

## **DISEÑO DE SOFTWARE DE SIMULACIÓN PARA GENERAR UN DOCUMENTO CLÍNICO XML - CDA**

### **SIMULATION SOFTWARE DESIGN TO GENERATE A CLINICAL XML - CDA DOCUMENT**

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](mailto: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](mailto: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](mailto: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](mailto:Myriam.torres@unad.edu.co)

### **RESUMEN.**

El presente documento de trabajo se centra en el proceso de diseño de un simulador que permita llenar 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 mapeo de un documento clínico al estándar HL7 CDA, sin embargo, el equipo de investigadores decidió

llevar el proyecto mas allá de lo esperado y para la generación del CDA se propuso la creación de un simulador software, para que los estudiantes puedan interactuar con el y tener una experiencia mucho más completa que solo recibir la explicación del mapeo.

En el diseño del software se tuvieron en cuenta los requerimientos encontrados haciendo el cruce de las temáticas abordadas por cursos académicos afines y las áreas de cobertura del simulador. Para realizar el diseño se utilizaron diagramas del lenguaje de modelado unificado UML además del modelo relacional de bases de datos.

En adelante se procede a explicar el proceso de diseño 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 design process of a simulator that allows filling in 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 health records".

In the project "Design and construction of an HL7 CDA document for Colombian medical records", the mapping of a clinical document to the HL7 CDA standard is proposed, however, the team of researchers decided to take the project beyond what was expected and for the The generation of the CDA proposed the creation of a software simulator, so that students can interact with it and have a much more complete experience than just receiving the mapping explanation.

In the software design, the requirements found were taken into account by crossing the topics addressed by related academic courses and the

simulator coverage areas. To carry out the design, diagrams of the unified modeling language UML were used in addition to the relational database model.

From now on, the simulator design 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 diseño de un simulador software que permita a lo usuarios editar el documento denominado epicrisis con el fin de realizar una practica que le permita identificar los tipos de dato, conjuntos de datos, estándares y legislación que se encuentra detrás de la creación de un documento clínico. Además de identificar la importancia de los mecanismos de interoperabilidad en salud.

La etapa de diseño pate de los requerimientos expresados en la etapa de análisis donde se encontraron aspectos clave para la implementación de cara al sistema y al usuario, dichos requerimientos fueron llevados a diseños usando esencialmente el lenguaje de modelado unificado UML y el modelo relacional de bases de.

En conclusión, se logró establecer un modelo que puede cumplir con los requerimientos tanto de cara al software como a los usuarios, en adelante se profundiza en cada una de las actividades.

## **METODOLOGÍA.**

Para el desarrollo del proyecto en general se seleccionó como metodología el modelo de ciclo de vida clásico de la ingeniería del software, que se

divide en las fases de análisis, diseño, desarrollo, implementación y pruebas. Como se explica en (1).

El presente documento de trabajo se enfoca en la segunda etapa de dicho modelo denominada diseño para la cual se desarrollaron las tareas correspondientes a desarrollo de diagramas de casos de uso, diagrama de clases y diagrama de la base de datos, además de establecer decisiones para la etapa de implementación posterior a la presente.

En la figura 1, se muestran las tareas desarrolladas, en principio para los casos de uso en único insumo son los requerimientos, para el diseño de la base de datos se requieren los casos de uso y requerimientos, teniendo como resultado el diseño de esta, por último, para el diagrama de clases se requieren los requerimientos, casos de uso y base de datos. Dando como resultado el diagrama de clases que a después será implementado.

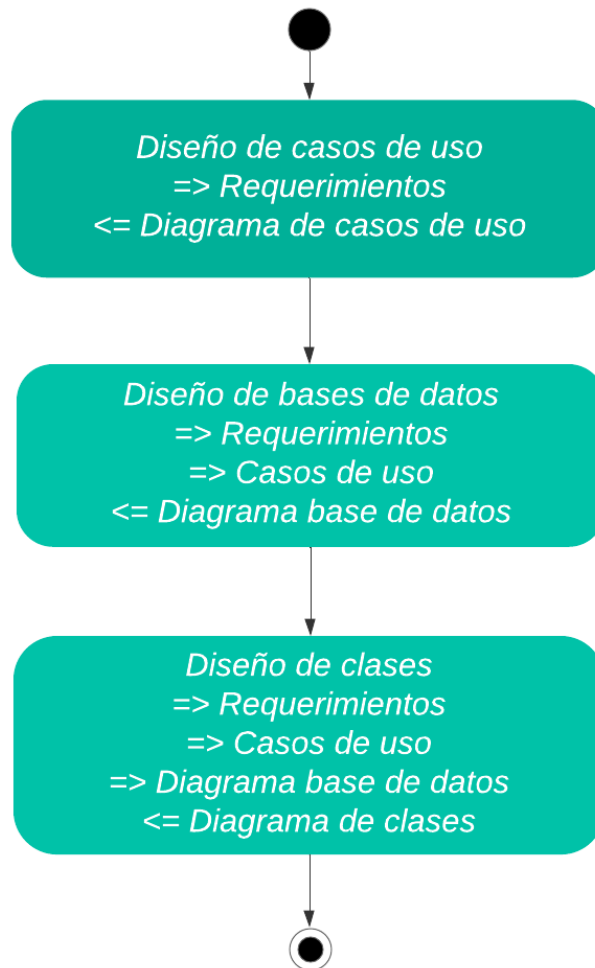


Figura 1. Actividades correspondientes a la fase de diseño. Fuente. Los autores

Para las actividades de diseño se utilizó el modelo de diseño unificado UML, que es una herramienta estandarizada para proyectos de software(2), que incluye las especificaciones para los diagramas de casos de uso y diagrama de clases que son parte del presente documento.

### Diseño de casos de uso.

El sistema únicamente cuenta con tres casos de uso, como se muestra en la figura 2. El usuario en el sistema después de registrar los datos puede generar un documento XML CDA o generar un documento PDF, es posible también que ingrese los datos y no genere ningún documento como se indica en el diagrama de casos de uso.

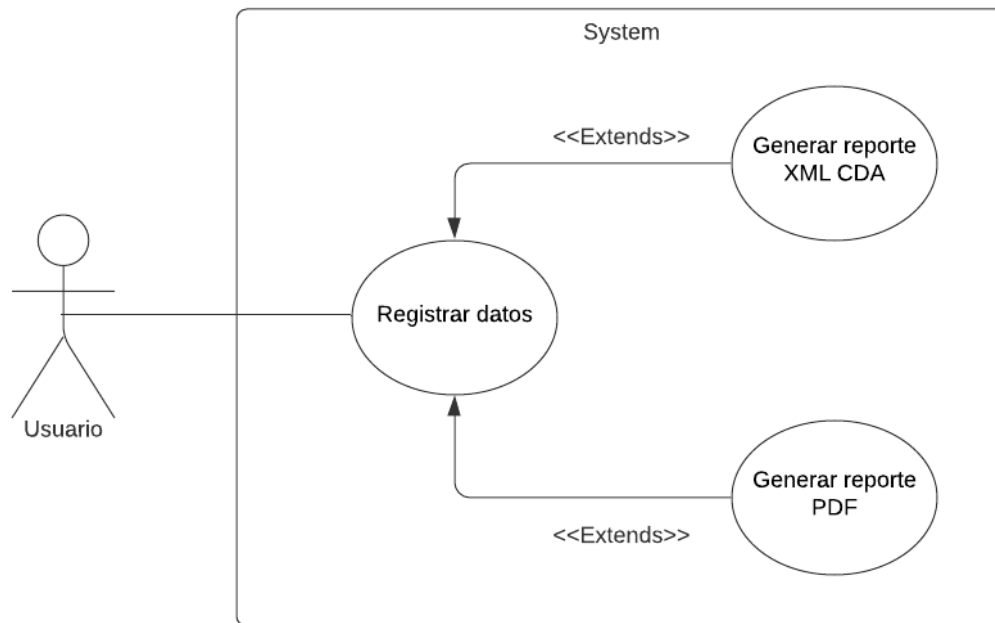


Figura 2. Diagrama de casos de uso del proyecto. Fuente. Los autores

## Diseño de la base de datos.

La base de datos del proyecto no almacena los registros ingresados con por el usuario según los requerimientos, la única función del simulador es proporcionar un entorno para que los estudiantes (usuarios) generen los reportes XML-CDA y PDF.

Dado lo anterior la base de datos del proyecto solo se usa para almacenar los datos, de los diferentes conjuntos que deben estar presentes en el documento epicrisis (3) que se muestran en la tabla 1.

**Tabla 1.** Conjuntos de datos y numero de registros.

Conjunto de datos	Número de registros
CIE 10	12.550
CUPS sin títulos	10.005
DIVIPOLA Municipios	1.121
CIUO 08	672
EPS	49
DIVIPOLA Departamentos	33
Documento	8
Régimen	5
Tipo usuario	3
<b>Total</b>	<b>24.446</b>

La base de datos del proyecto se desarrolló sin relaciones pues no se requieren para el simulador.

## Diseño de clases.

El diagrama de clases representa las entidades de la base de datos, para poder conectarlas se utilizó el patrón de diseño DAO, que permite tener clases de mapeo que se ven en color amarillo en la figura 1. Las clases

donde se generan los reportes se ven en un color azul más oscuro y en azul claro en las clases que representan las entidades de la base de datos.

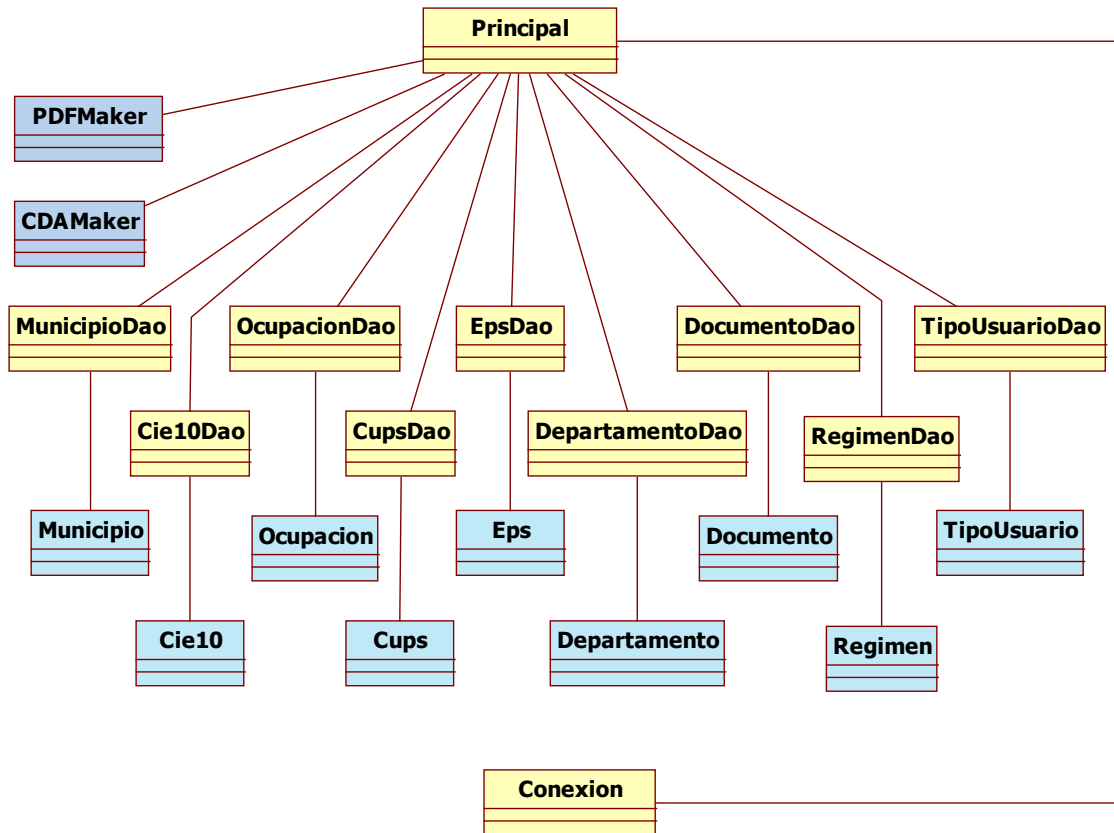


Figura 3. Diagrama de clases. Fuente: Los autores.

## DISCUSIÓN Y RESULTADOS.

Como resultado se tuvo un diseño que cumple con los requerimientos obtenidos en la etapa de análisis, los cuales por las particularidades del proyecto eran problemáticos a pesar de la simplicidad del software por aspectos como la necesidad de una implementación liviana que pueda correr en casi cualquier equipo de cómputo.

## **CONCLUSIONES.**

En conclusión, se obtiene un diseño que cumple con el propósito fundamental del proyecto teniendo en cuenta los requerimientos establecidos para el proyecto.

## **BIBLIOGRAFÍA.**

1. Pressman RS. Ingeniería del software: un enfoque práctico [Internet]. 7.<sup>a</sup> ed. México, D. F.: MCGRAW-HILL; 2013 [citado 5 de mayo de 2020]. 777 p. Disponible en: [http://www.ingebook.com/ib/NPcd/IB\\_BooksVis?cod\\_primaria=1000187&codigo\\_libro=4272](http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272)
2. Unified Modeling Language, v2.5.1. Unified Model Lang. :796.
3. Resolucion-3374-de-2000.pdf [Internet]. [citado 5 de mayo de 2020]. Disponible en: <https://consultorsalud.com/wp-content/uploads/2020/10/Modificacion-de-la-resolucion-3374-de-2000.pdf>