

Tipo de artículo: Artículo original

Temática: Soluciones Informáticas

Recibido: 10/08/2018 | Aceptado: 10/12/2018 | Publicado: 20/12/2018

Análisis y diseño de un sistema para la gestión de información de análisis de urocultivo

Analysis and design of a system for the management of urine culture analysis information

Karla García Benitez^{1*}, **Leyanys Acosta Calderón**², **Yosbel Lázaro Guirola Manresa**³

¹ Centro de Estudio de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. kbenitez@estudiantes.uci.cu

² Centro de Estudio de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. leyaac19@gmail.com

³ Centro de Estudio de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales. Universidad de las Ciencias Informáticas. yguirola@uci.cu

* Autor para correspondencia: kbenitez@estudiantes.uci.cu

Resumen

La presente investigación describe una solución que realiza una implementación de un conjunto de algoritmos computacionales, que permiten graficar el crecimiento bacteriano, a partir de la de la unidad formadora de colonias (UFC)/ml que presenta una muestra en tiempo real, mediante su curva de crecimiento. Se generan además un conjunto de reportes que facilita el diagnóstico clínico y permite la visualización de los datos generados por el procesamiento de la muestra de urocultivo de un paciente.

Palabras clave: curvas de crecimiento; sistema de gestión; urocultivo.

Abstract

The present investigation describes a solution that performs an implementation of a set of computational algorithms, which allow to graph the bacterial growth, from the one of the colony forming unit (CFU) / ml that presents a sample in real time, through its curve of growth. A set of reports is also generated that facilitates the clinical diagnosis and allows the visualization of the data generated by the processing of a patient's urine sample.

Keywords: growth curves; Diramic; management system; uroculture.

Introducción

La herramienta DIRAMIC es un equipo semiautomático que dentro de las ventajas fundamentales se encuentra la determinación rápida de la infección urinaria en solo 4 horas, realiza la gestión para el análisis de urocultivo, el cual es un estudio de laboratorio destinado a detectar la presencia de posibles bacterias o microbios a partir de una muestra de orina (BERTONI *et al.* 2017).

Como parte del proceso de mejora continua de la calidad, se detectaron no conformidades de la herramienta DIRAMIC la cual se pudo constatar a partir del estudio de la literatura científica, la revisión de reporte técnicos y la aplicación de entrevistas dirigidas a usuarios que interactúan con la herramienta durante los últimos años, identificándose las siguientes deficiencias:

- La herramienta posee informaciones irrelevantes en los campos de entradas de los datos de los urocultivos.
- Se muestran resultados de falsos negativos relacionados con microorganismos de crecimiento lento.
- No implementa mecanismos de validación que restrinja la introducción de información duplicada.
- No implementa visualización del crecimiento de los microorganismos de una muestra en tiempo real.

A partir del análisis sobre las deficiencias identificadas se provoca gestión de datos irrelevante, almacenamiento de informaciones innecesarias lo que trae como consecuencias que se ralenticen las búsquedas de informaciones en la generación de reportes. Además no se identifican las distintas fases del crecimiento bacteriano, para un resultado preciso en el procesamiento de los datos de los urocultivos de los pacientes.

Problemas como los antes mencionados han sido abordados en la literatura científica a partir del estudio de las curvas de crecimiento de la unidad formadora de colonias (UFC)/ml, lectura en tiempo real y el comportamiento de los microorganismos existentes dentro de una muestra urinaria (DÍAZ 2017), (LEONARD *et al.* 2017), (VANEGAS and RAMÍREZ 2016), (GONZÁLEZ 2013), (MAR *et al.* 2017). A partir del análisis anterior se hace necesaria la implementación de una nueva herramienta capaz de superar las limitaciones actuales.

Para dar solución a la problemática planteada se crea un convenio de colaboración entre el CNIC y el Centro de Estudio de Matemática Computacional con el objetivo de crear una nueva herramienta. La nueva propuesta debe ser capaz de representar las curvas de crecimiento de los microorganismos de una muestra, estructurar la gestión de la información que garantice la validación de los datos que son procesados y aumentar la precisión de los resultados con el fin de eliminar los falsos negativos relacionados con microorganismos de crecimiento lento.

Materiales y métodos o Metodología computacional

Teniendo en cuenta las ideas expuestas que se emplearan en el desarrollo del sistema para la gestión de análisis de urocultivos, se realiza un levantamiento de los requisitos del sistema, con el propósito de dar una solución factible al problema en cuestión. En el presente capítulo se pretende definir el modelo conceptual, entidades y actores que intervienen, los requisitos funcionales y no funcionales, así como el diagrama de caso de uso del sistema y las relaciones existentes. El proceso de desarrollo será guiado por la metodología de desarrollo AUP (UCI)(GONZÁLEZ and MAR 2015), la cual será la base para la implementación del sistema informática para la gestión de análisis de urocultivos a partir de las curvas de crecimiento.

Propuesta de solución: Se propone la implementación de una aplicación para la gestión de procesamiento de los datos de los urocultivos a partir de las curvas de crecimiento. Dicha aplicación basa su funcionamiento en almacenar y graficar las curvas de crecimiento como método para el procesamiento de los datos de los urocultivos. La propuesta contribuirá a determinar el resultado de una muestra de urocultivo en positivo o negativo a partir de las curvas de crecimiento.

Modelo conceptual

Tiene como objetivo identificar los conceptos significativos en el proceso de objeto de estudio, así como exponer los atributos y las asociaciones existentes entre ellos. El modelo conceptual describe las entidades del proceso y no representa componentes del software. La figura 1 presenta el modelo conceptual que representa la descripción del proceso.

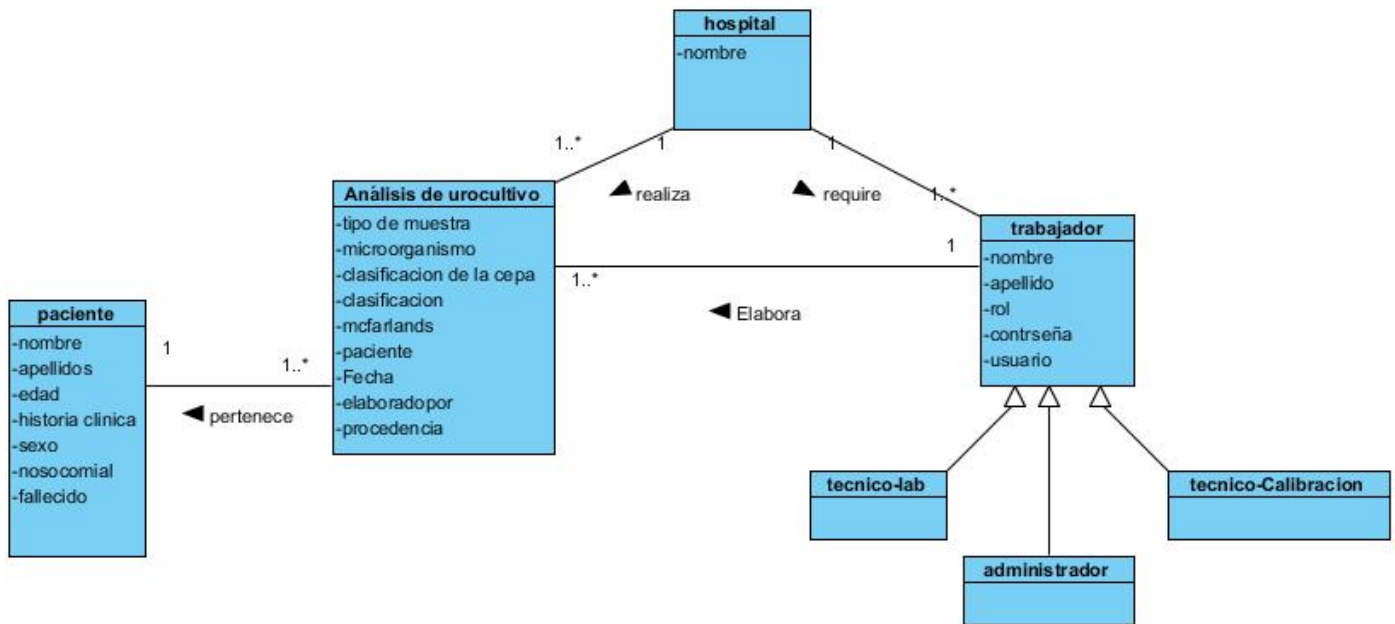


Figura 1. Diagrama de modelo conceptual

La tabla 1 presenta una descripción de los conceptos asociados al dominio del problema.

Tabla 1. Descripción de los conceptos

Concepto	Descripción
Hospital	Centro hospitalario donde se lleva a cabo el análisis de los urocultivos con el sistema semi-automatico Diramic
Paciente	Persona que contiene la muestra que más tarde será analizada
Urocultivo	Muestra urinaria dada por el paciente para ser examinada
Trabajador	Personal capacitado para trabajar en los laboratorios de microbiología
administrador	Persona encargada de gestionar los usuarios
Tecnico_lab	Persona encargada de dar los resultados del análisis de las muestras
Tecnico_calibrador	Persona encargada de la calibración del sistema

Especificación de los requisitos del sistema

Una etapa inicial y muy importante dentro del proceso de la Ingeniería de Software, es la Ingeniería de Requisitos, donde se lleva a cabo el proceso de descubrir, analizar, escribir y verificar los servicios y restricciones, del sistema de software que se desea producir; este proceso se realiza mediante la obtención, el análisis, la especificación, la validación y la administración de los requerimientos del software. Los requisitos de software son las necesidades de

los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar (SOMMERVILLE 2005).

Requisitos Funcionales

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, también pueden declarar explícitamente lo que el sistema no debe hacer (SOMMERVILLE 2005). A partir de la descripción del proceso se logró identificar 28 requisitos funcionales que son relacionados a continuación.

RF1. Autenticar usuario (administrador)

RF2. Insertar los datos del usuario (administrador)

RF3. Modificar los datos del usuario (administrador)

RF4. Eliminar usuario (administrador)

RF5. Buscar usuario a partir de los datos insertado (administrador)

RF6. Listar usuarios (administrador)

RF7. Cambiar contraseña de usuarios (tec lab, tec calibración, administrador)

RF8. Leer la concentración de células de cada muestra de urocultivo (tec lab)

RF9. Insertar datos de una muestra de urocultivo (tec lab)

RF10. Mostrar los datos de una muestra del urocultivos (tec lab)

RF11. Modificar los datos de una muestra de urocultivo (tec lab)

RF12. Detener la lectura de la muestra seleccionada (tec lab)

RF13. Eliminar los datos de una muestra de urocultivo que fue detenida (tec lab)

RF14. Buscar los datos de un paciente por historia clínica (tec lab)

RF15. Insertar los datos del paciente que tiene una muestra de urocultivo (tec lab)

RF16. Modificar los datos del paciente que tiene una muestra de urocultivo (tec lab)

RF17. Insertar los datos del paciente (tec lab)

RF18. Modificar los datos del paciente (tec lab)

RF19. Eliminar los datos del paciente (tec lab)

RF20. Buscar los datos de un paciente (tec lab)

RF21. Listar los datos del paciente (tec lab)

RF22. Buscar los datos de una muestra de urocultivo (tec lab)

RF23. Listar los datos de una muestra de urocultivo (tec lab)

RF24. Modificar datos de una muestra de urocultivo registrado (tec lab)

RF25. Eliminar datos de una muestra de urocultivo registrado (tec lab)

RF26. Imprimir reporte del urocultivo (tec lab)

RF27. Calibración del sistema automático (tec calibración)

RF28. Calibración del calibrador independiente. (tec calibración)

Requisitos No Funcionales

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. A menudo se aplican al sistema en su totalidad. Normalmente apenas se aplica a características o servicios individuales del sistema (SOMMERVILLE 2005), (MAR and GULÍN 2018). Para la presente investigación se identificaron 7 requisitos no funcionales.

RNF1. Apariencia e interfaz externa.

Interfaz de usuario la información en esta interfaz, está organizada por un menú principal, el cual tiene un conjunto de opciones a escoger entre ellas la opción leer urocultivos que debe mostrar 48 canales de medición con los colores del semáforo. El color rojo representa los urocultivos que presentan infección debido a la existencia de microorganismos. El color verde representa los urocultivos que no poseen infección o que aún no se ha detectado. El color amarillo representa los urocultivos dudosos, es decir lo que aún no se ha podido comprobar la línea de crecimiento. Dentro de cada canal se mostrarán los datos necesarios para el registro del paciente con su muestra de análisis de urocultivo. Interfaz de hardware se necesita un puerto de USB.

RNF2. Usabilidad.

La estructura de botones y vínculos del menú principal debe de estar organizada según la funcionalidad que le corresponde, así como a los submenús que despliegan los mismos, con el objetivo de facilitar al usuario la interacción con la herramienta. Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos, y no deben revelar información interna.

RNF3. Portabilidad.

El sistema debe ser empleado en plataforma Windows y Linux. Para su correcto funcionamiento. Las diferentes plataformas donde sea instalado o las diferentes computadoras deben contar con las características necesarias para trabajar con bases de datos postgres SQL y la máquina virtual de java.

RNF4. Disponibilidad.

El sistema debe garantizar el acceso a personas autorizadas a los datos con los que se trabaja de forma segura y sencilla. Así el profesional debe encargarse de poner en marcha los sistemas que ayuden a conseguir este fin sin poner en riesgo la seguridad de la información.

RNF5. Soporte.

El sistema cuenta con un manual de ayuda, con el objetivo de explicar cómo funciona la gestión y el análisis de los datos de los urocultivos con los que trabajan los especialistas. También ofrece una breve descripción de como interactuar con las interfaces del sistema.

RNF6. Software.

Para el usuario: el sistema debe operar sobre los sistemas operativos Windows 7 o Linux Debian 6 y tener la máquina virtual de java 11 que maneja openjdk 11.

RNF7. Hardware.

Para la instalación del sistema se debe tener una computadora con las siguientes características:

RAM: 1GB o superior.

Disco Duro: 20GB o superior.

CPU: Dual Core a 2.0GHz o superior.

Resultados y discusión

Definición de los actores del sistema: Los actores del sistema son entidades distintas a los usuarios en el sentido de que estos son las personas reales que utilizaran el sistema, mientras que los actores representan cierta función que una persona real realiza (WEITZENFELD 2005). La tabla 2 presenta una descripción de los actores del sistema.

Tabla 2. Descripción de los actores del sistema

Actor del sistema	Descripción
administrador	Personal encargado de asignar los permisos a los usuario
Tec lab	Personal encargado de llevar a cabo la realización de los análisis de urocultivos, así como la gestión de todos los pacientes del sistema
Tec Calibración	Personal encargado de realizar la calibración del de sistema en cuestión

Diagrama de Caso de Uso del Sistema: Los casos de uso del sistema (CUS) se documentan con el empleo de un diagrama de caso de uso de alto nivel. El conjunto de casos de uso representa todas las interacciones posibles que se describirán en los requerimientos del sistema. Los actores en el proceso, que pueden ser individuos u otros sistemas,

se representan como figuras sencillas. Cada clase de interacción se constituye como una elipse con etiqueta. Líneas vinculan a los actores con la interacción. De manera opcional, se agregan puntas de flecha a las líneas para mostrar cómo se inicia la interacción (SOMMERVILLE 2005).

CUS1 (Autenticar usuario): RF1

CUS2 (Gestionar usuario): RF2, RF3, RF4, RF5, RF6

CUS3 (Cambiar contraseña de usuario): RF7

CUS4 (Leer datos de las muestras de urocultivo): RF8

CUS5 (Gestionar urocultivo): RF9, RF10, RF11, RF12, RF13

CUS6 (Gestionar paciente que posee una muestra de urocultivo): RF14, RF15, RF16

CUS7 (Gestionar paciente): RF17, RF18, RF19, RF20, RF21

CUS8 (Gestionar urocultivo registrado): RF22, RF23, RF24, RF25

CUS9 (Imprimir reporte de la muestra de urocultivo): RF26

CUS10 (Calibrar): RF27, RF28

CRUD (del inglés, Creating, Reading, Updating, Deleting) completo vigente en los CUS: “Gestionar urocultivo”, “Gestionar paciente”, “Gestionar urocultivo registrado”, “Gestionar paciente que posee una muestra de urocultivo”, “Gestionar usuario”

Múltiples actores (herencia o generalización entre actores) con roles comunes: el Administrador, tec.Calibración y tec.lab se comportan como un actor Usuario que se relaciona con el CUS “Autentica usuario” o “Cambiar contraseña”.

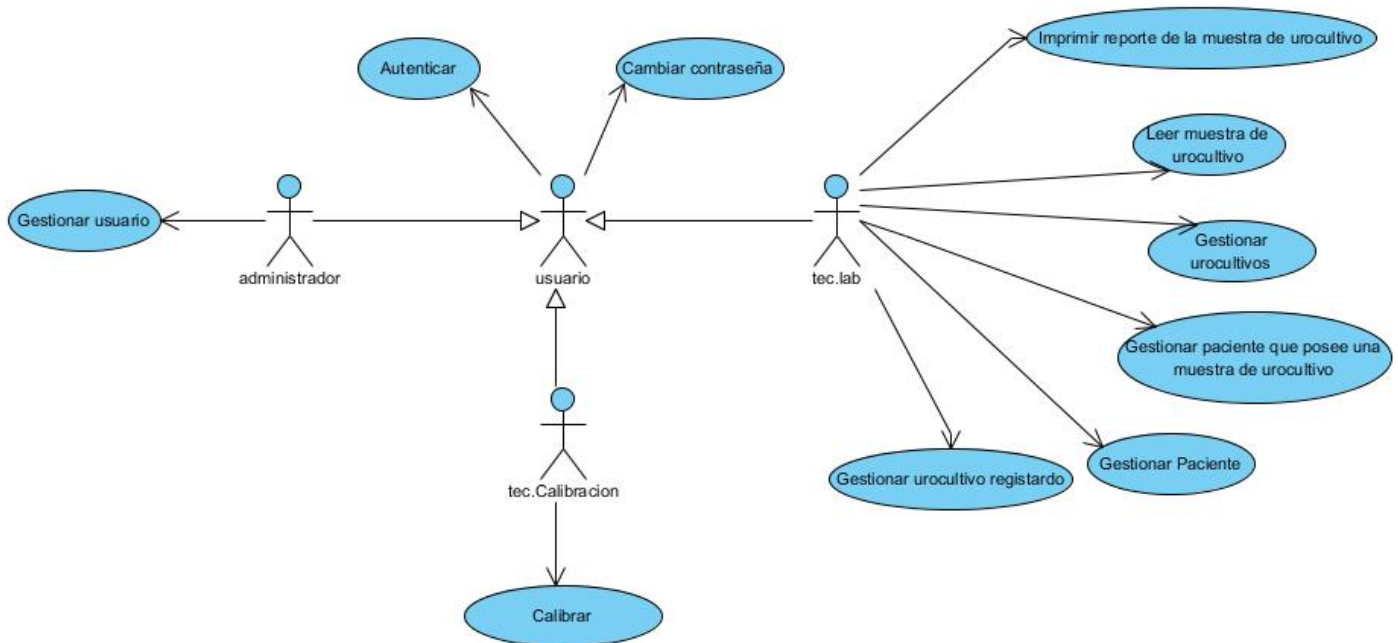


Figura 2. Diagrama de Caso de Uso del Sistema

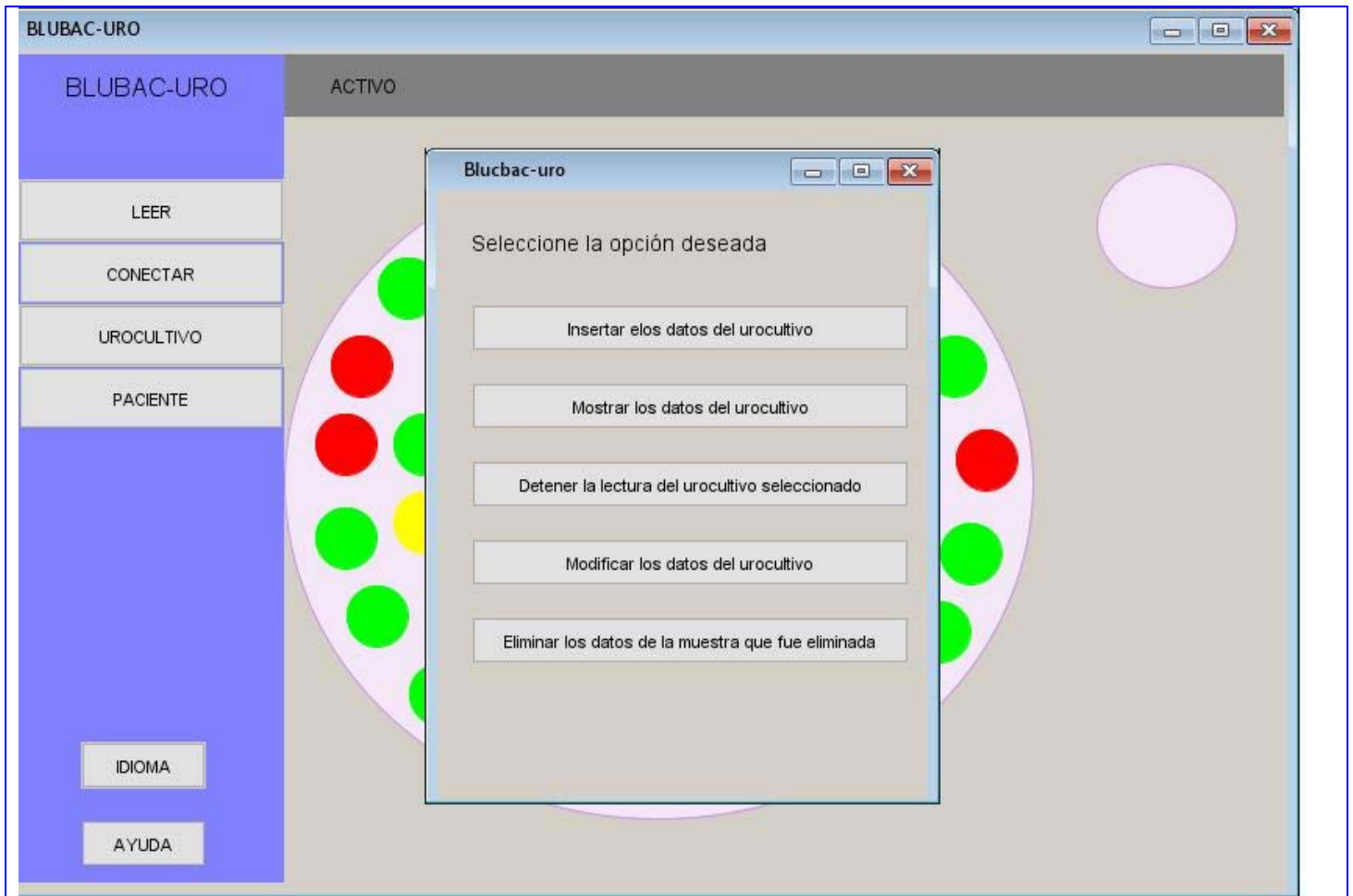
Descripción textual de los Casos de Uso del Sistema

La tabla 3 presenta una descripción del caso de uso gestionar urocultivo.

Tabla 3. Descripción de caso de uso gestionar urocultivo

Caso de Uso	<i>Gestionar urocultivo</i>
Actores	<i>Tec.lab</i>
Resumen	<i>Se registran todos los datos del urocultivo que están siendo leídos en el momento. Las acciones que se podrán ejecutar son: insertar datos de una muestra de urocultivo, mostrar los datos de una muestra del urocultivos, modificar los datos de una muestra de urocultivo, detener la lectura de la muestra seleccionada, eliminar los datos de una muestra de urocultivo que fue detenida. Estas acciones sólo se verán reflejadas en la BD al ejecutarse la acción guardar en el formulario.</i>
Complejidad	<i>Alto</i>
Prioridad	<i>Crítico</i>
Precondiciones	<i>Se se tiene que haber insertado el número de muestras</i>
Postcondiciones	<i>En dependencia de la acción que ejecute:</i> <ul style="list-style-type: none"> • <i>Insertar datos de una muestra de urocultivo</i>

	<ul style="list-style-type: none"> • <i>Mostrar los datos de una muestra del urocultivos</i> • <i>Modificar los datos de una muestra de urocultivo</i> • <i>Detener la lectura de la muestra seleccionada</i> • <i>Eliminar los datos de una muestra de urocultivo que fue detenida</i>
Referencias	<i>RF9, RF10, RF11, RF12, RF13</i>
Flujo de eventos	
Flujo básico	
Actor	Sistema
1.	<i>Selecciona de la interfaz leer el círculo que representa una muestra de urocultivo</i>
2.	<p><i>Permite realizar varias acciones:</i></p> <ul style="list-style-type: none"> • <i>Insertar un urocultivo Ver Sección 1: “Insertar datos de una muestra de urocultivo”</i> • <i>Mostrar un urocultivo. Ver Sección 2: “Mostrar los datos de una muestra del urocultivos”.</i> • <i>Modificar un urocultivo. Ver Sección 3: “Modificar los datos de una muestra de urocultivo”.</i> • <i>Detener un urocultivo. Ver Sección 4: “Detener la lectura de la muestra seleccionada”.</i> • <i>Eliminar un urocultivo. Ver Sección 5: “Eliminar los datos de una muestra de urocultivo que fue detenida”.</i>
3.	<i>Termina el caso de uso</i>
Prototipo de interfaz	



Prototipo de interfaz CUS Gestionar urocultivo

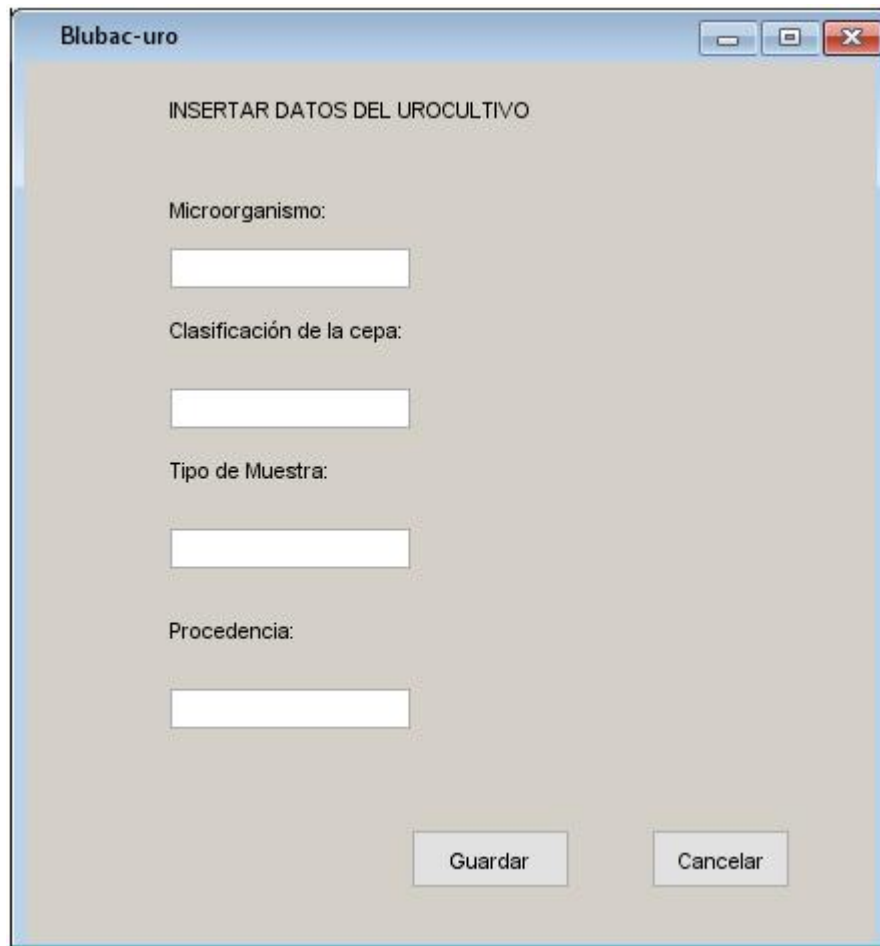
• **Sección 1** : “Insertar datos de una muestra de urocultivo”

Actor	Sistema
1. <i>Selecciona la opción insertar datos del urocultivo dentro de la ventana emergente</i>	2. <i>Muestra la ventana Insertar datos de un urocultivo</i>
3. <i>Se insertan los datos correspondientes del urocultivo y se selecciona la opción de guardar</i>	4. <i>.Se registran los datos en la base de datos</i>

Flujos alternos

	Actor	Sistema
	<i>3b Se selecciona la opción cancelar</i>	<i>5b Se cierra la interfaz y no se registra ningún dato en la base de datos</i>

Prototipo de interfaz



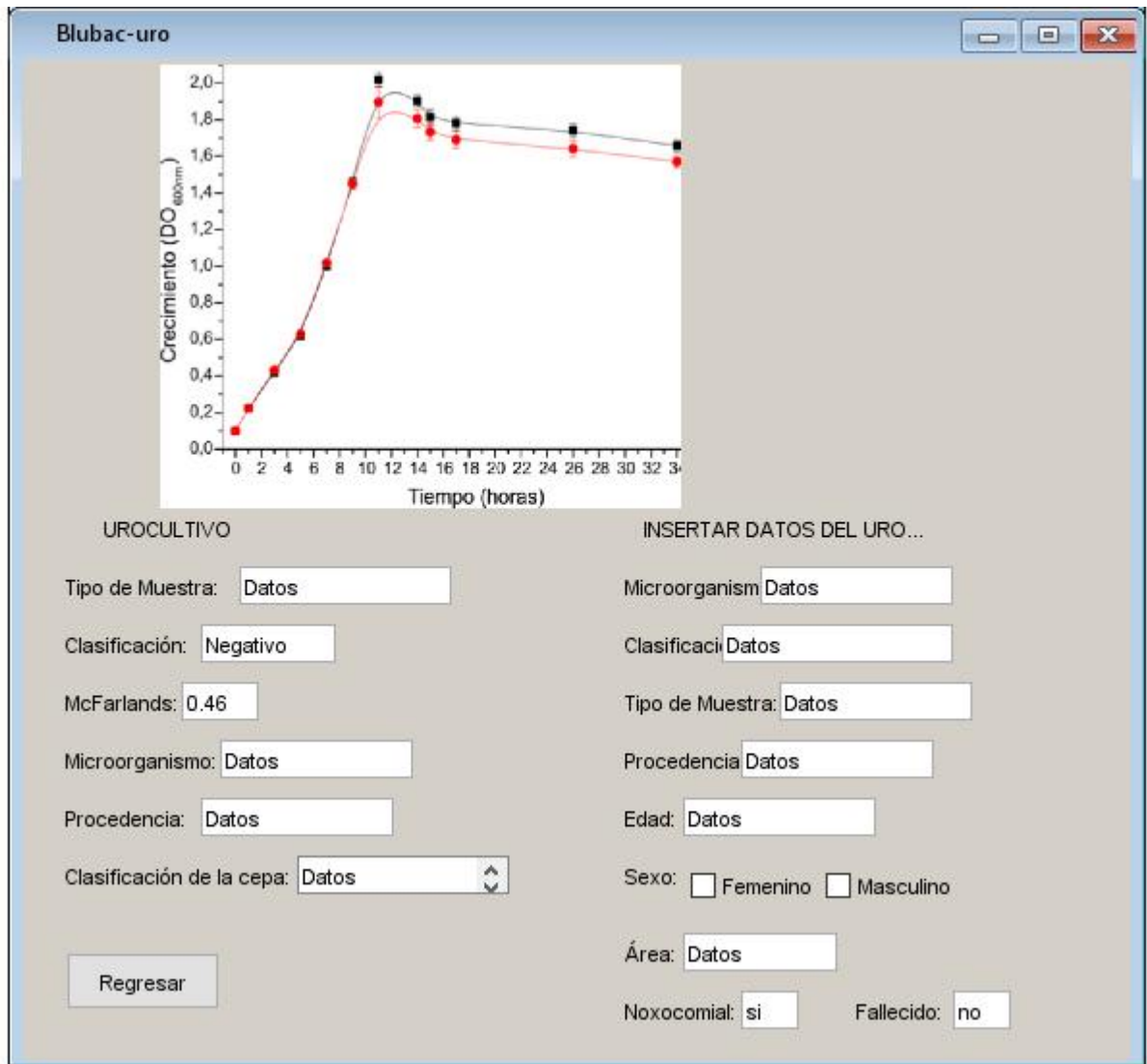
Prototipo de interfaz: insertar datos urocultivo

Sección 2 “Mostrar los datos de una muestra del urocultivos.”

	Actor	Sistema
	<i>1. Selecciona la opción mostrar datos del urocultivo dentro de la ventana emergente</i>	<i>2. Muestra la ventana que contine los datos de un urocultivo</i>

Flujos alternos	
Actor	Sistema

Prototipo de interfaz

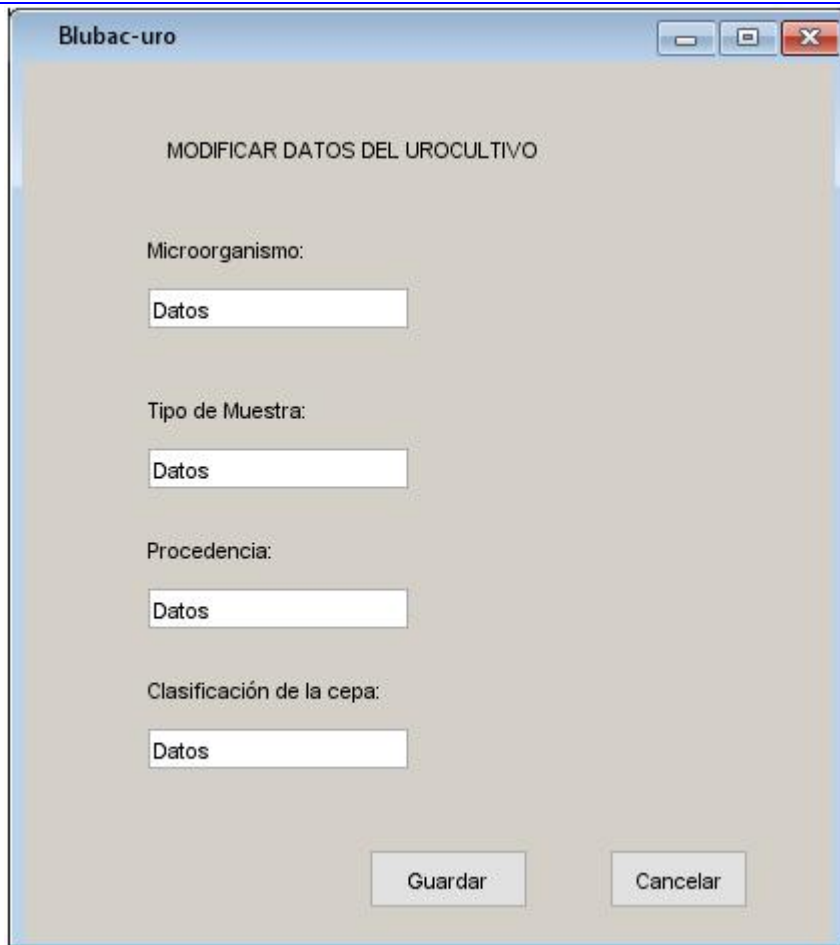


Prototipo de interfaz: mostrar datos del urocultivo

Sección 3 “Modificar los datos de una muestra de urocultivo”.

Actor	Sistema

	1. <i>Selecciona la opción modificar datos del urocultivo dentro de la ventana emergente</i>	2. <i>Muestra la ventana que contiene los datos de un urocultivo ya insertados</i>
	3. <i>Se modifican los datos del urocultivo deseado y se selecciona la opción guardar</i>	4. <i>Se registran los datos en la base de datos</i>
Flujos alternos		
	Actor	Sistema
	<i>3b. Selecciona la opción Cancelar.</i>	<i>5b. Se cierra la interfaz y no se registra ningún dato en la base de datos</i>
Prototipo de interfaz		



Prototipo de interfaz: modificar los datos del urocultivo

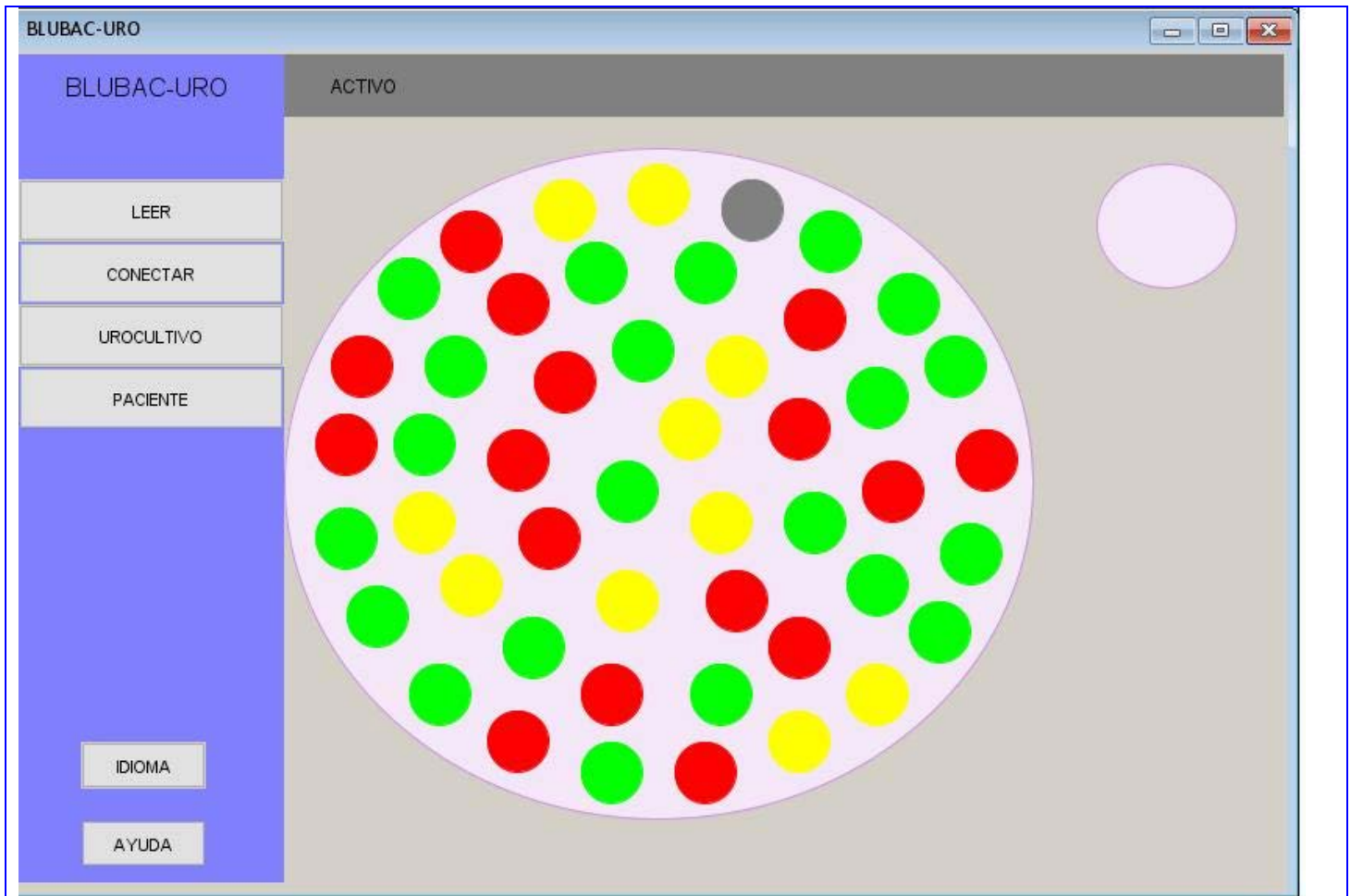
Sección 4 “Detener la lectura de la muestra seleccionada”

	Actor	Sistema
	<i>1.Selecciona la opción detener la lectura del urocultivo seleccionado lectura dentro de la ventana emergente</i>	<i>2.Se detiene el proceso de lectura del urocultivo poniéndose inactivo el urocultivo seleccionado</i>

Flujo Alternativo

	Actor	Sistema

Prototipo de Interfaz

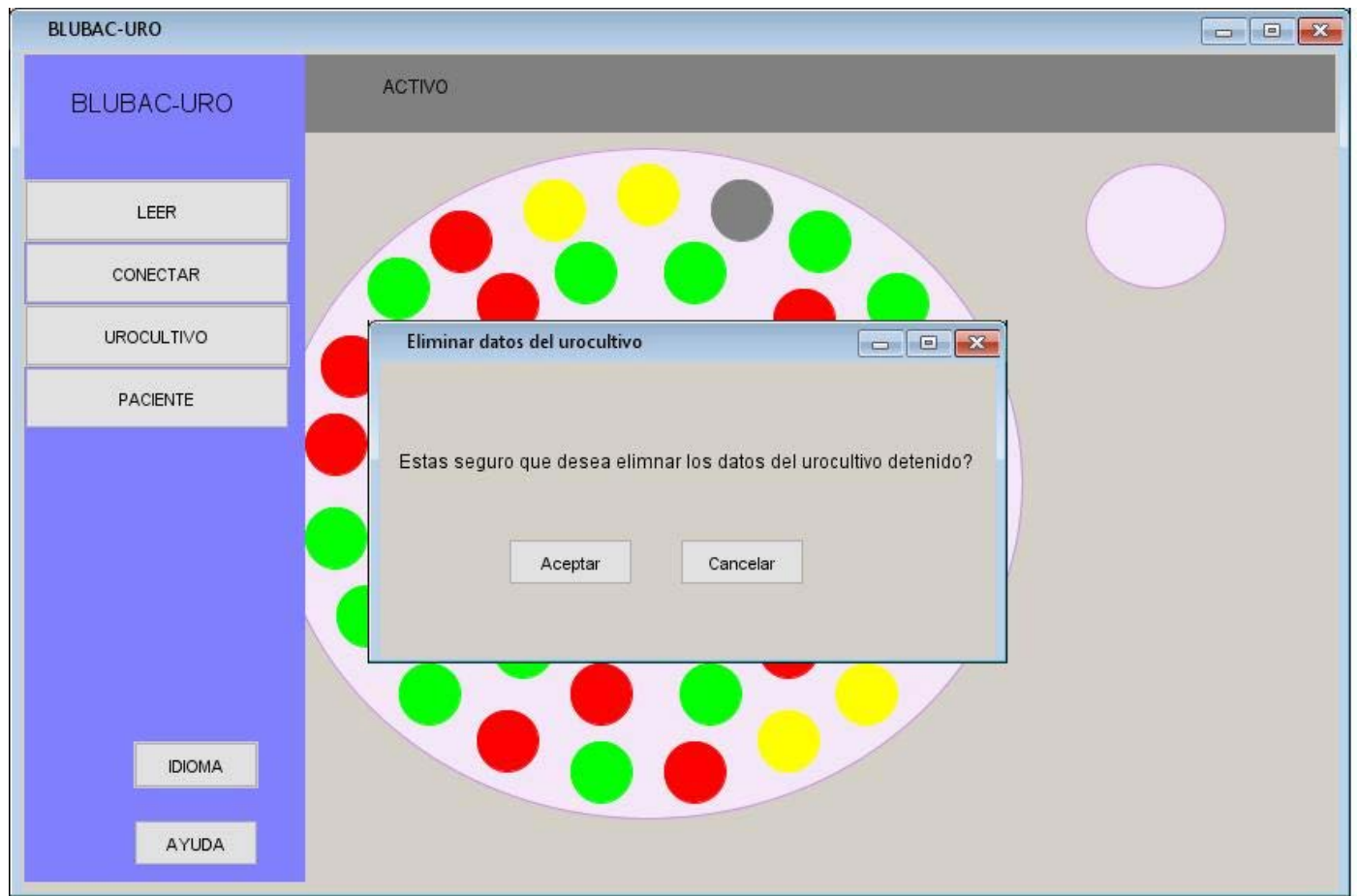


Prototipo de interfaz: detener la lectura

Sección 5 “Eliminar los datos de una muestra de urocultivo que fue detenida”.

	Actor	Sistema
	1. Selecciona la opción eliminar datos del urocultivo dentro de la ventana emergente	2. Muestra la ventana con el mensaje “ Estas seguro que desea eliminar los datos del urocultivo detenido”
	3. Se selecciona la opción Aceptar	4. Se eliminan los datos del urocultivo de la base de datos
Flujos Alternos		
	Actor	Sistema
	3b. Selecciona la opción Cancelar.	5b. Se cierra la interfaz y no se registra ningún dato en la base de datos

Prototipo de Interfaz



Prototipo de interfaz: eliminar los datos del urocultivo

Relaciones	CU Incluidos	<i>No procede</i>
	CU Extendidos	<i>No procede</i>
Requisitos funcionales	no	<i>No procede</i>

Patrones de diseño: Los patrones de diseño son descripciones de clases y objetos relacionados que están determinados para resolver un problema de diseño general en un determinado contexto. Estos nominan, abstraen e identifican los aspectos de una estructura de diseño lo que los hace útiles para crear un diseño orientado a objetos reutilizable. Estos

identifican las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades expresados como patrones (Cárdenas Escalante, 2014).

Durante el desarrollo del sistema a implementar se utilizaron patrones básicos de asignación de responsabilidades como Patrones Generales de Software para Asignar Responsabilidades (GRASP, del inglés, General Responsibility Assignment Software Patterns). Los patrones GRASP describen entre otros elementos la asignación de responsabilidades a objetos, expresados en forma de patrones (LÓPEZ 2004). Entre los patrones GRASP utilizados están:

Experto: Plantea que se debe asignar la responsabilidad de realizar determinada operación, a la clase que tiene la información necesaria para cumplir con dicha responsabilidad (LÓPEZ 2004).

En la solución propuesta este patrón se pone de manifiesto ya que se utiliza Hibernate como objeto relacional de mapeo (ORM por sus siglas en inglés) para la capa del modelo. Éste se encarga de crear una clase experta por cada tabla de la BD del modelo; lo que permite que se pueda manejar su información como un objeto de tipo la entidad mapeada.

Controlador: Consiste en tener una clase que sirva como intermediaria entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado (LÓPEZ 2004).

El sistema utiliza el “*EditarController*” como controlador frontal, y luego este a su vez realiza peticiones a “*UrocultivoController*” o “*GraficaController*” o “*PacienteController*” para acceder a los servicios que solicita a las interfaces repositorios que llaman a la entidad y retorna hacia atrás.

Creador: Este patrón se tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por la clase que contiene la información necesaria.

El sistema lo utiliza por ejemplo al crear un objeto de tipo paciente que solamente lo debe inicializar el controlador de paciente (“*PacienteController*”) que debe acceder al servicio paciente (“*PacienteService*”) que utiliza el repositorio paciente (“*PacienteRepository*”) que a su vez utiliza por debajo la entidad “*paciente*”.

Bajo acoplamiento: Es la medida en que los cambios de un componente tienden a necesitar cambios de otro componente.

Alta Cohesión: Asignar a las clases responsabilidades de forma tal que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.

Conclusiones

La realización del análisis del sistema a implementar propuso una representación visual de las clases conceptuales del entorno a través del modelo de dominio. Se identificaron 8 requisitos funcionales que fueron agrupados en 4 de uso del sistema, que el sistema debe cumplir para su correcto funcionamiento. Se realizó el modelo de diseño con el propósito de describir cómo se debe implementar el sistema y se identificaron los patrones de diseño: experto, creador, alta cohesión, bajo acoplamiento y controlador de la familia GRASP.

Referencias

- BERTONI, G.; P. PESSACQ, *et al.* Etiología y resistencia a antimicrobianos de la infección no complicada del tracto urinario *Medicina (Buenos Aires)*, 2017, 77: 304-308.
- DÍAZ, V. *Caracterización molecular de genes de resistencia a β -lactámicos en aislados clínicos de Escherichia coli provenientes de urocultivos y pruebas de inhibición con péptidos de Boana rosenbergi y Rana sp*, PUCE, 2017. p.
- GONZÁLEZ, J. Propuesta de algoritmo de clasificación genética *RCI*, 2013, Vol. 4 (No.2): 37-42.
- GONZÁLEZ, J. and O. MAR. *Algoritmo de clasificación genética para la generación de reglas de clasificación*, 2015. [Vol.8]. Disponible en: https://www.redib.org/recursos/Record/oai_articulo983540-algoritmo-clasificacion-genetica-generacion-reglas-clasificacion
- LEONARD, I.; B. VARGAS, *et al.* Las curvas de crecimiento de dos pastos introducidos en Ecosistemas Amazónicos *REDVET. Revista Electrónica de Veterinaria*, 2017, 18(7).
- LÓPEZ, E. T., A. O. RAMON, E. M. SARROCA AND C. G. SEONE. *Diseño de sistemas software en UML*. C.U.P. 2004. p. 8498800757

- MAR, O. and J. GULÍN Modelo para la evaluación de habilidades profesionales en un sistema de laboratorios a distancia *Revista científica*, 2018, 3(33): 332-343.
- 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.
- SOMMERVILLE, I. *Ingeniería del Software*. Pearson Educacion, S.A. 2005. p. 84-7829-074-5
- VANEGAS, D. and M. RAMÍREZ Correlación del Crecimiento de *Pseudomonas fluorescens* en la Producción de Polihidroxialcanoatos de Cadena Media (PHA MCL) mediante Modelos Primarios de Gompertz, Logístico y Baranyi *Información tecnológica*, 2016, 27(2): 87-96.
- WEITZENFELD, A. Ingeniería de software orientada a objetos con UML, Java e Internet *Cengage Learning Editores*, 2005.