** Pruebas objetivo 18, valores frontera para el número de registros máximo almacenado y para la generación de los informes.
- **Estados.** Pruebas objetivo 18 más el uso de un graphwalker. Pruebas en la transición de los estados.
- **Sensor.** Pruebas objetivo 12, valores frontera para los datos medidos y las piscinas asignadas.
- **Alarma.** Pruebas objetivo 69, valores frontera con los objetivos de las distintas alarmas.
- **PiscinaImp.** Pruebas objetivo 28, valores frontera con los distintos componentes usando mocks.
- **Plantilla.** Pruebas objetivo 4, valores frontera.

- En pruebas de integración:

- **Registro con piscina.** Pruebas objetivo 8, pruebas basadas en procedimientos para ver el comportamiento de ambos componentes.
- **Sensor con piscina.** Pruebas objetivo 12, pruebas basadas en procedimientos para ver el comportamiento de ambos componentes.
- **Piscina con alarma y plantilla**. Pruebas objetivo 12, pruebas basadas en procedimientos para ver el comportamiento de ambos componentes.

3. ESPECIFICACIÓN DE PRUEBAS

- **PRUEBAS DE UNIDAD:** Antes de contemplar las pruebas de unidad, comentamos brevemente las herramientas utilizadas para desarrollar dichas pruebas.

  - **JUnit:** JUnit es un framework que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún

valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces Junit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

- **Mockito:** Mockito es un framework para pruebas de aplicaciones Java. Permite simular comportamientos complejos de objetos, además facilita el desarrollo de test unitarios y l detección de errores. Mockito se basa en el principio de un objeto mock que simula el comportamiento de otro objeto.

PRUEBAS DE UNIDAD: REGISTRO						
Nombre	**Clase**	**Método**	**Tipo**	**Entrada**	**Salida**	**Inicial**
pr_UN_Registro_001	Registro	addSuceso	Frontera	0	0	-
pr_UN_Registro_002	Registro	addSuceso	Frontera	-1	0	-
pr_UN_Registro_003	Registro	addSuceso	Frontera	1	1	-
pr_UN_Registro_004	Registro	addSuceso	Frontera	10	10	-
pr_UN_Registro_005	Registro	setNumReg	Frontera	0	0	0
pr_UN_Registro_006	Registro	setNumReg	Frontera	0	0	-1
pr_UN_Registro_007	Registro	setNumReg	Frontera	0	0	10
pr_UN_Registro_008	Registro	setNumReg	Frontera	-1	0	0
pr_UN_Registro_009	Registro	setNumReg	Frontera	-1	0	-1
pr_UN_Registro_010	Registro	setNumReg	Frontera	-1	0	1
pr_UN_Registro_011	Registro	setNumReg	Frontera	1	1	0
pr_UN_Registro_012	Registro	setNumReg	Frontera	1	1	-1
pr_UN_Registro_013	Registro	setNumReg	Frontera	1	1	1
pr_UN_Registro_014	Registro	setNumReg	Frontera	2	2	1
pr_UN_Registro_015	Registro	Informe	Frontera	-	“”	0
pr_UN_Registro_016	Registro	Informe	Frontera	-	“Suceso1”	1
pr_UN_Registro_017	Registro	Informe	Frontera	-	“Suceso 1 Suceso 1”	2
pr_UN_Registro_018	Registro	Informe	Frontera	-	“”	0

PRUEBAS DE UNIDAD: ESTADOS						
Nombre	**Clase**	**Método**	**Tipo**	**Entrada**	**Salida**	**Inicial**
pr_UN_Estados_001	Piscina	activar	Estado	Activa	Activa	-
pr_UN_Estados_002	Piscina	cerrar	Estado	Activa	Activa	-
pr_UN_Estados_003	Piscina	mantenimiento	Estado	Activa	Activa	-
pr_UN_Estados_004	Piscina	evacuar	Estado	Activa	Evacuación	-
pr_UN_Estados_005	Piscina	activar	Estado	Cerrada	Cerrada	-
pr_UN_Estados_006	Piscina	cerrar	Estado	Cerrada	Cerrada	-
pr_UN_Estados_007	Piscina	mantenimiento	Estado	Cerrada	Mantenim.	-
pr_UN_Estados_008	Piscina	evacuar	Estado	Cerrada	Cerrada	-
pr_UN_Estados_009	Piscina	activar	Estado	Mantenimiento	Activa	-
pr_UN_Estados_010	Piscina	cerrar	Estado	Mantenimiento	Cerrada	-
pr_UN_Estados_011	Piscina	mantenimiento	Estado	Mantenimiento	Mantenim.	-
pr_UN_Estados_012	Piscina	evacuar	Estado	Mantenimiento	Mantenim.	-
pr_UN_Estados_013	Piscina	activar	Estado	Evacuación	Evacuación	-
pr_UN_Estados_014	Piscina	activar	Estado	Evacuación 10 personas	Evacuación	-

pr_UN_Estados_015	Piscina	evacuar	Estado	Evacuación 0 personas	Mantenimi.	-
pr_UN_Estados_016	Piscina	evacuar	Estado	Evacuación 10 personas	0 personas Mantenimi.	-
pr_UN_Estados_017	Piscina	cerrar	Estado	Evacuación	Evacuación	-
pr_UN_Estados_018	Piscina	mantenimiento	Estado	Evacuación	Evacuación	-

PRUEBAS DE UNIDAD: SENSOR						
Nombre	**Clase**	**Método**	**Tipo**	**Entrada**	**Salida**	**Inicial**
pr_UN_Sensor_001	sensorAgua	medirAgua	Frontera	10	10	0
pr_UN_Sensor_002	sensorCloro	medirCloro	Frontera	10	10	0
pr_UN_Sensor_003	sensorPH	medirPH	Frontera	10	10	0
pr_UN_Sensor_004	sensorSales	medirSales	Frontera	10	10	0
pr_UN_Sensor_005	sensorPers.	medirPers.	Frontera	10	10	0
pr_UN_Sensor_006	sensorTemp.	medirTemp.	Frontera	10	10	0
pr_UN_Sensor_007	sensorAgua	getPiscina	Frontera	piscina	piscina	-
pr_UN_Sensor_008	sensorCloro	getPiscina	Frontera	piscina	piscina	-
pr_UN_Sensor_009	sensorPH	getPiscina	Frontera	piscina	piscina	-
pr_UN_Sensor_010	sensorSales	getPiscina	Frontera	piscinaRelax	piscinaRelax	-
pr_UN_Sensor_011	sensorPers.	getPiscina	Frontera	piscina	piscina	-
pr_UN_Sensor_012	sensorTemp.	getPiscina	Frontera	piscina	piscina	-

PRUEBAS DE UNIDAD: ALARMA						
Nombre	**Clase**	**Método**	**Tipo**	**Entrada**	**Salida**	**Inicial**
pr_UN_Alarma_001	nivelAgua	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_002	nivelAgua	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_003	nivelAgua	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_004	nivelAgua	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_005	nivelAgua	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_006	nivelAgua	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_007	nivelAgua	addRespons.	Frontera	empleado	empleado	-
pr_UN_Alarma_008	nivelCloro	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_009	nivelCloro	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_010	nivelCloro	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_011	nivelCloro	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_012	nivelCloro	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_013	nivelCloro	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_014	nivelCloro	addRespons.	Frontera	empleado	empleado	-
pr_UN_Alarma_015	nivelCloroEv	addRespons.	Frontera	equipo	equipo	-

pr_UN_Alarma_016	nivelCloroEv	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_017	nivelCloroEv	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_018	nivelCloroEv	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_019	nivelCloroEv	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_020	nivelCloroEv	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_021	nivelCloroEv	addRespons.	Frontera	empleado	empleado	-
pr_UN_Alarma_022	nivelPH	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_023	nivelPH	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_024	nivelPH	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_025	nivelPH	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_026	nivelPH	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_027	nivelPH	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_028	nivelPH	addRespons.	Frontera	empleado	empleado	-
pr_UN_Alarma_029	nivelSales	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_030	nivelSales	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_031	nivelSales	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_032	nivelSales	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_033	nivelSales	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_034	nivelSales	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_035	nivelSales	addRespons.	Frontera	empleado	empleado	-
pr_UN_Alarma_036	nivelPers.	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_037	nivelPers.	addRespons.	Frontera	equipo	excepción	-
pr_UN_Alarma_038	nivelPers.	addRespons.	Frontera	equipo	equipo	-
pr_UN_Alarma_039	nivelPers.	addRespons.	Frontera	empleado	excepción	-
pr_UN_Alarma_040	nivelPers.	addRespons.	Frontera	empleado	excepción	-
pr_UN_Alarma_041	nivelPers.	addRespons.	Frontera	empleado	excepción	-
pr_UN_Alarma_042	nivelTemp.	addRespons.	Frontera	equipo	empleado	-
pr_UN_Alarma_043	nivelTemp.	addRespons.	Frontera	-	equipo	-
pr_UN_Alarma_044	nivelTemp.	addRespons.	Frontera	equipo	excepción	-
pr_UN_Alarma_045	nivelTemp.	addRespons.	Frontera	-	equipo	-
pr_UN_Alarma_046	nivelTemp.	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_047	nivelTemp.	addRespons.	Frontera	-	excepción	-
pr_UN_Alarma_048	nivelTemp.	addRespons.	Frontera	empleado	excepción	-
pr_UN_Alarma_049	nivelAgua	setPlantilla	Frontera	plantilla	empleado	-
pr_UN_Alarma_050	nivelAgua	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_051	nivelAgua	informe	Formato	-	-	“”
pr_UN_Alarma_052	nivelCloro	setPlantilla	Frontera	plantilla	plantilla	-

pr_UN_Alarma_053	nivelCloro	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_054	nivelCloro	informe	Formato	-	-	“”
pr_UN_Alarma_055	nivelCloroEv	setPlantilla	Frontera	plantilla	plantilla	-
pr_UN_Alarma_056	nivelCloroEv	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_057	nivelCloroEv	Informe	Formato	-	-	“”
pr_UN_Alarma_058	nivelPH	setPlantilla	Frontera	plantilla	plantilla	-
pr_UN_Alarma_059	nivelPH	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_060	nivelPH	informe	Formato	-	-	“”
pr_UN_Alarma_061	nivelSales	setPlantilla	Frontera	plantilla	plantilla	-
pr_UN_Alarma_062	nivelSales	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_063	nivelSales	informe	Formato	-	-	“”
pr_UN_Alarma_064	nivelPers.	setPlantilla	Frontera	plantilla	plantilla	-
pr_UN_Alarma_065	nivelPers.	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_066	nivelPers.	informe	Formato	-	-	“”
pr_UN_Alarma_067	nivelTemp.	setPlantilla	Frontera	plantilla	plantilla	-
pr_UN_Alarma_068	nivelTemp.	setRegistro	Frontera	registro	registro	-
pr_UN_Alarma_069	nivelTemp.	informe	Formato	-	-	“”

PRUEBAS DE UNIDAD: PISCINA IMP						
Nombre	**Clase**	**Método**	**Tipo**	**Entrada**	**Salida**	**Inicial**
pr_UN_Piscina_001	Piscina	setNivelAgua	Frontera	0	-	0
pr_UN_Piscina_002	Piscina	setNivelAgua	Frontera	-1	Excepción	0
pr_UN_Piscina_003	Piscina	setNivelAgua	Frontera	10	-	0
pr_UN_Piscina_004	Piscina	setNivelAgua	Alea. +	Random	-	0
pr_UN_Piscina_005	Piscina	setTemp.	Frontera	0	-	0
pr_UN_Piscina_006	Piscina	setTemp.	Frontera	-273,15	-	0
pr_UN_Piscina_007	Piscina	setTemp.	Frontera	-273,16	Excepción	0
pr_UN_Piscina_008	Piscina	setTemp.	Alea. R.	Random	-	0
pr_UN_Piscina_009	Piscina	setNivelCloro	Frontera	10	-	0
pr_UN_Piscina_010	Piscina	setNivelCloro	Frontera	0	-	0
pr_UN_Piscina_011	Piscina	setNivelCloro	Frontera	-0.01	Excepción	0
pr_UN_Piscina_012	Piscina	setNivelCloro	Frontera	100	-	0
pr_UN_Piscina_013	Piscina	setNivelCloro	Frontera	100.01	Excepción	0
pr_UN_Piscina_014	Piscina	setNivelCloro	Alea. R.	Random	-	0
pr_UN_Piscina_015	Piscina	setNivelPH	Frontera	10	-	0
pr_UN_Piscina_016	Piscina	setNivelPH	Frontera	0	-	0
pr_UN_Piscina_017	Piscina	setNivelPH	Frontera	-0.01	Excepción	0

pr_UN_Piscina_018	Piscina	setNivelPH	Frontera	14	-	0
pr_UN_Piscina_019	Piscina	setNivelPH	Frontera	14.01	Excepción	0
pr_UN_Piscina_020	Piscina	setNivelPH	Alea. R.	Random	-	0
pr_UN_Piscina_021	Piscina	setPersonas	Frontera	0	-	0
pr_UN_Piscina_022	Piscina	setPersonas	Frontera	-1	Excepción	0
pr_UN_Piscina_023	Piscina	setPersonas	Frontera	10	-	0
pr_UN_Piscina_024	Piscina	setPersonas	Alea. +	Random	-	0
pr_UN_Piscina_025	Piscina	setPersonas	Frontera	Sensor pisc	Igual	0
pr_UN_Piscina_026	Piscina	setPersonas	Frontera	Sensor pisc	Distinta	0
pr_UN_Piscina_027	Piscina	setPersonas	Frontera	Sensor pisc =	Excepción	1
pr_UN_Piscina_028	Piscina	informe	Formato	-	“Informe..”	-

PRUEBAS DE UNIDAD: PLANTILLA						
Nombre	**Clase**	**Método**	**Tipo**	**Entrada**	**Salida**	**Inicial**
pr_UN_Plantilla_001	Plantillla	addEmpleado	Frontera	empleados	empleados	-
pr_UN_Plantilla_002	Plantilla	addEquipo	Frontera	equipos	equipos	-
pr_UN_Plantilla_003	Plantilla	setEmpleados	Frontera	empleados	empleados	-
pr_UN_Plantilla_004	Plantilla	setEquipos	Frontera	equipos	equipos	-

- **PRUEBAS DE INTEGRACIÓN:**

PRUEBAS DE INTEGRACIÓN: PISCINA - REGISTRO			
Nombre	**Clase**	**Tipo**	**Descripción**
pr_IN_PiscinaReg_001	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al modificar un parámetro no se deben registrar los cambios.
pr_IN_PiscinaReg_002	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al activarla y cambiar el nivel de Agua debemos obtener 2 registros.
pr_IN_PiscinaReg_003	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al activarla y cambiar el nivel de Cloro debemos obtener 2 registros.
pr_IN_PiscinaReg_004	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al activarla y cambiar el nivel de PH debemos obtener 2 registros.
pr_IN_PiscinaReg_005	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al activarla y cambiar el número de personas debemos obtener 2 registros.
pr_IN_PiscinaReg_006	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al activarla y cambiar el nivel de sales debemos obtener 2 registros.
pr_IN_PiscinaReg_007	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al cerrarla y cambiar el nivel de sales debemos obtener 0 registros

pr_IN_PiscinaReg_008	Piscina y Registro	Basadas en procedimientos	Para una piscina en estado inicial mantenimiento al activarla y añadirle personas, evacuarla y cambiarle un parámetro debemos obtener 4 registros.

PRUEBAS DE INTEGRACIÓN: PISCINA - SENSOR			
Nombre	**Clase**	**Tipo**	**Descripción**
pr_IN_PiscinasSensor_001	Piscina y Sensor	Basadas en procedimientos	Modificar el nivel del agua y comprobar que se actualizó en la piscina.
pr_IN_PiscinasSensor_002	Piscina y Sensor	Basadas en procedimientos	Modificar el nivel del cloro y comprobar que se actualizó en la piscina.
pr_IN_PiscinasSensor_003	Piscina y Sensor	Basadas en procedimientos	Modificar el nivel del PH y comprobar que se actualizó en la piscina.
pr_IN_PiscinasSensor_004	Piscina y Sensor	Basadas en procedimientos	Modificar el nivel de sales y comprobar que se actualizó en la piscina relax.
pr_IN_PiscinasSensor_005	Piscina y Sensor	Basadas en procedimientos	Modificar la cantidad de personas y comprobar que se actualizó en la piscina.
pr_IN_PiscinasSensor_006	Piscina y Sensor	Basadas en procedimientos	Modificar la temperatura y comprobar que se actualizó en la piscina.
pr_IN_PiscinasSensor_007	Piscina y Sensor	Basadas en procedimientos	Comprobar que el objeto piscina es el mismo al utilizado por el sensorAgua.
pr_IN_PiscinasSensor_008	Piscina y Sensor	Basadas en procedimientos	Comprobar que el objeto piscina es el mismo al utilizado por el sensorCloro.
pr_IN_PiscinasSensor_009	Piscina y Sensor	Basadas en procedimientos	Comprobar que el objeto piscina es el mismo al utilizado por el sensorPh.
pr_IN_PiscinasSensor_010	Piscina y Sensor	Basadas en procedimientos	Comprobar que el objeto piscinaRelax es el mismo al utilizado por el sensorSales.
pr_IN_PiscinasSensor_011	Piscina y Sensor	Basadas en procedimientos	Comprobar que el objeto piscina es el mismo al utilizado por el sensorPersonas.
pr_IN_PiscinasSensor_012	Piscina y Sensor	Basadas en procedimientos	Comprobar que el objeto piscina es el mismo al utilizado por el sensorTemperatura.

PRUEBAS DE INTEGRACIÓN: PISCINA – ALARMA - PLANTILLA			
Nombre	**Clase**	**Tipo**	**Descripción**
pr_IN_PiscAlarmPlantill_001	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por encima de esa, se avisó al Socorrista.
pr_IN_PiscAlarmPlantilla_002	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por debajo de esa, no se avisó al Socorrista.
pr_IN_PiscAlarmPlantill_003	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por encima de esa, se avisó al Encargado.
pr_IN_PiscAlarmPlantill_004	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por debajo de esa, no se avisó al Encargado..
pr_IN_PiscAlarmPlantill_005	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por encima de esa, se avisó al Mantenimiento.
pr_IN_PiscAlarmPlantill_006	Piscina, Alarma,	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por debajo de

	Plantilla		esa, no se avisó al Mantenimiento.
pr_IN_PiscAlarmPlantill_007	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por encima de esa, se avisó al Marketing.
pr_IN_PiscAlarmPlantill_008	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por debajo de esa, no se avisó al Marketing.
pr_IN_PiscAlarmPlantill_009	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por encima de esa, se avisó al Gestor.
pr_IN_PiscAlarmPlantill_010	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por debajo de esa, no se avisó al Gestor.
pr_IN_PiscAlarmPlantill_011	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por encima de esa, se avisó al Equipo.
pr_IN_PiscAlarmPlantill_012	Piscina, Alarma, Plantilla	Basadas en procedimientos	Asignar un máximo de la alarma del agua, y comprobar que fijando una cantidad por debajo de esa, no se avisó al Equipo.

- **PRUEBAS DE SISTEMA**

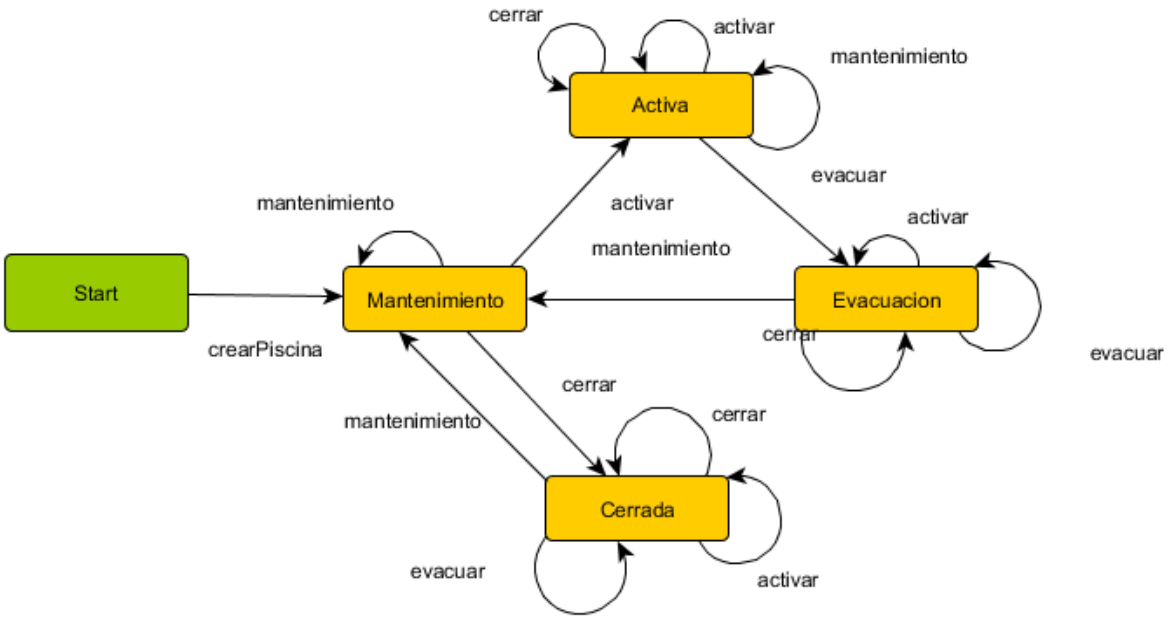
Hemos realizado una prueba que engloba el uso de todos los componentes en una sola ejecución para comprobar su correcto funcionamiento.

- **PUEBAS CON SELECCIÓN DE DATOS ALEATORIOS**

En este caso, en lugar de emplear una herramienta como Quickcheck, hemos realizado a mano generadores de datos random para las pruebas de unidad.

- **PUEBAS BASADAS EN MODELOS**

Debido a que tenemos un patrón estado en nuestra aplicación hemos empleado Graphwalker para realizar una prueba basada en el diagrama de estados correspondiente a la aplicación. Para ello hemos realizado en Yed el diagrama correspondiente y hemos ejecutado pruebas de timeout y de cobertura aleatoria de líneas.



- **ANÁLISIS ESTÁTICA DE CAJA BLANCA:**

CHECKSTYLE. Al principio contábamos con un total de 4041 errores en el código (cuya cuantía resaltamos con una issue en el repositorio de GitHub). Fuimos revisando clase por clase, y solucionando los errores (la mayoría de formato), hasta llegar a la cuantía de 0 errores en nuestro código de este tipo. Es altamente configurable y se puede hacer para apoyar casi cualquier estándar de codificación. De tal manera que se puedan suministrar diferentes estándares de código para su posterior comprobación mediante la herramienta.

SONARQUBE: Aprovechando que teníamos instalado el complemento de SonarQube para el desarrollo de otra asignatura, hemos ido solucionando en la medida de lo posible, la mayoría de errores que detecta dicha herramienta. Los más comunes que no hemos podido solucionar son los relacionados con la impresión por pantalla de texto.

- **MUTACIÓN DE CÓDIGO:**

Para estas pruebas utilizamos la herramienta PiTest (existente en el marketPlace de eclipse). A continuación, mostramos unas imágenes de como ha quedado, en resumen, la cobertura del código mediante las ejecuciones de los reportes.

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage	
38	89%	601/676	48%	150/315

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage	
vvs.alarma	8	71%	134/188	49%	33/67
vvs.piscinas	7	96%	175/183	76%	81/106
vvs.plantilla	8	94%	65/69	24%	6/25
vvs.pruebafinal	1	99%	105/106	0%	0/75
vvs.registro	8	91%	73/80	59%	17/29
vvs.sensor	6	98%	49/50	100%	13/13

Report generated by PIT 1.1.9

Pit Test Coverage Report

Package Summary

vvs.piscinas

Number of Classes	Line Coverage		Mutation Coverage	
7	96%	175/183	76%	81/106

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Activa.java	100%	19/19	64%	7/11
Cerrada.java	89%	17/19	55%	6/11
Evacuacion.java	90%	19/21	54%	7/13
Mantenimiento.java	100%	20/20	69%	9/13
Piscina.java	85%	11/13	100%	10/10
PiscinaImp.java	100%	77/77	93%	37/40
PiscinaRelax.java	86%	12/14	63%	5/8

Report generated by PIT 1.1.9

Pit Test Coverage Report

Package Summary

vvs.alarma

Number of Classes	Line Coverage		Mutation Coverage	
8	71%	134/188	49%	33/67

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Alarma.java	100%	20/20	86%	6/7
AlarmaNivelAgua.java	100%	24/24	78%	7/9
AlarmaNivelCloro.java	77%	20/26	44%	4/9
AlarmaNivelCloroEvacuacion.java	73%	19/26	36%	4/11
AlarmaNivelPh.java	25%	6/24	44%	4/9
AlarmaNivelSales.java	100%	24/24	44%	4/9
AlarmaPersonas.java	75%	15/20	0%	0/4
AlarmaTemperatura.java	25%	6/24	44%	4/9

Report generated by PIT 1.1.9

Pit Test Coverage Report

Package Summary

vvs.plantilla

Number of Classes	Line Coverage		Mutation Coverage	
8	94%	65/69	24%	6/25

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Empleado.java	100%	8/8	0%	0/4
Encargado.java	64%	7/11	0%	0/5
Equipo.java	100%	13/13	0%	0/4
Gestor.java	100%	4/4	0%	0/2
Mantenimiento.java	100%	3/3	0%	0/1
Marketing.java	100%	3/3	0%	0/1
Plantilla.java	100%	24/24	86%	6/7
Socorrista.java	100%	3/3	0%	0/1

Report generated by PIT 1.1.9

Pit Test Coverage Report

Package Summary

vvs.sensor

Number of Classes	Line Coverage		Mutation Coverage	
6	98%	49/50	100%	13/13

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
SensorNivelAgua.java	90%	9/10	100%	3/3
SensorNivelCloro.java	100%	8/8	100%	2/2
SensorNivelPh.java	100%	8/8	100%	2/2
SensorNivelSales.java	100%	8/8	100%	2/2
SensorPersonas.java	100%	8/8	100%	2/2
SensorTemperatura.java	100%	8/8	100%	2/2

Report generated by PIT 1.1.9

- **COBERTURA DE CÓDIGO:**

Para realizar la cobertura de las clases proporcionadas por los test hemos empleado un plugin de maven que realiza la cobertura cuando se ejecuta la meta site. Nos establecimos un límite de cobertura no inferior al 85%. Los resultados del código son los siguientes:

Package	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	42	88% 609/686	81% 113/138	1,473
vvs.alarma	8	71% 134/188	75% 51/68	3,043
vvs.piscinas	8	96% 180/187	90% 36/40	1,313
vvs.plantilla	9	93% 69/74	80% 8/10	1,188
vvs.pruebafinal	1	98% 104/106	N/A N/A	2
vvs.registro	9	91% 73/80	94% 17/18	1,409
vvs.sensor	7	96% 49/51	50% 1/2	1,077

Puesto que dicha información se puede generar mediante mvn site y se puede consultar en ella de forma más eficiente no hemos añadido más imágenes. Destacar que la cobertura inicial del sistema era 0 y que actualmente sobrepasa el 85% que era nuestro objetivo inicial.

- **DOCUMENTACIÓN DE CÓDIGO:**

Todas las clases han sido documentadas con su javadoc correspondiente detallando parámetros de entrada, salida y excepciones lanzadas.

- **JAVA MISSION CONTROL:**

Realizamos en la herramienta proporcionado por el jdk, un control del rendimiento a la hora de la ejecución del programa. Dejamos unas imágenes, para la visualización de los resultados:

Como vemos en la foto, hemos medido el uso del procesador a la hora de ejecutar los tests (cuya prueba final, implica la creación de 50 piscinas y unos cuantos sleep). También medimos la memoria que necesita el programa para la ejecución, vemos que aumenta considerablemente según se ejecutan y avanza.

Con este test podemos ver como el sistema tiene una evolución lineal en el uso de memoria lo que nos permite garantizar escalabilidad. Del mismo modo podemos estimar que el gasto de memoria por cada piscina ronda los 80 kb.

4. REGISTRO DE PRUEBAS

Para la realización de las pruebas se ha seguido la siguiente aproximación:
Puesto que el software ya estaba desarrollado la especificación de las pruebas se ha realizado una vez realizado el software. El conjunto de pruebas más importante está basado en las pruebas de unidad, así como la comprobación de sus errores con test de mutación y test para verificar las máquinas de estado.
Para realizar una prueba la hemos detallado partiendo de la especificación original de la práctica sin contemplar el código con el objetivo de no estar influenciados por él. Una vez detallada la prueba, la mayor parte de las veces pruebas de valor frontera, se ha implementado y realizado la prueba. En caso de error esta ha sido registrada como una issue en el scm del proyecto. En caso de éxito de las pruebas desarrolladas se suben al scm donde una herramienta de integración continua, en este caso, travis, vuelve a ejecutar las pruebas en su propia máquina para verificar su correcto funcionamiento. En caso de error se sigue el mismo procedimiento que en caso de hallar un fallo en local.
Una vez detectado un error, se registra como una issue en github, se le asigna a un miembro y cuando este se ha solventado y pasa las pruebas tanto en local como en integración se da la issue como resuelta.
Destacar que en cuanto al tipo de pruebas de rendimiento el programa está orientado más a ser una librería, debido a esto no tenemos un conjunto de usuarios realizando peticiones, por ello nos hemos centrado más en el espacio en memoria que requiere el sistema.

5. REGISTRO DE ERRORES

A continuación citamos en una tabla los problemas que hemos ido solucionando (todos marcados como issues en el repositorio nombrado en la portada):

Nombre	Descripción	Asignada	Etiqueta	Inicio	Cerrada
Registro_002	Para el valor -1 del registro se espera salida de la lista con tamaño 0 y el valor máximo 0. Obtenemos 0 y -1.	Alejandro	Bug	13 oct	27 oct
Registro_008	Para el valor inicial de 0 y un set de -1 obtenemos un -1 como tamaño y 0 como size, ambos deberían ser 0.	Alejandro	Bug	16 oct	27 oct
Registro_009	Valor inicial de -1 y un set de -1 obtenemos un -1 como tamaño y 0 como size. Ambos deberían ser 0.	Alejandro	Bug	16 oct	27 oct
Registro_010	Para el valor inicial de 0 y un set de -1 obtenemos un -1 como tamaño y 0 como size, ambos deberían ser 0.	Alejandro	Bug	16 oct	27 oct

Registro_016	Para el valor inicial de 2 sucesos con nombre "suceso1" y un informe obtenemos los dos juntos sin espacio en medio.	Alejandro	Bug	16 oct	27 oct
Estados_006	Una piscina cerrada pasa al estado mantenimiento cuando se activa el método mantenimiento. Está pasando a evacuación.	Costris	Bug	18 oct	27 oct
Estados_015	Una piscina que está activa con 0 personas, al evacuarse ha de pasar a estado mantenimiento directamente.	Alejandro	Bug	18 oct	27 oct
Piscina_006	La temperatura puede ser negativa hasta -273,15 teóricamente, no permite negativos el sistema, es posible que una piscina de sales este a temperaturas bajo 0.	Alejandro	Bug	18 oct	27 oct
Piscina_013	El cloro es un porcentaje por lo tanto va desde 0.00 hasta 100.00, para 100.01 no salta excepción.	Alejandro	Bug	18 oct	29 oct
Piscina_019	El PH tiene un rango desde 0.00 hasta 14.00, para 14.01 no salta excepción.	Costris	Bug	18 oct	29 oct
Piscina_026	Cuando se añade un sensor cuya piscina no es la misma a la que se añade falla, en principio dicho sensor debería ser inicializado con la piscina asignada.	Alejandro	Bug	18 oct	29 oct
CheckStyle 4041 warnings	Error de checkstyle de 4041 warning bajado a 319 formateando de forma automática todo el proyecto resto, keep in progress.	-	Enhance ment	20 oct	23 oct
CheckStyle 0 warnings	Solucionados todos los warnings de checkstyle.	-	Invalid	21 oct	27 oct
Travis-CI falla	A partir de ahora, el único motivo aceptable para que las build de Travis fallen debería ser que estén fallando casos de prueba (es decir, que haya bugs detectados aún no resueltos). Vuestro repositorio ahora mismo non se construye porque falla la compilación de las pruebas. Procurad que esto no ocurra.	Costris	Wontfix	22 oct	27 oct
Excepciones	La manera adecuada de asegurarse en un caso de prueba de que se lanza una excepción (prueba negativa) es usar el assert en la parte del catch de la estructura try-catch que habéis empleado, o alternativamente usar el tag @Test (expected = ExcepcionConcreta.class). Modificad las pruebas para corregirlo.	Costris	Wontfix	22 oct	24 oct
GraphWalker	Error del garphwalker al realizar mvn test	Alejandro	Wontfix	1 dic	10 dic

Por último, añadimos un enlace al repositorio de gitHub, donde almacenamos las issues de nuestro proyecto (enlace a las issues cerradas):
https://github.com/AlejandroPeralTaboada/VVS/issues?q=is%3Aissue+is%3Aclosed

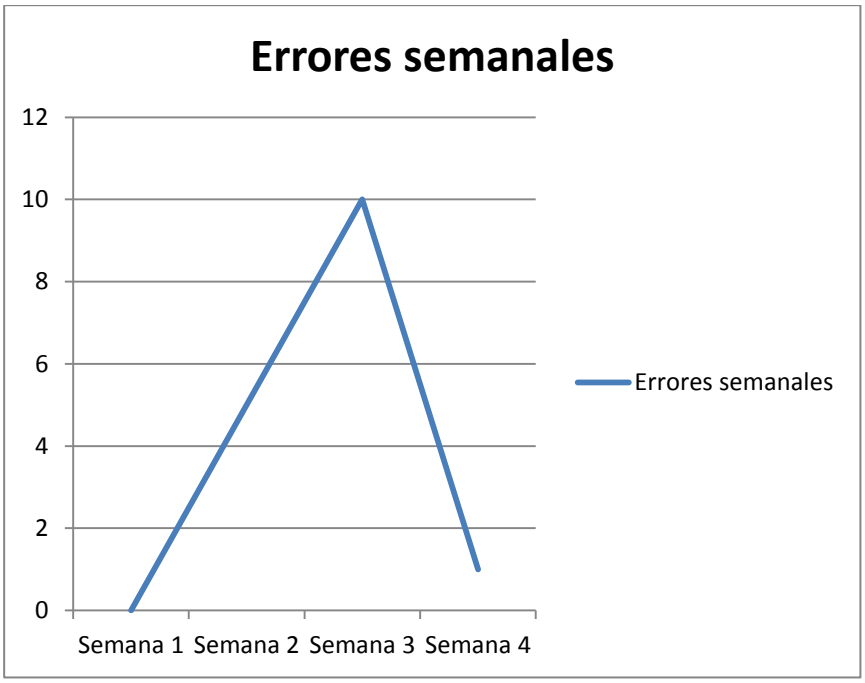
6. ESTADÍSTICAS

Mostramos las estadísticas de nuestro proyecto en diferentes apartados, serán los siguientes:

- **Número de errores encontrados diariamente y semanalmente.**
  - **Diariamente:**

DÍA	ERRORES
13-10-2016	1
16-10-2016	4
18-10-2016	6
20-10-2016	1
21-10-2016	1
22-10-2016	2
01-12-2016	1
TOTALES:	16

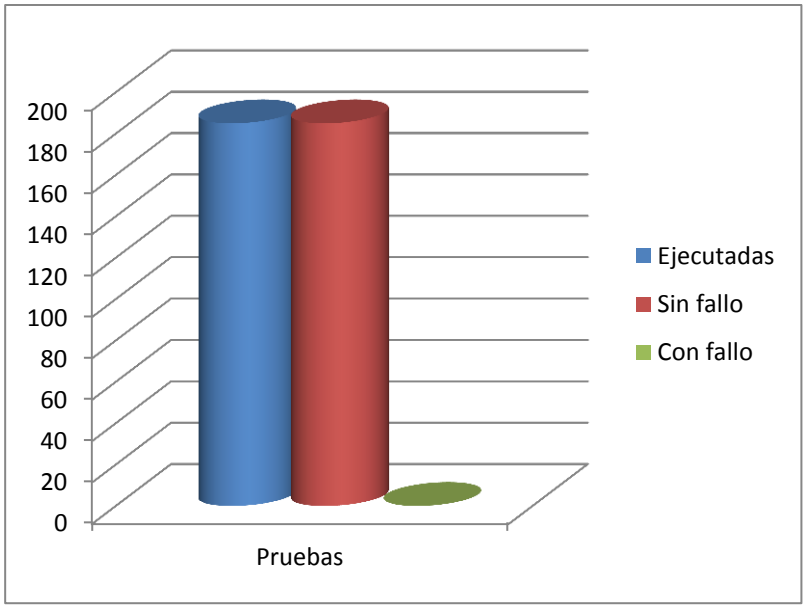
- **Semanalmente:**



SEMANA	ERRORES
Semana 1	0
Semana 2	5
Semana 3	10
Semana 4	1
TOTAL	16

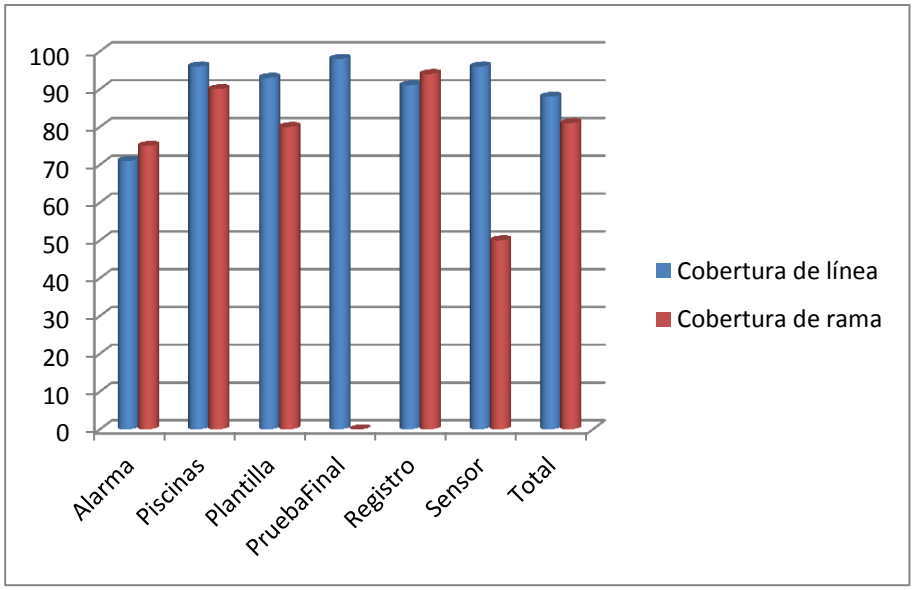
Nivel de progreso en la ejecución de las pruebas.

- **% Ejecución Pruebas**



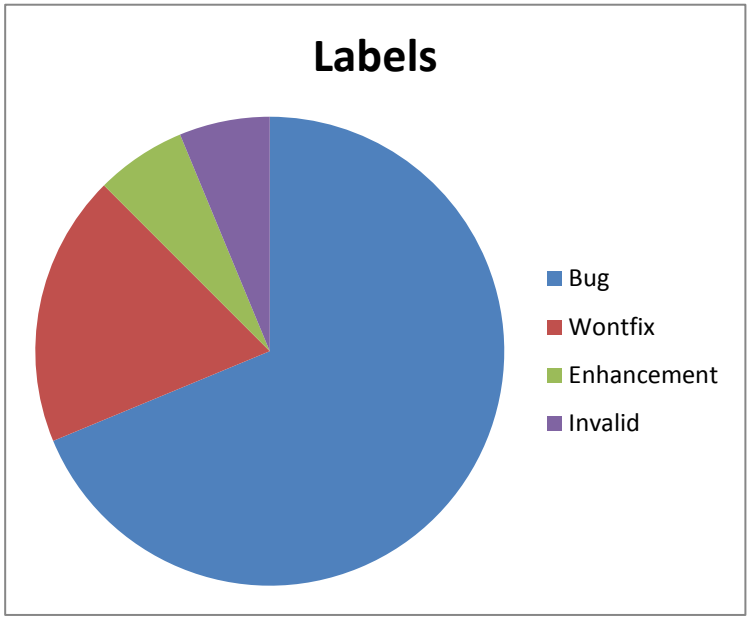
Vemos que todas las pruebas que han sido diseñadas, se han implementado, ejecutado y que actualmente todas ellas son exitosas.

- **Cobertura:**



La cobertura del sistema vemos que es alta, con una media del 88% y una desviación típica de un 9%. La cobertura de rama tiene un 81% de media.

Análisis del perfil de detección de errores.



Vemos que casi todos los errores los hemos clasificado con la etiqueta Bug (11 veces con un 69%) Le siguen los errores de WontFix o Arreglo (3 veces y 19%). Por último aparecen las etiquetas Enhancement o Mejora y Invalid o Invalida (1 vez y 6 % respectivamente).

Informe de errores abiertos y cerrados por nivel de importancia.

Nombre	Gravedad
Registro_002	Baja
Registro_008	Baja
Registro_009	Baja
Registro_010	Baja
Registro_016	Baja
Estados_006	Alta
Estados_015	Media
Piscina_006	Alta
Piscina_013	Alta
Piscina_019	Alta
Piscina_026	Media
CheckStyle 4041 warnings	Baja
CheckStyle 0 warnings	Baja (quedó invalidada)
Travis-CI falla	Alta
Excepciones	Media
GraphWalker	Media

Hemos asignado esas gravedades siguiendo la lógica que nos parecía más importante. Por ejemplo, consideramos muy importante que la Build de travis falle, y lo relacionado con los valores de la piscina, como son el PH, temperatura, o el % de cloro, que no se permite pasar de ciertos valores.
Otro ejemplo que consideramos gravedad alta, es cuando la piscina tiene que estar en un estado que afecte a las personas de la piscina.
Con gravedad media, consideramos el resto puesto que es lo referido con los informes emitidos para valores poco utilizados. También lo utilizamos en la piscina cuando se añade un sensor a una piscina que no es la misma a la que se añadió.
Por último, con gravedad baja, marcamos los errores que nos marcaba la herramienta Checkstyle, pues consideramos que son fáciles de solucionar, a pesar de la cantidad, y la mayor parte eran de formato.

Evaluación global del estado de la calidad y estabilidad actual.

Actualmente hemos elevado la calidad del proyecto, debido a la gran cantidad de pruebas que hemos realizado ha elevado la confianza en el software. Del mismo modo nos hemos visto obligados a refactorizar dos clases (Piscina) para mejorar la mantenibilidad del código.
Hemos empleado todas las técnicas propuestas acorde a nuestro proyecto, que consiste en una librería con la excepción de QuickCheck, para el cual lo hemos reemplazado en los test, mediante la utilización de valores random.
Para abordar el testeo de este proyecto, nos hemos centrado principalmente en las pruebas de unidad e integración, porque hablamos de una librería sobre la cúal se construirán productos de mayor envergadura, y un componente base ha de funcionar perfectamente.
No hemos detectado gran cantidad de errores, por lo tanto el número de issues creadas fue bastante bajo. En consecuencia, las gráficas no aportan datos significativos como para realizar un análisis profundo de las issues.
Finalmente, las pruebas de rendimiento no observan gran uso de recursos y la librería es rápida y eficiente. Consideramos que está lo suficientemente probada, para utilizar segmento de producción.

7. OTROS ASPECTOS DE INTERÉS

No consideramos ningún aspecto de interés a mayores lo explicado anteriormente.