

Tipo de artículo: Artículo original

Temática: Soluciones Informáticas

Recibido: 11/10/2018 | Aceptado: 22/12/2018 | Publicado: 28/01/2019

## Implementación de pruebas de software para un sistema para la gestión de datos de urocultivo

### *Implementation of software tests for a system for urine culture data management*

Karla García Benitez <sup>1\*</sup>, Leyanys Acosta Calderón <sup>2</sup>, Yosbel Lázaro Guirola Manresa <sup>3</sup>

<sup>1</sup> Centro de Estudio de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. [kbenitez@estudiantes.uci.cu](mailto:kbenitez@estudiantes.uci.cu)

<sup>2</sup> Centro de Estudio de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. [leyaac19@gmail.com](mailto:leyaac19@gmail.com)

<sup>3</sup> Centro de Estudio de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. [yguirola@uci.cu](mailto:yguirola@uci.cu)

\* Autor para correspondencia: [kbenitez@estudiantes.uci.cu](mailto:kbenitez@estudiantes.uci.cu)

---

#### Resumen

A partir del proceso de implementación del sistema para la gestión de datos de urocultivo, se hace necesario comprobar el correcto funcionamiento de los requisitos especificados. Las pruebas de software representan la forma de comprobar la correspondencia entre los elementos solicitados por el cliente y el correcto funcionamiento de esto. La presente investigación describe una solución al proceso de prueba después de la implementación de los requisitos definidos para lo cual se realizan pruebas de interfaz utilizando el método de partición equivalencia.

**Palabras clave:** pruebas de software; sistema de gestión; urocultivo.

#### Abstract

*From the process of implementing the system for the management of urine culture data, it is necessary to verify the correct functioning of the specified requirements. The software tests represent the way to check the correspondence between the elements requested by the client and the correct functioning of this. The present investigation describes a solution to the test process after the implementation of the defined requirements for which interface tests are performed using the equivalence partition method.*

**Keywords:** software tests; management system; uroculture.

## **Introducción**

La presente investigación describe los diferentes mecanismos utilizados para llevar a cabo el desarrollo y validación del sistema para la gestión de datos de urocultivo. Además, basado en la propuesta de solución planteada en el capítulo anterior, se procede a desarrollar el flujo de trabajo de implementación. En la que se describen sus principales artefactos, como el modelo de implementación que incluye componentes, subsistemas de implementación, diagramas de componentes y el modelo de despliegue del sistema. Posteriormente se le aplican las pruebas al sistema partiendo de la confección y descripción de los casos de prueba.

## **Materiales y métodos o Metodología computacional**

Modelo de implementación: El modelo de implementación describe la estructura general del *software* con vistas a su construcción, ejecución e instalación. Describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, ficheros de código fuente, ejecutables También describe cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación y lenguajes de implementación empleados, y cómo dependen los componentes unos de otros (JACOBSON 2000), (SALAS and CERÓN 2014), (PONCE *et al.* 2014). Se muestra a continuación como han quedado modelado el diagrama de componentes y modelo de despliegue del sistema.

### **Diagrama de componentes**

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software: código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las limitaciones imputadas por los lenguajes de programación y las herramientas utilizadas en el desarrollo (JACOBSON 2000), (FEBLES *et al.* 2011), (PÉREZ 2007).

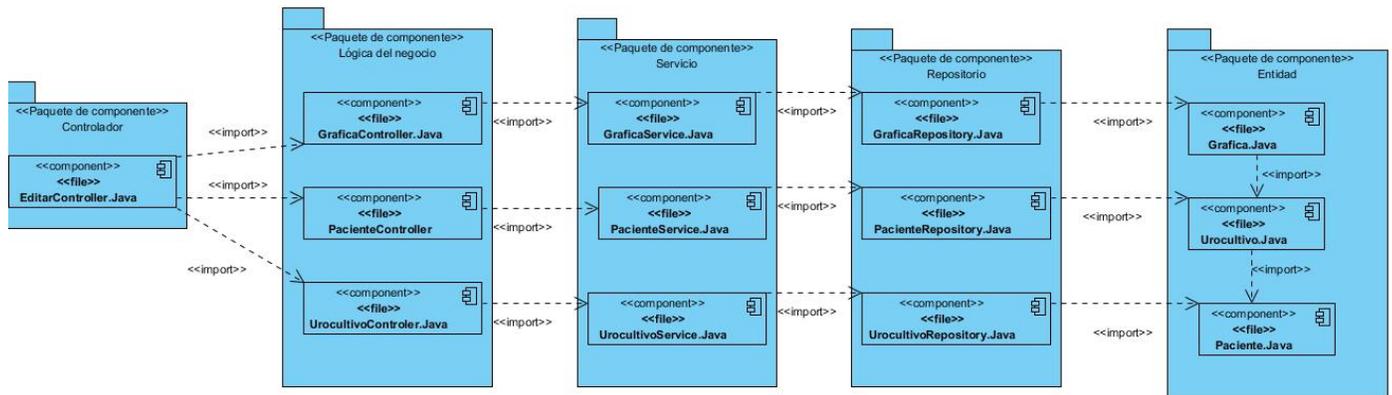


Figura 1 Diagrama de componentes

Modelo de despliegue: Un diagrama de despliegue muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Muestra la configuración de los elementos de procesamiento en tiempo de ejecución, los componentes de software, hardware, procesos y objetos que los ejecutan. Este diagrama es útil para ilustrar la arquitectura física de un sistema (JACOBSON 2000), (KANER *et al.* 1999 ), (MAR, O *et al.* 2016)

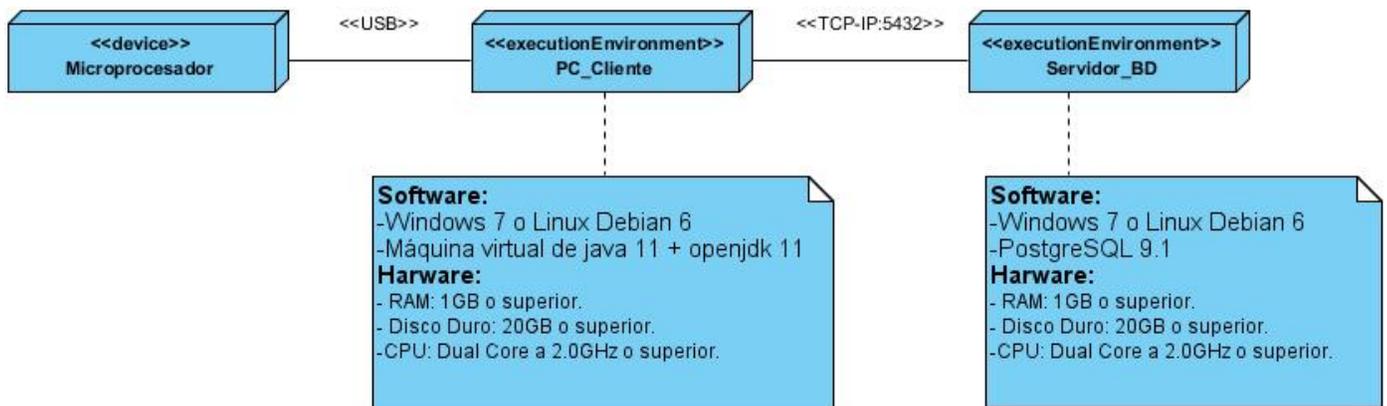


Figura 2. Diagrama de despliegue

Para interactuar con el sistema es necesario contar con una *PC\_Cliente* que tenga instalado la máquina virtual de java 11 y el openjdk 11, además necesita conectarse con el microprocesador por puerto usb permita comunicación con el equipo automático (GONZÁLEZ 2013), (MAR, O. *et al.* 2017). Los datos persistentes resultantes del sistema se

almacenan en el servidor de bases de datos (*Servidor\_BD*), accesible mediante el protocolo TCP-IP (protocolo de control de transmisión de datos entre computadoras, del inglés, *Transmission Control Protocol – Internet Protocol*).

Estándares de codificación

Los estándares de codificación permiten entender de manera rápida, fácil y sencilla el código empleado en el desarrollo de un *software*. Además, garantizan un mantenimiento óptimo de dicho código por parte del programador. Se presenta los estilos de codificación utilizado:

Los comentarios se realizan mediante el símbolo de número // seguido de un espacio.

Casi sin excepción, los nombres de clases deben utilizar la convención “BuscarController” (palabras que comienzan con mayúsculas).

Usa 4 (cuatro) espacios por guiones para java.

Las importaciones deben estar en líneas separadas.

```
16 import javafx.stage.Stage;
17 import javafx.util.Callback;
18 import org.slf4j.Logger;
19 import org.slf4j.LoggerFactory;
20 import org.springframework.beans.factory.annotation.Autowired;
21 import org.springframework.stereotype.Controller;
22
23 import java.io.IOException;
24 import java.util.ArrayList;
25 import java.util.List;
26 import java.util.Locale;
27
28 @Controller
29 public class BuscarController { //Clase controladora de la vista Buscar Paciente
30     @FXML
31     private Button bbuscar;
32     @FXML
33     private Button hp;
34     @FXML
35     private Button ayudabn;
```

Figura 3 Ejemplo de estándar de Programación

## Resultados y discusión

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. El objetivo de las mismas es encontrar el máximo número posible de errores con una cantidad manejable de esfuerzo aplicado en un período realista de tiempo (PRESSMAN 2008). Con las mismas se garantiza que el producto final funcione como fue diseñado e implemente de manera correcta los requisitos identificados.

Para dar inicio a las pruebas de un *software* lo primero es describir una estrategia de prueba donde quede plasmado los niveles de prueba a tratar, así como los tipos de prueba a emplear en cada nivel, los métodos de prueba a aplicar, así como las técnicas a utilizar para cada método. Las estrategias de pruebas describen y verifican el enfoque de la misma. Luego de realizar una estrategia de prueba se decidió aplicar pruebas de unidad e integración para medir el cumplimiento del objetivo de la presente investigación.

### **Pruebas de unidad**

Las **pruebas de unidad** se realizarán con el objetivo de comprobar que el sistema implementado, entendido como una unidad funcional de un programa independiente, está correctamente codificado. Para la generación de casos de prueba de unidad, se decidió utilizar la técnica de camino básico del método de caja blanca y la técnica de partición equivalente del método de caja negra; para probar de la manera más completa posible el sistema implementado.

Método de caja blanca: las pruebas unitarias son una de las piedras angulares de AUP. Todos los sistemas deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, las pruebas deben ser definidas antes de realizar el código (“Test-driven programming”). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

Detección y corrección de errores: Cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Así mismo, se generan nuevas pruebas para verificar que el error haya sido resuelto (MORENO 2013). Las pruebas unitarias realizadas a través del sistema que ofrece Spring llamado test, se le fueron realizadas a las clases Grafica, Paciente, Urocultivo, GraficaRepository y AppApplication. A continuación, se muestra el resultado de las pruebas unitarias que se realizan al sistema, utilizando el módulo test

Figura 4. Resultado de las pruebas

### **Método de caja negra**

El método de prueba de caja negra, también denominada prueba de comportamiento, se centra en los requisitos funcionales del software. O sean, permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del programa.

Para el caso de las pruebas de caja negra, se aplicará la técnica de partición equivalente, la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (PRESSMAN 2008).

### **Desarrollo de la prueba**

El diseño de casos de prueba (DCP) para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. A continuación se muestra un ejemplo de los DCP realizados, específicamente el del CUS Gestionar urocultivo.

#### **DCP del CUS: Gestionar urocultivo**

Descripción general: El CUS se inicia cuando el usuario selecciona de la interfaz “leer”, el círculo que representa una muestra de urocultivo y culmina con la realización de una de las acciones: Insertar, Mostrar, Modificar, Detener la lectura de la muestra seleccionada o Eliminar los datos de una muestra de urocultivo que fue detenida

Condiciones de ejecución: El usuario debe estar autenticado en el sistema, así como contar con los permisos para realizar alguna de las operaciones antes mencionadas. En el caso de que se pretenda Modificar, Mostrar o Detener alguna muestra de urocultivo, este debe haber sido insertado anteriormente al sistema, para Eliminar la muestra debe haber sido detenida anterioridad. En el caso de que se pretenda Insertar los datos de alguna muestra de urocultivo, el usuario solo podrá realizar la acción si la muestra de urocultivo fue leída e insertada en la base de datos del, en ese caso podrá insertar los datos o no ejecutar la acción hasta se haya leído una muestra de urocultivo.

**Secciones a probar en el CUS: Gestionar urocultivo**

Tabla 1. DCP Gestionar urocultivo

<i>Nombre de la sección</i>	<i>Escenarios de la sección</i>	<i>Descripción de la funcionalidad</i>	<i>Flujo Central</i>
<p><b>SC 1:</b>                      “Insertar datos de una muestra de urocultivo”</p>	<p><b>EC 1.1:</b>                      El usuario inserta los datos requeridos de manera correcta.</p>	<p>El sistema inserta los datos del urocultivo seleccionado</p>	<ol style="list-style-type: none"> <li>1. Clic en “Leer”.</li> <li>2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”.</li> <li>3. Clic en “Insertar datos de urocultivo”.</li> <li>4. Inserta datos en los campos.</li> <li>5. Clic en “Guardar”.</li> </ol>
	<p><b>EC 1.2:</b>                      El usuario deja campos vacíos.</p>	<p>El sistema muestra un mensaje “Campo vacío” especificando que debe rellenar el campo que dejo en blanco.                      El borde del campo de texto se torna de color rojo</p>	<p>Clic en “Leer”.                      Clic en “Circulo correspondiente a la muestra de urocultivo deseada”.                      Clic en “Insertar datos de urocultivo”.                      Inserta datos en los campos.                      Clic en “Guardar”.</p>
	<p><b>EC 1.3:</b>                      El usuario selecciona la opción cancelar.</p>	<p>El sistema elimina los datos creados</p>	<ol style="list-style-type: none"> <li>1. Clic en “Leer”.</li> <li>2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”.</li> <li>3. Clic en “Insertar datos de urocultivo”.</li> <li>4. Inserta datos en los campos.</li> </ol>

			5. Clic en “Cancelar”.
<b>SC 2:</b> <i>“Mostrar los datos de una muestra del urocultivos”.</i>	<b>EC 2.1:</b> Muestra detalles de las muestras de urocultivo.	El sistema muestra los datos asociados a las muestra de urocultivo.	1. Clic en “Leer”. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”. Clic en “Mostrar urocultivo”. .
	<b>EC 2.2</b> El usuario selecciona la opción imprimir	El sistema brinda la posibilidad de imprimir	1. Clic en “Leer”. 2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”. 3. Clic en “Mostrar urocultivo”. 4. Clic en “Imprimir”
<b>SC 3:</b> <i>“Modificar los datos de una muestra de urocultivo”.</i>	<b>EC 3.1:</b> El usuario modifica los datos de manera correcta.	El sistema muestra un mensaje de confirmación: "¿Está usted seguro que desea modificar?" y se actualizan los datos correspondientes al urocultivo modificado.	1. Clic en “Leer”. 2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”. 3. Clic en “Modificar urocultivo”. 4. Inserta datos en los campos. 5. Clic en “Guardar”.
	<b>EC 3.2:</b> El usuario deja campos a modificar vacíos.	El sistema muestra un mensaje “Campo vacío” especificando que debe rellenar el campo que dejo en blanco. El borde del campo de texto se torna de color rojo	1. Clic en “Leer”. 2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”. 3. Clic en “Modificar urocultivo”. 4. Inserta datos en los campos. 5. Clic en “Guardar”.
	<b>EC 3.3:</b> El usuario selecciona la opción cancelar.	El sistema elimina los datos creados	6. Clic en “Leer”. 7. Clic en “Circulo correspondiente a la muestra de urocultivo

			deseada”. 8. Clic en “Modificar urocultivo”. 9. Inserta datos en los campos. 10. Clic en “Cancelar”.
<b>SC 4:</b> “Detener la lectura de la muestra seleccionada”.	<b>EC 4.1:</b> El usuario detiene la lectura de una muestra de urocultivo	El sistema deja de leer la muestra seleccionada en el dispositivo y cambia su color a gris	1. Clic en “Leer”. 2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”. 3. Clic en “Detener urocultivo”.
<b>SC 5:</b> “Eliminar los datos de una muestra de urocultivo que fue detenida”.	<b>EC 5.1:</b> Eliminar muestra de urocultivo.	El sistema muestra un mensaje de confirmación: “¿Está usted seguro que desea eliminar?” y elimina la muestra de urocultivo con todos sus datos.	1. Clic en “Leer”. 2. Clic en “Circulo correspondiente a la muestra de urocultivo deseada”. 3. Clic en “Eliminar urocultivo”.

### Matrices de datos

Tabla 2. Insertar datos de la muestra de urocultivo

Id. Del escenario	Escenario	Clasificación de la cepa	Tipo de muestra	Microorganismo	Procedencia	Paciente	Respuesta del sistema	Resultado de la prueba
EC 1.1	El usuario inserta los datos requeridos de manera correcta.	V	V	V	V	V	El sistema inserta los datos del urocultivo seleccionado	Satisfactorio

EC 1.2	El usuario deja campos vacíos.	I	V	V	I	V	El sistema muestra un mensaje “Campo vacío” especificando que debe rellenar el campo que dejo en blanco. El borde del campo de texto se torna de color rojo	Satisfactorio
EC 1.3	El usuario selecciona la opción cancelar.	NA	NA	NA	NA	NA	El sistema elimina los datos creados	Satisfactorio

Tabla 3. Mostrar los datos de una muestra de urocultivo

Id. Del escenario	Escenario	McFarland	Clasificación	Clasificación de la cepa	Tipo de muestra	Microorganismo	Procedencia	Fecha	Nombre	Apellidos	Elaborado por	Historia Clínica	Sexo	Nosocomial	Fallecido	Respuesta del sistema	Resultado de la prueba
-------------------	-----------	-----------	---------------	--------------------------	-----------------	----------------	-------------	-------	--------	-----------	---------------	------------------	------	------------	-----------	-----------------------	------------------------

EC 2.1	Muestra detalles de las muestras de urocultivo.	NA	El sistema muestra los datos de los indicadores insertados en el sistema por asignatura.	Satisfactorio												
EC 2.1	El usuario selecciona la opción imprimir	NA	El sistema brinda la posibilidad de imprimir													

Tabla 4. Modificar los datos de una muestra de urocultivo

Id. Del escenario	Escenario	Clasificación de la cepa	Tipo de muestra	Microorganismo	Procedencia	Paciente	Respuesta del sistema	Resultado de la prueba
EC 3.1	El usuario modifica los datos de manera correcta.	V	V	V	V	V	El sistema muestra un mensaje de confirmación: "¿Está usted seguro que desea modificar?" y se actualizan los datos correspondientes al urocultivo modificado.	Satisfactorio
EC 3.1	El usuario deja campos a modificar vacíos.	I	V	V	I	V	El sistema muestra un mensaje "Campo vacío" especificando que debe rellenar el campo que dejo en blanco. El borde del campo de texto se torna de color rojo	Satisfactorio
EC 3.1	El usuario selecciona la opción cancelar.	NA	NA	NA	NA	NA	El sistema elimina los datos creados	Satisfactorio

Tabla 5. Detener la lectura de la muestra seleccionada

Id. Del escenario	Escenario	Clasificación de la cepa	Tipo de muestra	Microorganismo	Procedencia	Paciente	Respuesta del sistema	Resultado de la prueba
-------------------	-----------	--------------------------	-----------------	----------------	-------------	----------	-----------------------	------------------------

<b>EC 4.1</b>	El usuario detiene la lectura de una muestra de urocultivo	NA	NA	NA	NA	NA	El sistema deja de leer la muestra seleccionada en el dispositivo y cambia su color a gris	Satisfactorio
---------------	--	----	----	----	----	----	--	---------------

Tabla 61. Eliminar los datos de una muestra de urocultivo que fue detenida

Id. Del escenario	Escenario	Clasificación de la cepa	Tipo de muestra	Microorganismo	Procedencia	Paciente	Respuesta del sistema	Resultado de la prueba
EC 5.1	Eliminar muestra de urocultivo	NA	NA	NA	NA	NA	El sistema muestra un mensaje de confirmación: "¿Está usted seguro que desea eliminar?" y elimina la muestra de urocultivo con todos sus datos.	Satisfactorio

### Resultado de la aplicación de las pruebas

Como parte de la ejecución de las pruebas de caja negra se realizaron 12 casos de pruebas en 3 iteraciones de pruebas para comprobar el correcto funcionamiento del sistema. En la primera iteración se identificaron 8 no conformidades (NC). Una vez corregidas, se procedió a realizar una segunda en la que se identificaron 3 nuevas NC y finalmente se realizó una iteración en la que no se encontró unas deficiencias que fue resuelta, razón por la que se definió no realizar más iteraciones.

Para mayor entendimiento de este proceso se muestra en la tabla la descripción de las iteraciones realizadas durante las pruebas y un gráfico de barras figura que ilustra las NC por cada iteración.

Tabla 2. Resultado de la aplicación de las pruebas

No. Iteración	Cantidad NC	Tipo Error	Impacto
<b>1</b>	<b>8</b>	4 Usab - Estética y diseño	Medio
		2 Validación	Alto

		2 Funcional	Alto
2	3	2 Validación	Alto
		1 Usab - Estética y diseño	Medio
3	1	1 Usab - Estética y diseño	Medio

Las iteraciones permitieron garantizar que el sistema cumpla con las especificaciones que se trazaron en el transcurso de la investigación.

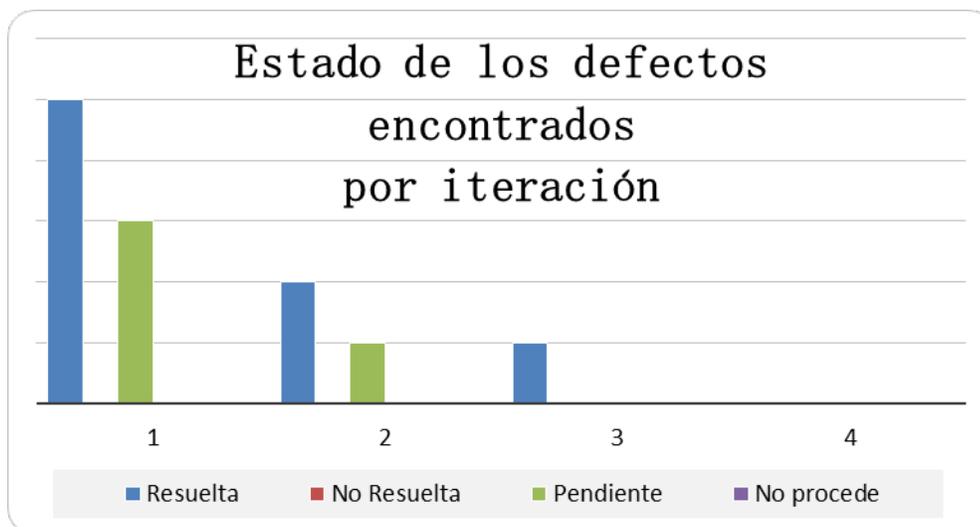


Figura 5. Resultado de la aplicación de las pruebas

## Conclusiones

Con la implementación del sistema para la gestión de los datos de los urocultivos, se obtuvo como resultado del proceso de ingeniería, la construcción del diagrama de componentes para el CUS “Gestionar urocultivos”, en el cual se representó una vista estática del sistema, especificándose las relaciones y dependencias entre cada uno de los componentes del mismo.

A partir de la elaboración del diagrama de despliegue se identificaron los recursos necesarios para el despliegue y correcto funcionamiento de la solución. La descripción del empleo de los estándares de codificación y los estilos de

programación permitieron a su vez hacer más fácil el entendimiento del código del programador y facilitar el mantenimiento futuro del sistema.

Una vez implementado el sistema se realizó las pruebas a nivel de unidad; logrando probar de la manera más completa posible la solución implementada y el cumplimiento de los requisitos funcionales y no funcionales trazados durante la presente investigación. Con las pruebas de integración se garantizó que el sistema se adaptara al ambiente para el cual fue concebido permitiendo realizar la lectura de muestra de análisis de urocultivo.

## Referencias

- FEBLES, A.; T. CAPOTE, *et al.* Una experiencia novedosa para el testing desarrollada por un departamento de pruebas de software *RCCI* 2011, Vol 5(No 2).
- GONZÁLEZ, J. Propuesta de algoritmo de clasificación genética *RCI*, 2013, Vol. 4 (No.2): 37-42.
- JACOBSON. *El proceso unificado de desarrollo de software*. 2000. p.
- KANER, C.; J. FALK, *et al.* *Testing Computer Software 2nd Edition*, John Wiley 1999
- MAR, O.; L. ARGOTA, *et al.* Módulo para la evaluación de competencias a través de un Sistema de Laboratorios a Distancias *Revista Cubana de Ciencias Informáticas*, 2016, 10(2): 132-147.
- MAR, O.; I. SANTANA, *et al.* Competency assessment model for a virtual laboratory system and distance using fuzzy cognitive map *Revista Investigación Operacional*, 2017, 38(2): 170.178.
- MORENO, S. Manejo de sistemas de información para la organización de procesos de la gerencia de protección y aseguramiento de ingresos de la compañía, 2013.
- PÉREZ, B. Estrategia de gestión de las pruebas funcionales en el Centro de Ensayos de Software *REICIS*, 2007, Vol.3 (No. 3).
- PONCE, J. F.; F. DOMINGUEZ, *et al.* Pruebas de aceptación orientadas al usuario. contexto ágil para un proyecto de gestión *Ibersid: Revista de sistemas de información y comunicación*, 2014.
- PRESSMAN, R. S. *Ingeniería de Software*. Sexta Edición. 2008. p.
- SALAS, M. and C. CERÓN Sistema Web para Evaluar las Competencias mediante Pruebas Objetivas en Educación Superior *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 2014, (12).