

Sistema de Laboratorios Remoto para el estudio de la Microbiología y Parasitología Médica

Remote Laboratory System for the study of Medical Microbiology and Parasitology

Omar Mar Cornelio^{1*} 0000-0002-0689-6341
Bárbara Bron Fonseca² 0000-0001-9463-8408
Jorge Gulín González¹ 0000-0001-7912-2665

¹ Centro de Estudios de Matemática Computacional, Universidad de las Ciencias Informáticas, La Habana, Cuba.

² Departamento de Informática, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas, La Habana, Cuba.

*Autor para la correspondencia: omarmar@uci.cu

RESUMEN

Introducción: como parte del proceso de formación de enfermeros, médicos y tecnólogos de la salud son habilitados temas relacionados con la microbiología. Sin embargo, a partir del conjunto de medidas de seguridad y la disponibilidad de recursos físicos, no es posible el estudio de diversos microorganismos.

Objetivo: desarrollar un Sistema de Laboratorios Remoto para la práctica de Microbiología y Parasitología Médica.

Materiales y métodos: el sistema de Laboratorios Remoto posee un microscopio electrónico controlado mediante una interface de comunicación con un ordenador conectado a la red.

Resultados: se obtuvo como resultado un Sistema de Laboratorios Remoto que puede ser accedido mediante Internet o la red institucional. Facilita el estudio y la interpretación de diferentes muestras biológicas. Brinda un conjunto de reportes y estadísticas que permiten realizar análisis históricos de comportamiento.

Conclusiones: a partir del desarrollo de las prácticas de laboratorios a distancia, es posible el estudio de diferentes microorganismos sin riesgos biológicos para el estudiante.

Palabras claves: sistema de laboratorios remoto; microbiología; prácticas de laboratorio.

ABSTRACT

Introduction: as part of the training process for nurses, physicians and health technologists, topics related to microbiology are enabled. However, based on the set of security measures and the availability of physical resources, the study of various microorganisms is not possible.

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

Objective: to develop a Remote Laboratory System for the practice of the subject Medical Microbiology and Parasitology.

Methods: the Remote Laboratory System has an electronic microscope controlled by a communication interface with a computer connected to the network.

Results: a Remote Laboratory System that can be accessed through the Internet or the institutional network. The system facilitates the study and interpretation of different biological samples and also provides a set of reports and statistics that allow for historical behavior analysis.

Conclusions: from the development of remote laboratory practices, it is possible to study different microorganisms without biological risks for the student.

Keywords: remote laboratory system, microbiology, laboratory practices.

Recibido: 14/4/2020

Aprobado: 27/8/2020

Introducción

El estudio de microorganismos representa una actividad que deben desarrollar los médicos y tecnólogos de la salud. En el proceso de formación se transita por la asignatura de Microbiología y Parasitología médica durante la cual se identifican diferentes microorganismos ⁽¹⁾. Materializar las actividades prácticas constituye una acción que representa riesgos biológicos para los estudiantes en formación. Para evitar el riesgo, en las prácticas programadas se estudian microorganismos que no representan peligros para las personas, lo que imposibilita la interacción con diferentes fuentes de infecciones que también son de interés ⁽²⁾.

Una solución a este tipo de problema puede ser abordada actualmente desde diferentes perspectivas. Los Sistemas de Laboratorios Remoto (SLR) constituyen una alternativa a la problemática descrita en la que es posible el estudio de diferentes microorganismos de forma remota ^{(3), (4), (5)}, disminuyendo así el riesgo biológico.

Un SLR representa un lugar o entorno cuya función es realizar un control sobre un sistema físico a distancia, con el objetivo de tele operar un sistema real, realizar experimentos y acceder a los datos a través de la red ^{(6), (7)}. Los experimentos en los SLR son eficientes producto de que se encuentran disponibles sin limitaciones temporales y disminuyen el consumo de materiales ya que no precisan de equipamiento técnico en cada una de las infraestructuras docentes ^{(8), (9)}. Una estación de trabajo disponible en el SLR puede ser utilizada centralmente mediante Internet o intranet institucional. Los SLR representan una tecnología ideal para su aplicación en las Ciencias Médicas ⁽¹⁰⁾.

A partir de la problemática anteriormente descrita se define como objetivo de la presente investigación: desarrollar un Sistema de Laboratorios Remoto para la práctica de Microbiología y Parasitología Médica.

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

Método

La presente sección describe el diseño del Sistema de Laboratorios propuesto. La solución se encuentra dividida en dos aplicaciones, una aplicación de escritorio que conformará la estación de trabajo y una aplicación Web para el SLR. La aplicación de escritorio estará en la estación de trabajo que posee el microscopio electrónico y contará entre sus funcionalidades con la transmisión de un flujo de video. La aplicación Web se encargará de visualizar el flujo de video transmitido a partir de la tele operación. A continuación se describe detalladamente cada una de ellas:

1. Aplicación de escritorio (estación de trabajo): La aplicación de escritorio se encuentra desplegada en la estación de trabajo. Es la encargada de listar los dispositivos de tipo cámara que se encuentren conectados y permitir la visualización de uno de ellos. El dispositivo seleccionado debe encontrarse acoplado a un microscopio, lo que permite la observación de la muestra que se encuentre colocada en este. Otra de sus funcionalidades es la transmisión por la red del flujo de video que se encuentra emitiendo el dispositivo seleccionado. Además permite la grabación y captura de fotografías de las muestras colocadas en el microscopio seleccionado.
2. Aplicación Web (SLR): La aplicación Web permite integrar una estación de trabajo al SLR. Esta se encarga de visualizar los flujos de video que se encuentran siendo transmitidos por la aplicación de escritorio desde las estaciones de trabajo. Además de permitir capturar y guardar fotografías del flujo de video que se encuentra visualizando.

La solución propuesta, permite a los educandos de universidades y otros centros educativos del país potenciar la adquisición de conocimientos en la rama de la microbiología, al poder observar el progreso de muestras reales en tiempo real, además de realizar capturas de fotos de la mismas, todo esto haciendo únicamente uso de un navegador Web.

Los principales conceptos que conforman el entorno que se modela son representados en un modelo conceptual ⁽¹¹⁾. La figura 1 muestra un esquema que ilustra la interacción de las diferentes entidades del negocio.

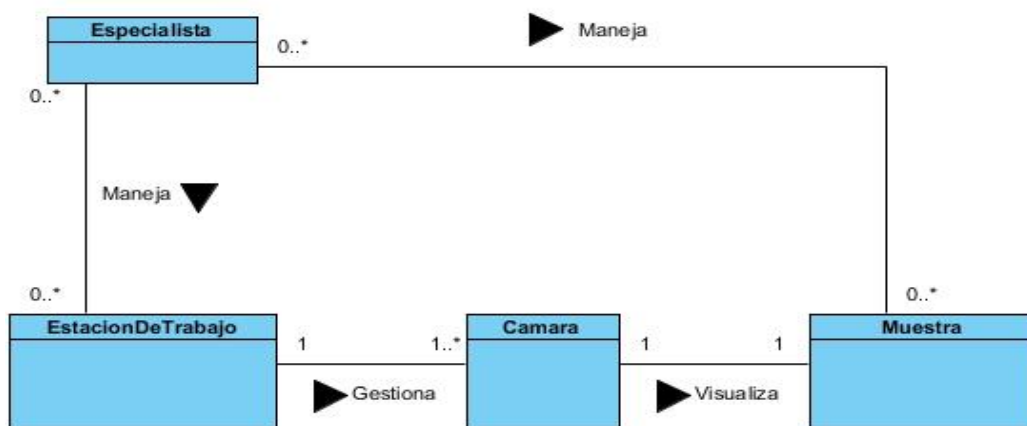


Fig. 1- Modelo conceptual.

Para un mejor entendimiento del modelo conceptual se realiza una descripción de las clases existentes:

Especialista: Especialista que maneja la estación de trabajo y las muestras.

Estación de trabajo: Computadora con acceso a la red a la que se encuentra conectada la cámara que esta acoplada al microscopio.

Cámara: Dispositivo que se encarga de la emisión de un flujo de video.

Muestra: Pequeña porción de microorganismos o de tejido celular que se visualizará con el microscopio.

El proceso de observación de la muestra se inicia cuando el especialista coloca la muestra en el microscopio. La cámara se encuentra acoplada al microscopio electrónico y se encarga de visualizar la muestra colocada en este. El especialista manipula la cámara a través de la estación de trabajo, mediante la cual puede visualizar la muestra que esté colocada en el microscopio.

Como parte de las restricciones de servicios o funciones ofrecidas por el sistema, se definen los requisitos no funcionales ⁽¹²⁾. Los requisitos no funcionales se enfocan fundamentalmente en las características del sistema y las condiciones que se deben cumplir para su correcto funcionamiento ^{(13), (14)}. Para la presente investigación se definen requisitos de usabilidad, software, hardware, interfaz y diseño ^{(15), (16), (17)}.

Se definen como requisitos de usabilidad:

1. El sistema en su totalidad está diseñado para que el usuario realice las operaciones en la menor cantidad de eventos posibles.
2. La solución posee un diseño sencillo que permite potenciar la facilidad de aprendizaje, tanto para aprender sus funcionalidades básicas, como para realizar correctamente la tarea que desea ejecutar y poder aplicarla en el futuro.

Se definen como requisitos de software para el funcionamiento del sistema:

<http://scielo.sld.cu>



1. PC Cliente de la aplicación de escritorio: Sistema Operativo Linux. Biblioteca LibVLC en su versión 2.2.0.
2. Servidor de aplicaciones: Sistema Operativo Ubuntu 12.04, debe contar con el servidor Web Apache Tomcat 7.0.1 con la aplicación Liferay Portal 6.0.5, JDK6, PostgreSQL 9.4, Flumotion Streaming Server en su versión 0.10.
3. PC Cliente Web: Las estaciones clientes para acceder al sistema deben utilizar como navegador Mozilla Firefox versión 36.0, Google Chrome 40.0, Opera 20.0, Safari 5.1 o cualquiera de las versiones superiores.

Para el correcto funcionamiento de sistema se definen los requisitos de hardware:

1. PC Cliente de la aplicación de escritorio: 1 Gb de RAM o superior, procesador Dual Core a 2.20 GHz o superior. Tarjeta de red con una velocidad de 100 Mbps o superior.
2. Servidor de aplicaciones: 2Gb de memoria RAM, procesador Dual Core a 2.20 GHz o superior. Tarjeta de red con una velocidad de 100 Mbps o superior.
3. PC Cliente Web: 256 Mb de RAM o superior, procesador Pentium IV o superior. Tarjeta de red con una velocidad de 100 Mbps o superior.

Se definen como requisitos de interfaz:

1. El sistema está diseñado para que se visualicen las alarmas con el color llamativo rojo.
2. Cuando se ordene una acción en cualquiera de los componentes del sistema debe mostrarse un icono que indique la acción que se encuentra ejecutándose: grabando o transmitiendo o ambas. En el caso de encontrarse transmitiendo, se muestran otros detalles significativos tales como el puerto y la URL.

Se definen como requisitos de diseño:

1. Lenguaje de programación: Se emplea en el desarrollo de la aplicación de escritorio el lenguaje C++, haciendo uso del framework QT Creator en su versión 3.0, mientras que para el lado del cliente Web se utilizará el lenguaje Java Server Page (JSP), haciendo uso del framework Spring.
2. Bibliotecas: Para el correcto funcionamiento de la aplicación de escritorio se utiliza la biblioteca LibVLC en su versión 2.2.0.

Para transformar los conceptos descritos, se realiza una agrupación de los Requisitos Funcionales. Los requisitos funcionales son declaraciones de servicios que el sistema debe proporcionar, define la manera en que éste debe reaccionar a determinadas entradas y cómo se debe comportar en situaciones particulares⁽¹²⁾. Cada requisito representa características que debe cumplir el SLR propuesto y expresan la capacidad de acción del mismo.

Para definir las funcionalidades que pueden ejecutar los usuarios, se definen los actores del sistema. Para el Sistema de Laboratorios propuesto se determina su nivel de interacción con

<http://scielo.sld.cu>



cada caso de uso. La tabla 1 realiza una descripción de los actores del sistema y sus descripciones.

Tabla 1-Descripción de actores del sistema

Actor del Sistema	Descripción
Especialista	El especialista interactúa con la aplicación de escritorio, permitiéndole seleccionar de la lista de cámaras disponibles la que desea visualizar y transmitir, además de permitirle capturar fotos y grabaciones de video, y configurar las propiedades de dichas capturas.
Usuario	El usuario tiene permitido seleccionar de la lista de flujos de video, el que desea visualizar, además de capturar fotogramas y almacenar el mismo en un directorio de su preferencia, todo a través de la aplicación Web.
Administrador	Tiene permitido seleccionar de la lista de flujos de video, el que desea visualizar, capturar fotogramas y almacenarlos en un directorio de su preferencia. Además puede insertar, modificar o eliminar el origen del flujo de video, todo a través de la aplicación Web.

Un Diagrama de Casos de Uso de Sistema (DCUS) muestra la relación entre los actores y los casos de uso del sistema, de esta forma se representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa ⁽¹⁸⁾. La figura 1 muestra el DCUS.

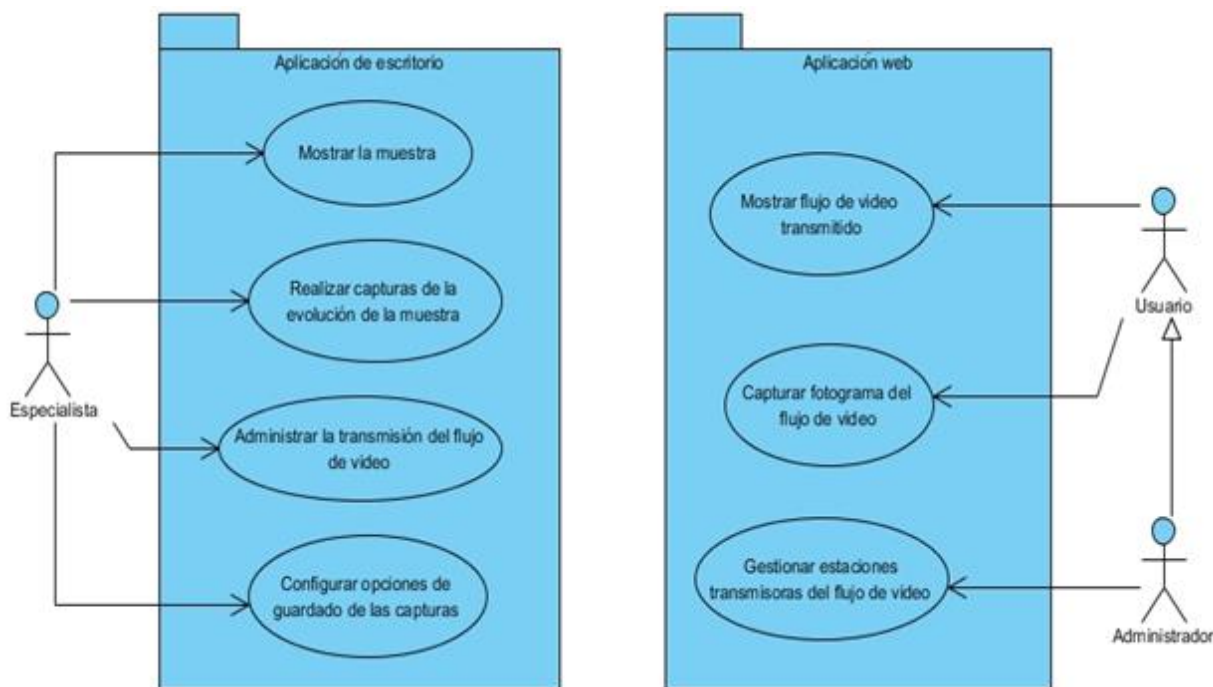


Fig. 2- Diagrama de Casos de Uso del Sistema.

A continuación se relacionan los Requisitos Funcionales y los Casos de Uso del Sistema teniendo en cuenta la aplicación en la cual se encuentran.

Agrupación de Casos de Usos del Sistema de la aplicación de escritorio (estación de trabajo)

Caso de Uso 1. Mostrar la muestra:

Requisito Funcional 1: Listar cámaras disponibles para su uso.

Requisito Funcional 2: Seleccionar una cámara de la lista de disponibles

Requisito Funcional 3: Visualizar la muestra con la cámara seleccionada.

Caso de Uso 2. Realizar capturas de la evolución de la muestra:

Requisito Funcional 4: Capturar fotografía de la muestra con la cámara seleccionada.

Requisito Funcional 5: Guardar la fotografía capturada.

Requisito Funcional 6: Iniciar grabación de la evolución de la muestra con la cámara seleccionada.

Requisito Funcional 7: Detener grabación de la cámara.

Requisito Funcional 8: Guardar video grabado de la evolución de la muestra.

Caso de Uso 3. Administrar la transmisión del flujo de video:

Requisito Funcional 9: Configurar los parámetros del video para la transmisión.

Requisito Funcional 10: Iniciar la transmisión del flujo de video emitido por la cámara seleccionada.

Requisito Funcional 11: Detener la transmisión del flujo de video.

Caso de Uso 4. Configurar opciones de guardado de las capturas:

Requisito Funcional 12: Configurar las opciones de guardado de las capturas.

Definición de Casos de Usos del Sistema de la aplicación Web (SLR)

Caso de Uso 5. Mostrar flujo de video transmitido:

Requisito Funcional 13: Listar las estaciones transmisoras disponibles.

Requisito Funcional 14: Seleccionar una estación transmisora de la lista de disponibles.

Requisito Funcional 15: Visualizar flujo de video de la estación transmisora seleccionada.

Caso de Uso 6. Capturar fotograma del flujo de video:

Requisito Funcional 16: Capturar fotograma del flujo de video de la estación seleccionada.

Requisito Funcional 17: Guardar fotograma tomado del flujo de video de la estación seleccionada.

<http://scielo.sld.cu>



Caso de Uso 7. Gestionar estaciones transmisoras del flujo de video:

Requisito Funcional 18: Insertar una estación transmisora de un flujo de video.

Requisito Funcional 19: Actualizar los datos de una estación transmisora de un flujo de video.

Requisito Funcional 20: Eliminar una estación transmisora de un flujo de video.

Resultados

Los diagramas de clases del diseño permiten modelar la vista de diseño estática de un sistema, estos describen gráficamente las especificaciones de las clases además de visualizar las relaciones entre estas. A continuación se muestran los diagramas de clases del diseño para los CU: “Administrar la transmisión del flujo de video” y “Mostrar flujo de video transmitido”.

Caso de Uso “Administrar la transmisión del flujo de video”

En el diagrama de clases del diseño referente al CU “Administrar la transmisión del flujo de video” (Fig. 3), se identifican dos paquetes: Presentación y Negocio, los que corresponden al patrón de diseño N-Capas, en su versión dos capas.

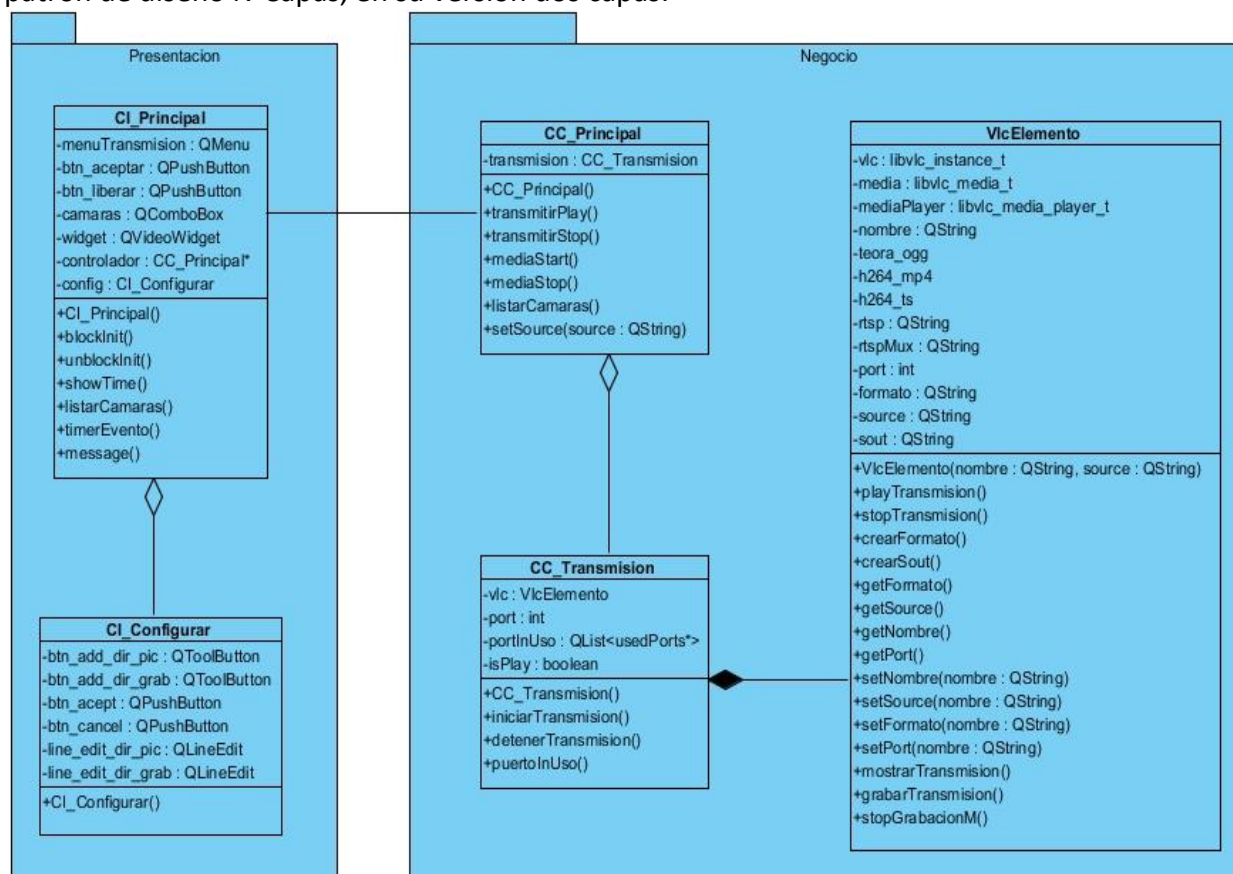


Fig.2- DCUS “Administrar la transmisión del flujo de video”

La clase CI_Principal es la clase a través de la cual el rol especialista puede acceder a todas las funcionalidades. Posee un objeto de tipo CC_Principal que es la clase controladora. La clase CC_Principal es la encargada de recibir las peticiones que le son enviados desde la clase interfaz, esta redirecciona a la clase controladora relacionada con la petición realizada, en este caso la clase CC_Transmisión, la cual realiza la transmisión del video a través del protocolo RTSP, haciendo uso de un objeto de tipo VlcElemento.

Caso de Uso “Mostrar flujo de video transmitido”

En el diagrama de clases del diseño referente al CU “Mostrar flujo de video transmitido” (Fig. 4), se identifican tres paquetes: Vista, Controlador y Modelo, los que corresponden a las capas del patrón de diseño MVC.

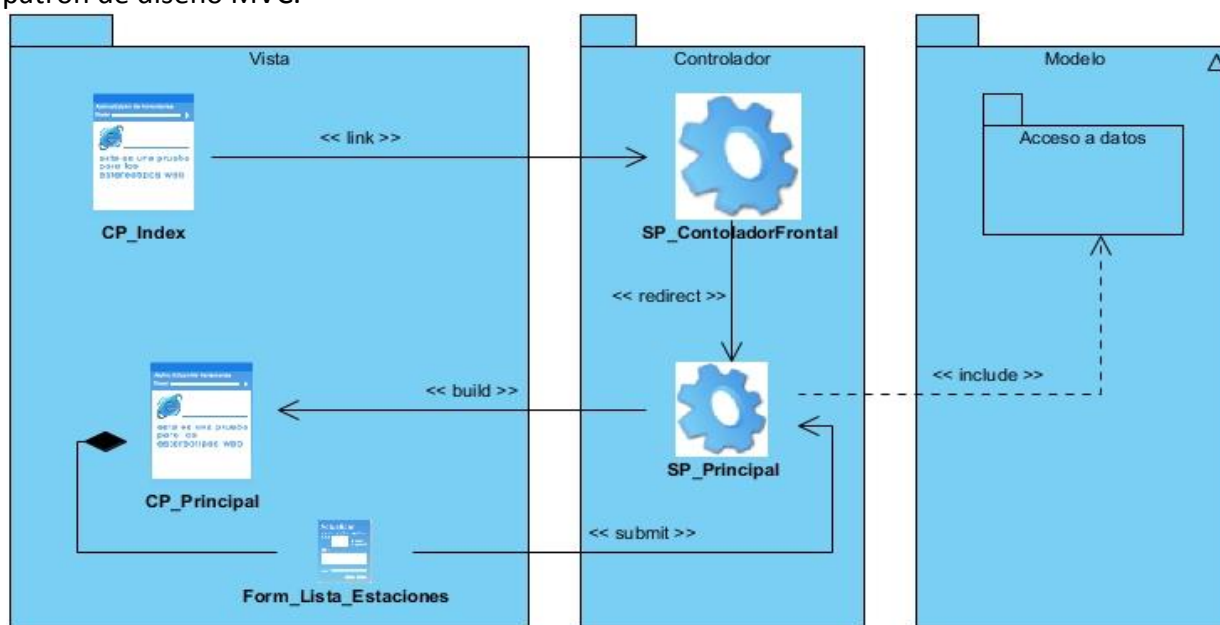


Fig. 3- DCUS “Mostrar flujo de video transmitido”

El usuario accede a la interfaz principal, “CP_Principal”, a través de la clase “CP_Index”, esta genera una solicitud que es enviada a la clase “SP_ControladorFrontal”, la cual la redirecciona a la clase “SP_Principal” que es la encargada de la lógica del negocio relacionada con la “CP_Principal”. La clase “CI_Principal” se construye con una lista de las estaciones que se encuentren disponibles, además contiene todas las funcionalidades necesarias para la visualización de los flujos de video.

En el diagrama de clases del diseño del CU “Mostrar flujo de video transmitido” se encuentra el paquete “Acceso a datos” (Fig.5), el cual define la clase “DAO”. Esta contiene todos los métodos necesarios para el acceso y gestión de las tablas de la base datos, utilizando para ello las clases

“Acceso_Estación” y “Estación”. Esta clase implementa el patrón de diseño DAO, ya que se encarga de la gestión de los datos entre las clases que contienen la lógica del negocio y las entidades.

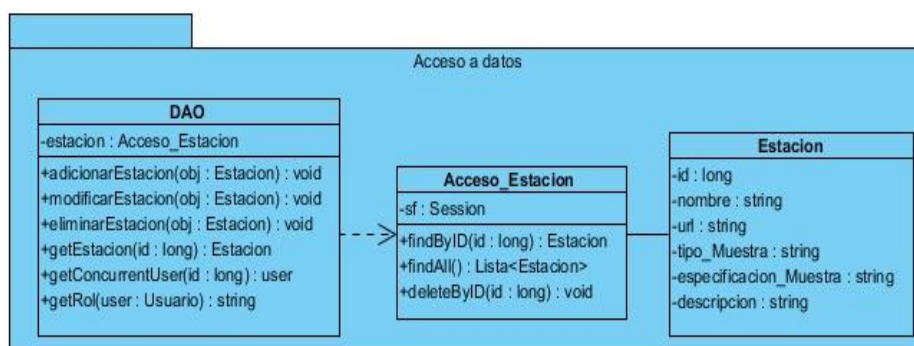


Fig.4- Paquete "Acceso a datos"

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Representa como cada subsistema es dividido en componentes y muestra las dependencias entre ellos. A continuación se muestra el diagrama de componentes para los CU: “Administrar la transmisión del flujo de video” (Fig. 6) y “Mostrar flujo de video transmitido” (Fig. 7).

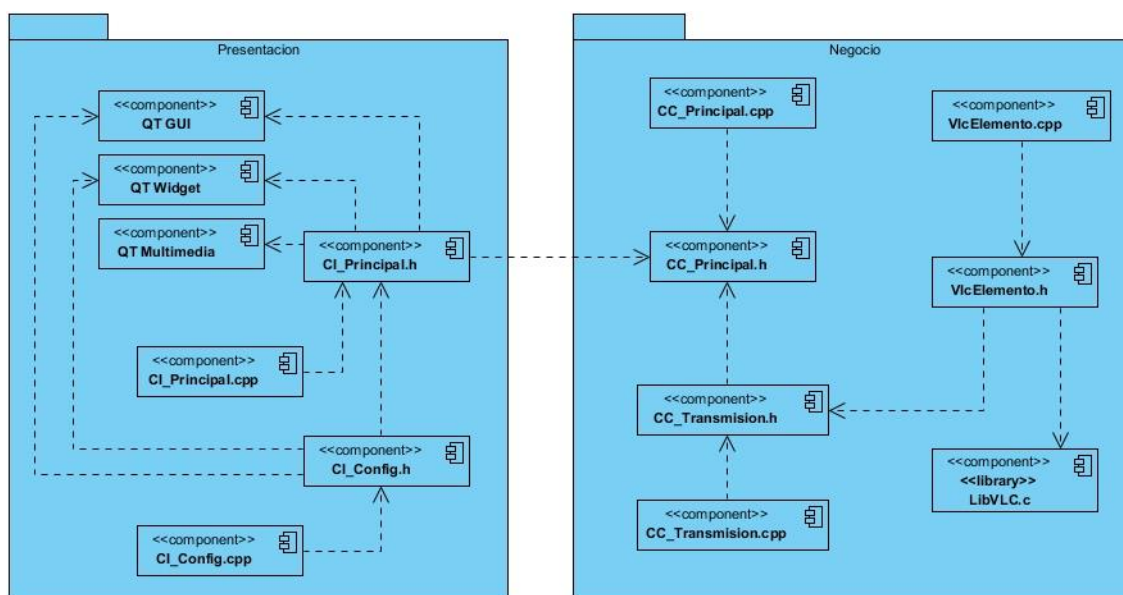


Fig.5-Diagrama de componentes CU "Administrar la transmisión del flujo de video"

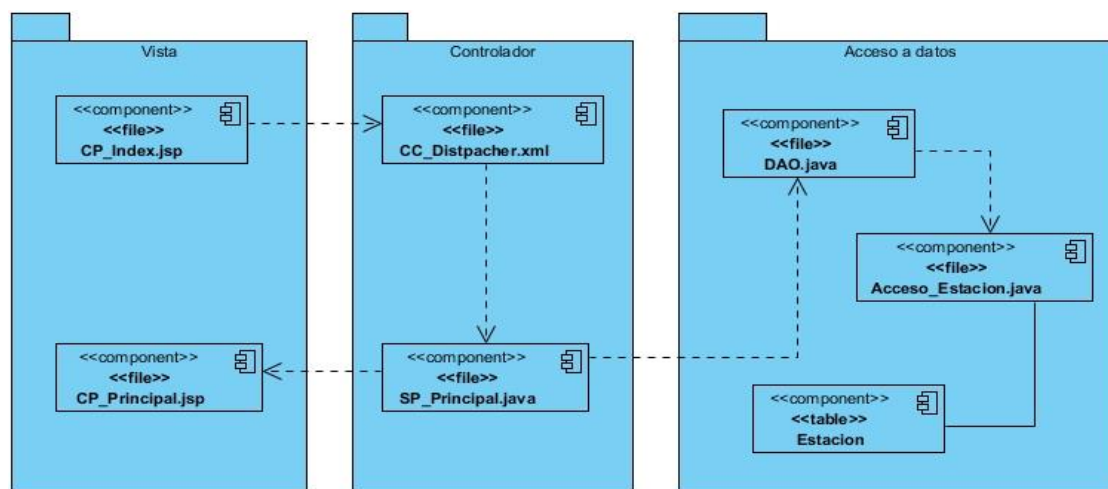


Fig. 6- Diagrama de componentes CU "Mostrar flujo de video transmitido"

La figura 8 muestra el diagrama de despliegue para el Sistema de Laboratorios Remoto propuesto. Representa el grafo de nodos unidos por conexiones de comunicación que muestra las relaciones físicas entre los componentes hardware y software ⁽¹⁹⁾.

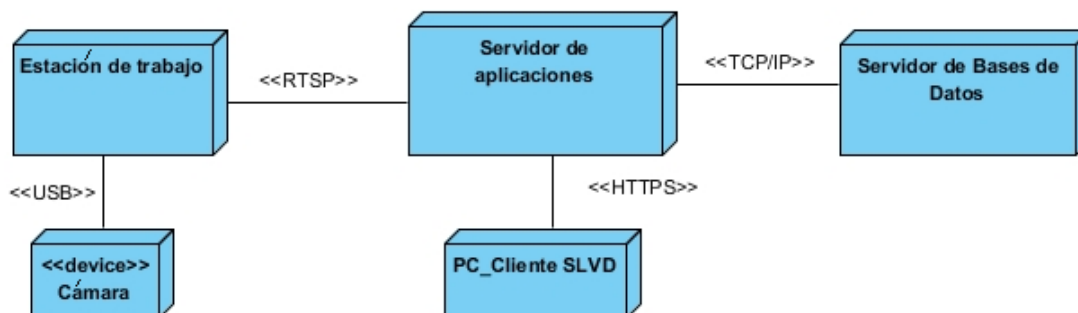


Fig. 7- Diagrama de Despliegue.

A continuación se describe cada uno de los nodos que conforman el diagrama de despliegue:

1. Cámara: Dispositivo físico encargado de emitir flujos de video.
2. Estación de trabajo: Terminal con acceso a la red en la cual se encuentra desplegada la aplicación de escritorio desarrollada. Por otro lado la estación de trabajo debe contar con la librería LibVLC en su versión 2.2.0 instalada, la cual es utilizada por la aplicación de escritorio para la transmisión del flujo de video de la cámara haciendo uso del protocolo RTSP.
3. Servidor de aplicaciones: Terminal con acceso a la red donde se halla alojado el servidor Web Apache Tomcat en su versión 7.0.1, el cual cuenta con el contenedor de portlet Liferay Portal,

<http://scielo.sld.cu>



versión 6.1, en este se encuentra desplegado el portlet para la microbiología. Además cuenta con un servidor de medias, el cual se encarga de todo lo relacionado con la recepción y posterior transmisión del flujo de video, disminuyendo la carga del servidor Web.

4. Servidor de bases de datos: Terminal con acceso a la red donde se encuentra el servidor de bases de datos PostgreSQL en su versión 9.4, al cual se accede a través del puerto 5432, haciendo uso del protocolo TCP/IP.

5. PC cliente del SLR: Terminal conectada a la red, mediante la cual el cliente accede al portlet de microbiología en el SLR a través de un navegador Web.

El dispositivo de tipo cámara que se encuentra conectado a la Estación de trabajo emite un flujo de video, el cual es decodificado y transmitido por la aplicación de escritorio desarrollada haciendo uso del protocolo RTSP. Este flujo de video es recibido en el servidor de aplicaciones por el servidor de medias y colocado bajo demanda. El cliente accede a través de una computadora al portlet alojado en el SLR, a través del protocolo HTTPS, el cual le permite visualizar los flujos de video que se encuentran disponibles y han sido previamente colocados en la bases de datos, a la cual se accede a través del protocolo TCP/IP.

Cada uno de los componentes definido fue implementado para obtener la solución propuesta. El proceso de desarrollo de los mismos fue guiado por los diagramas de clases y se respetaron las características de cada uno de los patrones arquitectónicos escogidos según el componente. A continuación se describe cada uno de estos componentes.

La Figura 9 muestra la descripción por áreas de las interfaces de usuario para la aplicación de Escritorio (Estación de trabajo).

1. Panel de visualización: Área donde se muestra la visualización de la cámara.
2. Opciones de configuración: Área que permite entrar a la interfaz de configuración.
3. Opciones de transmisión: Permite al usuario acceder a las opciones correspondientes con la transmisión.
4. Opciones de realizar capturas: Permite a los usuarios realizar fotografía o grabaciones con la cámara escogida.
5. Lista de cámaras: Muestra la lista de cámaras disponibles para su utilización. El usuario puede escoger una de estas y visualizarla.
6. Detalles y notificaciones: En esta área se muestran al usuario las notificaciones del sistema. Se muestra además información adicional sobre la transmisión (puerto y dirección IP por el que se transmite, URL utilizada), de encontrarse realizando una.

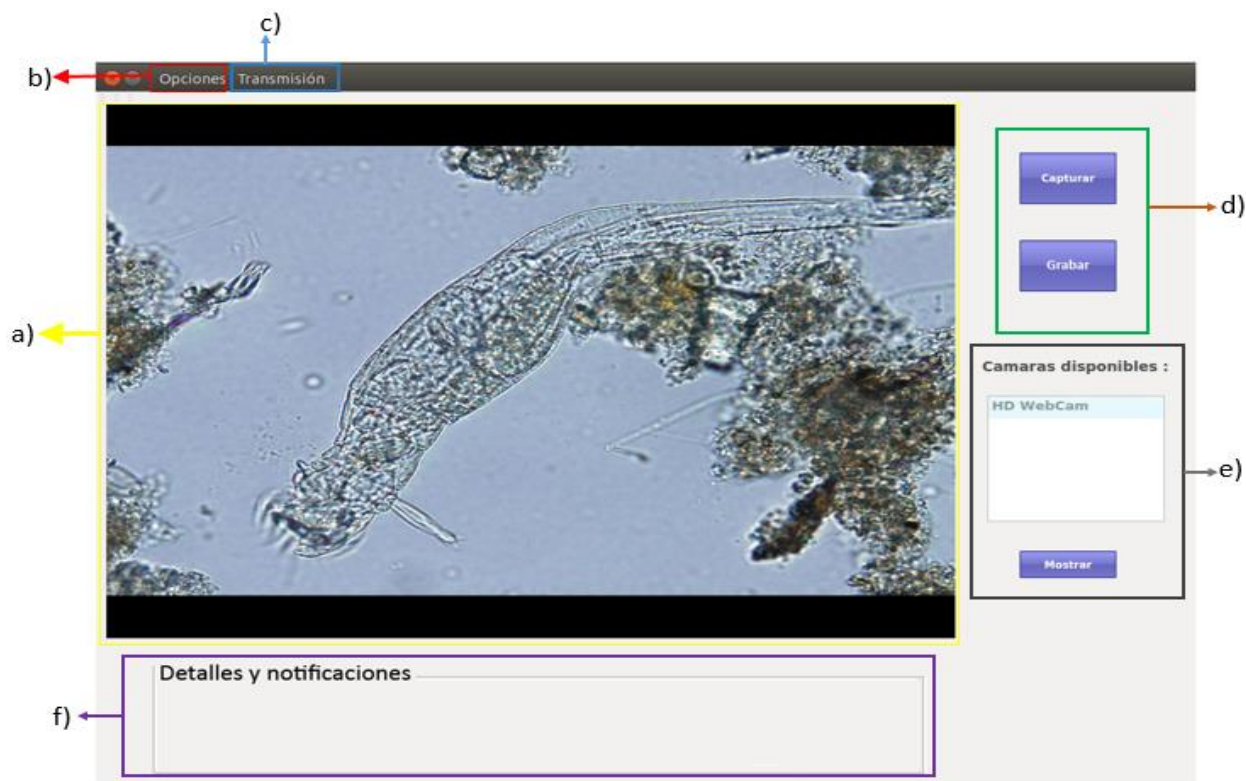


Fig. 9- Interfaz gráfica de la Aplicación de Escritorio

La Figura 10 muestra la descripción por áreas de las interfaces de usuario para la Aplicación Web.

1. Panel de visualización: Área donde se muestra el flujo de video que se encuentre transmitiendo la estación escogida.
2. Opciones para realizar captura: Permite al usuario realizar capturas el flujo de video que se encuentre visualizando.
3. Lista de estaciones: Muestra la lista de estaciones disponibles para su visualización.
4. Opciones de origen: Esta área solo podrá visualizarla un usuario con permisos de administración pues permite la gestión de la lista de estaciones accesibles.

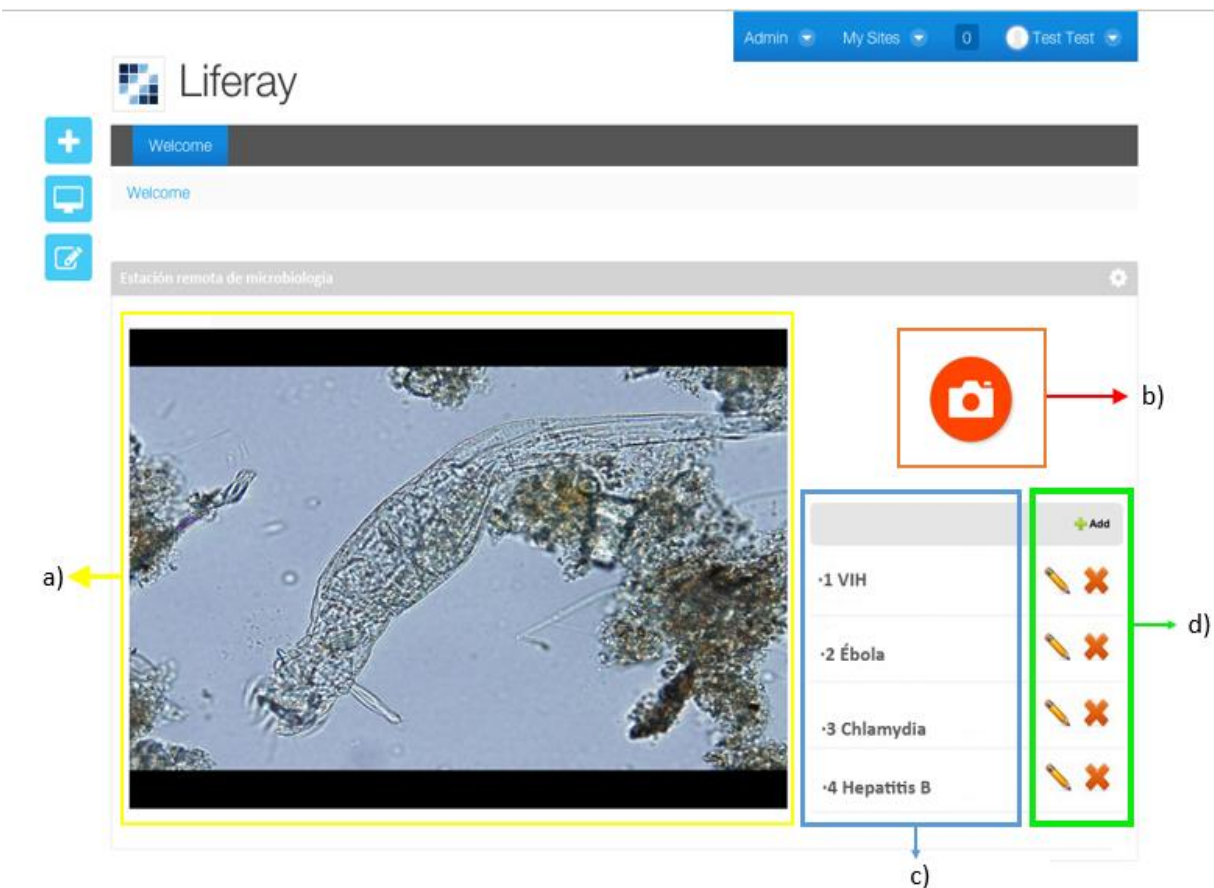


Fig. 10- Interfaz gráfica de la Aplicación Web.

Para el correcto funcionamiento del sistema propuesto se utilizan estándares de codificación. Los estándares de codificación comprenden todos los aspectos de la generación de código, este refleja un estilo armonioso estableciendo como operar con la base de código existente. Estos facilitan que el código generado en la fase de implementación se encuentre bien organizado, sea comprensible, sencillo y rápido de detectar errores. Se logra así una aplicación eficiente en términos de mantenimiento y extensión de acuerdo a las nuevas necesidades que surjan en el negocio.

En el desarrollo de la solución planteada se utilizan los siguientes estándares de codificación:

lowerCamelCase: Es una convención de nombres que forma parte de CamelCase en el que el nombre se forma de múltiples palabras que se unen como una sola donde la primera letra del nombre comienza con mayúscula. Específicamente en el caso de lowerCamelCase la primera letra del nombre comienza con minúscula. La figura 11 muestra un fragmento de código donde se muestra el estándar definido.

```

void VlcElemento::crearSout ()
{
    mySout.clear ();
    QString auxSout="";
    auxSout.append (soutFormat) ;
    auxSout.append (soutProtocol) ;
    mySout=auxSout.arg (w,h) ;
}

```

Fig. 8- Fragmento de código de la clase: "VlcElemento"

UpperCamelCase: Esta convención de nombres también forma parte de CamelCase, pero específica que la primera letra es con mayúscula, siendo fácil distinguirla de los nombres lowerCamelCase. Se utiliza mayormente para definir los nombres de las clases, a continuación se muestra un ejemplo. La figura 12 muestra un fragmento de código donde se implementa el estándar.

```

private:
    CC_Transmision *transmision;
    CC_Media *media;
    CC_Media *getMedia ();
    CC_Transmision *getTransmision ();
    QString name;
    QString source;
    QString format;
    int port;

```

Fig. 92- Fragmento de código de la clase: "CC_Principal"

Sentencias de inclusión o importación: No existen líneas en blanco entre estas. A continuación se muestra un ejemplo en la clase "VlcElemento". La figura 13 muestra un ejemplo del empleo del estándar.

```

#include "vlc/vlc.h"
#include "vlc/libvlc.h"
#include "vlc/libvlc_events.h"
#include "vlc/libvlc_media.h"
#include "vlc/libvlc_vlm.h"
#include <QObject>
#include "QDebug"
#include <QString>

```

Fig. 103- Fragmento de código de la clase: "VlcElemento"

Conclusiones

La caracterización y revisión del estado del arte actual de los Sistemas de Laboratorios Remoto y sus estaciones de trabajo permitió identificar la necesidad de implementar una solución que permita el desarrollo de prácticas de Microbiología y Parasitología Médica.

Con el desarrollo de la solución propuesta se introduce un Sistema de Laboratorios Remoto para la práctica de Microbiología y Parasitología Médica disponible para las instituciones educativas del país. Permite potenciar la adquisición de conocimientos por parte de los estudiantes al poder estos visualizar, desde cualquier terminal con acceso a la red, la evolución de las muestras en tiempo real.

A partir del proceso de diseño de la solución propuesta se obtuvieron los artefactos generados para el proceso de desarrollo que facilita continuar su implementación en el futuro y adecuarla a diferentes contextos.

Referencias

1. Espino Hernández M, Abín Vázquez L, Silva Reyes M, Álvarez González MM, Díaz Suárez LA, Alemán Mondeja L. Evaluación de una estrategia docente para las prácticas de laboratorio de Microbiología y Parasitología Médica en Medicina. *Educ med super*. 2011;25(4):438-50.
2. Escalante Collazo GE, González Argote J, García Rivero AA. Microbiología y parasitología médicas en el contexto epidemiológico. *FEM: Revista de la Fundación Educación Médica*. 2018;21(1):55-6.
3. Cruz Carballosa Y, Codorniú Pérez X, Torres Rojas L. MicrobiologíaSoft, entrenador de Microbiología y Parasitología médica. *Rev cuba inform méd [Internet]*. 2017 [citado 29 Jul 2020];9(1):61-72. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592017000100007.
4. Cabrera IdJH, Motas IFM, Hernández MJV, Suárez LAD, Páez IMV, González MMÁ. Recurso didáctico para la enseñanza de la asignatura Microbiología y Parasitología Médicas/Knowledge of the harmful effects of ICT of students of the Latin American School of Medicine. *Panorama Cuba y Salud*. 2020;15(1):18-21.
5. Rivera Fernández N, García Dávila P, Alpuche Hernández A. Las aplicaciones digitales como herramienta didáctica para el estudio de la Parasitología Médica. *Investigación en educación médica*. 2019;8(31):64-71.
6. Milner DA, Holladay EB. Laboratories as the Core for Health Systems Building. *Clinics in Laboratory Medicine*. 2018;38(1):1-9.
7. Jin X, Kumar L, Li Z, Feng H, Xu X, Yang G, et al. A review of data assimilation of remote sensing and crop models. *European Journal of Agronomy*. 2018;92:141-52.

<http://scielo.sld.cu>



8. Suárez LAD, Mondeja LDA, González MMÁ. Evaluación de un hiperentorno de aprendizaje sobre Virología Médica en la disciplina Microbiología y Parasitología Médicas. Escuela Latinoamericana de Medicina, 2012-2013. Panorama Cuba y Salud. 2016;11(2):30-42.
9. Mar Cornelio O, Cabrera Rubido MB. Práctica de Microbiología y Parasitología Médica integrado al Sistema de Laboratorios a Distancia en la carrera de Medicina. Revista de Ciencias Médicas de Pinar del Río. 2016;20(2):9-19.
10. Luciano MI, Notario R, Gambandé T, Aita J. Microbiología: desafío en la enseñanza-aprendizaje en la formación del médico. Revista Médica de Rosario. 2019;85(3):128-37.
11. Falgueras BC, Catalunya UOd. Ingeniería del software: Universitat Oberta de Catalunya; 2002.
12. Pressman RS. "Software Engineering". 7ma. ed: New York: Higher Education; 2010.
13. Buitrón SL, Flores-Rios BL, Pino FJ. Elicitación de requisitos no funcionales basada en la gestión de conocimiento de los stakeholders. Ingeniare Revista chilena de ingeniería. 2018;26(1):142-56.
14. Molina Hernández Y, Granda Dihigo A, Velázquez Cintra A. Los requisitos no funcionales de software. Una estrategia para su desarrollo en el Centro de Informática Médica. Rev cuba inform méd [Internet]. 2019 [citado 29 Jul 2020];13(2):77-90. Disponible en: <http://scielo.sld.cu/pdf/rcci/v13n2/2227-1899-rcci-13-02-77.pdf>.
15. Ortiz Matute MA, Villanueva H, Yerak B. Evaluación de metodologías de desarrollo web bajo el paradigma de desarrollo dirigido por modelos (MDD) con la integración de directrices para la captura de requisitos de usabilidad medido por la ISO/IEC 9126 para lograr la satisfacción del cliente[tesis carrera]. Universidad Señor del Sipán: Perú; 2017.
16. Navarro ME, Moreno MP, Aranda J, Parra L, Rueda JR, editors. Arquitectura de software en el proceso de desarrollo ágil: una perspectiva basada en requisitos significantes para la arquitectura. XX Workshop de Investigadores en Ciencias de la Computación (WICC) 2018. Argentina: Universidad Nacional del Nordeste; 2018. pp. 635-8.
17. Cardozzo DR. Desarrollo de Software: Requisitos, Estimaciones y Análisis. IT Campus Academy; 2016. 128 p.
18. Jacobson I, Booch G, Rumbaugh J. El proceso unificado de desarrollo de software. Addison Wesley; 2004. 458 p.
19. Hernández D, Bahena C. Arquitectura para el Despliegue Tridimensional en Dispositivos Móviles de Datos Generados por Tomógrafos. Tekhné. 2017;1(19):29-46.

Conflicto de interés

Los autores declaran que no tienen conflicto de intereses

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

Contribuciones de los autores

Omar Mar Cornelio y Bárbara Bron Fonseca: Aportaciones importantes a la conceptualización de la investigación y diseño del estudio, a la recogida de datos, al análisis e interpretación de datos, la redacción del borrador del artículo, la revisión crítica de su contenido intelectual sustancial y la aprobación final de la versión a publicar.

Jorge Gulín González: Aportaciones importantes a la idea y diseño del algoritmo computacional, la redacción del borrador del artículo y la aprobación final de la versión a publicar.

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).