

DESARROLLO DE SOFTWARE DE SIMULACIÓN PARA GENERAR UN DOCUMENTO CLÍNICO XML - CDA

DEVELOPMENT OF SIMULATION SOFTWARE TO MAP A CLINICAL DOCUMENT TO XML – CDA

Autor 1 – Luis Fernando Gómez Ortega
Ing MSc Universidad Nacional Abierta y a Distancia
ORCID <https://orcid.org/0000-0002-5839-8141>
email. Luis.gomez@unad.edu.co

Autor 2 – Leonardo Yunda Perlaza
Ing MSc Ph.D Universidad Nacional Abierta y a Distancia
ORCID <https://orcid.org/0000-0003-0166-9999>
email. Leonardo.yunda@unad.edu.co

Autor 3 – Myriam Leonor Torres Pérez
Ph.D Universidad Nacional Abierta y a Distancia
ORCID <https://orcid.org/0000-0003-2923-5754>
email. Myriam.torres@unad.edu.co

Autor 4 – Claudio Camilo González Clavijo
MSc Universidad Nacional Abierta y a Distancia
ORCID <https://orcid.org/0000-0003-1344-3858>
email. Myriam.torres@unad.edu.co

RESUMEN.

El presente documento de trabajo se centra en el proceso de desarrollo de un simulador que permita llenar los campos de un documento denominado epicrisis y generar un documento XML-CDA y PDF, en el marco del proyecto denominado “Diseño y construcción de un documento HL7 CDA para historia clínica colombiana”.

En el proyecto “Diseño y construcción de un documento HL7 CDA para historia clínica colombiana”, se plantea el desarrollo de un simulador basado en software para ser usado en la formación de estudiantes de la

escuela de ciencias de la salud de la Universidad Nacional Abierta y a Distancia.

El propósito del software es poder brindar a los estudiantes una experiencia mucho más completa que la actual, a través del diligenciamiento del documento epicrisis que incluye aspectos legislativos, conjuntos estandarizados de datos y el uso de estándares como Loinc y HL7.

En adelante se procede a explicar el proceso de desarrollo del simulador y su respectiva justificación de acuerdo con las necesidades del proyecto.

Palabras Clave.

HL7; CDA; Epicrisis; Software; Simulación

ABSTRACT.

This working document focuses on the process of developing a simulator that allows filling in the fields of a document called epicrisis and generating an XML-CDA and PDF document, within the framework of the project called "Design and construction of an HL7 CDA document for Colombian medical history".

In the project "Design and construction of an HL7 CDA document for Colombian medical history", the development of a software-based simulator is proposed to be used in the training of students of the school of health sciences of the Open National University. Distance.

The purpose of the software is to provide students with a much more complete experience than the current one, through the completion of the epicrisis document that includes legislative aspects, standardized data sets and the use of standards such as Loinc and HL7.

From now on, the simulator development process and its respective justification are explained in accordance with the needs of the project.

Keywords.

HL7; CDA; Epicrisis; Software; Simulation

INTRODUCCIÓN.

El presente documento se enfoca en el proceso de desarrollo de un simulador software, para el desarrollo se tuvieron presentes los requerimientos y diseños realizados en etapas anteriores del proceso modelo de ciclo de vida clásico (1).

Para la programación se utilizó el paradigma programación orientada a objetos en el lenguaje java, utilizando el entorno de desarrollo integrado IDE Netbeans, para generar el reporte XML se utilizó la librería DOM (2) y para generar el PDF la librería Apache PDF Box (3).

Como motor de bases de datos se utilizó SQLite que resulto ser la mejor alternativa dados los requerimientos de tamaño que se tenían para el proyecto. A continuación, se describen con mayor profundidad las actividades realizadas en la etapa de desarrollo del proyecto (4).

METODOLOGÍA.

Para el desarrollo del proyecto en general se seleccionó el modelo de ciclo de vida clásico, que se divide en las fases de análisis, diseño, desarrollo, implementación y pruebas. Como se explica en (1).

En la fase de desarrollo se estableció que primero se debería codificar la base de datos y después las clases de mapeo y por último las clases no relacionadas directamente con la base de datos como se muestra en la figura 1.

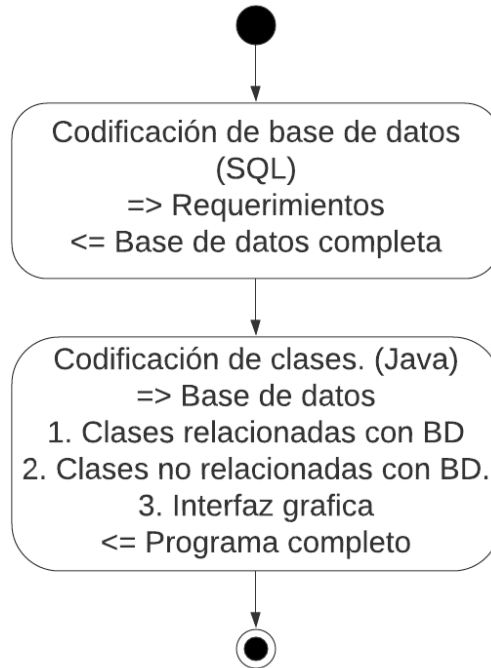


Figura 1. Actividades correspondientes a la fase de desarrollo. Fuente. Los autores

Se seleccionó SQLite para que fuera el motor de bases de datos porque es extremadamente liviano, característica indispensable para el proyecto dado que el software debía conservar el menor tamaño posible. Pues en la caracterización de los usuarios se encontró que muchos no contaban con internet veloz que les permitiera descargar archivos pesados rápidamente.

Por otra parte, al ser una librería que se incluye dentro del programa no se requiere la instalación de software adicional lo que resulta apropiado pues el 100% de los usuarios proyectados hasta ahora son estudiantes y profesores de ciencias de la salud que pueden verse afectados por actividades que son técnicas.

Como lenguaje de programación se encontró en Java la alternativa perfecta pues es un lenguaje multiplataforma y presenta la gran ventaja de tener una migración fácil a plataformas móviles como Android que hoy por hoy es el ecosistema más usado en el mundo por teléfonos celulares y tabletas con soporte en una amplia variedad de fabricantes (5).

El entorno de desarrollo usado fue Netbeans por su potencia y además por la preferencia del grupo investigador pues algunos de sus miembros ya habían trabajado con dicho IDE, con excelentes resultados.

En la figura 2. Se muestra la interfaz grafica del proyecto donde se tiene una botonera al lado izquierdo que es el lugar desde donde se pueden visualizar las diferentes funcionalidades, en el área de color gris se muestran las diferentes alternativas seleccionadas.

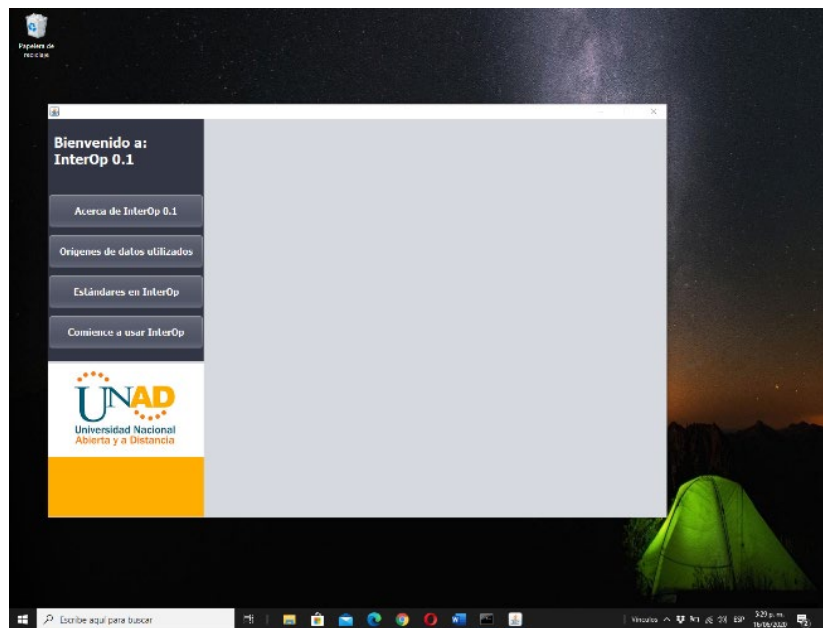


Figura 2. Software funcional en el sistema operativo Windows 10

DISCUSIÓN Y RESULTADOS.

Como resultado se tuvo un prototipo funcional del simulador que permite rellenar los datos del documento denominado epicrisis, además generar los archivos XML – CDA y PDF. De manera correcta.

Además, en las pruebas se demostró que el programa corre correctamente en Windows, Linux y Mac. Se espera en el futuro poder hacer el paso hacia Android.

CONCLUSIONES.

En conclusión, se pudo tener un prototipo funcional que puede ser implementado para la formación en diferentes cursos académicos como sistemas de información y calidad en salud, telesalud e interoperabilidad de los sistemas de información en salud. Todos de la escuela de ciencias de la salud de la UNAD impartidos en varios programas académicos.

BIBLIOGRAFÍA.

1. Pressman RS. Ingeniería del software: un enfoque práctico [Internet]. 7.ª ed. México, D. F.: McGRAW-HILL; 2013 [citado 17 de julio de 2020]. 777 p. Disponible en: http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272
2. Reading XML Data into a DOM (The Java™ Tutorials > Java API for XML Processing (JAXP) > Document Object Model) [Internet]. [citado 17 de julio de 2020]. Disponible en: <https://docs.oracle.com/javase/tutorial/jaxp/dom/readingXML.html>
3. Apache PDFBox | A Java PDF Library [Internet]. [citado 17 de julio de 2020]. Disponible en: <https://pdfbox.apache.org/>
4. SQLite Home Page [Internet]. [citado 16 de julio de 2020]. Disponible en: <https://www.sqlite.org/index.html>
5. Java | Oracle [Internet]. [citado 17 de julio de 2020]. Disponible en: <https://www.java.com/es/>