

# **El diseño del ensayo clínico controlado desde el sistema XAVIA SIDEC**

## ***The design of the clinical trial controlled from the XAVIA SIDEC system***

### **Resumen:**

El Centro de Inmunología Molecular realiza ensayos clínicos en diferentes pacientes con el objetivo de obtener nuevos medicamentos para el tratamiento de distintas enfermedades. Como medio de apoyo a la gestión de ensayos, la institución utiliza desde el 2012 el sistema XAVIA Clínicas, desarrollado por la Universidad de las Ciencias Informáticas, el cual consta de cuatro módulos, entre ellos Gestionar Estudios. Este módulo permite la creación de estudios, la confección del cronograma general, definir reglas para la validación de los campos de las hojas del Cuaderno de Recogida de Datos, sin embargo, presenta un conjunto de limitaciones haciendo que el trabajo de los especialistas resulte complejo. En el sistema Clínicas la gestión de grupos de sujetos se realiza en estudios diferentes, es decir, el sistema sólo gestiona por estudio un cronograma y por ende un sólo grupo de sujetos. Esta situación provoca que el análisis de un ensayo clínico sea difícil para los especialistas, debido a que la búsqueda de sus resultados se realiza de forma engorrosa. Una vez que se inicia la conducción de los ensayos, no es posible agregar nuevas reglas o modificar las existentes. En caso de realizar cambios, el especialista debe introducir los datos nuevamente, haciendo el proceso costoso en tiempo, tanto para el ensayo como para el especialista. Como parte del Sistema para el Manejo de Datos de Ensayos Clínicos (XAVIA SIDEC) se desarrolla el módulo Diseño, para solventar las insuficiencias detectadas. Los materiales y métodos aplicados fueron como metodología de desarrollo la variación del Proceso Unificado Ágil para la UCI, como gestor de base de datos PostgreSQL, como servidor de aplicación JBoss

Server, como herramienta de desarrollo Eclipse, para la creación de los artefactos Visual Paradigm y como lenguaje de programación Java. El desarrollo de la solución tiene varios beneficios asociados, entre ellos: la obtención de resultados correspondientes a sujetos con características similares, permite un mejor análisis del estudio realizado. La flexibilidad al permitir modificar y agregar reglas después de iniciada la conducción del ensayo, reduce el tiempo de desarrollo y evita el atraso de los mismos. Además, proporciona un acceso inmediato y actualizado del estado de los estudios. Genera un ahorro significativo en capital monetario, garantizando la adquisición de un software de desarrollo propio sin incurrir en gastos por concepto de licencias. Contribuye a la política de soberanía tecnológica llevada a cabo a partir de la implementación de los lineamientos del partido.

**Palabras clave: Ensayos Clínicos; Gestión de Ensayos Clínicos; Sistema para el Manejo de Datos Clínicos.**

**Abstract:**

*Molecular Immunology Center carries out clinical tests in different patients with the aim of getting new drugs to treat various diseases. As means of supporting tests management, the institution uses from 2012 the XAVIA Clinics system, developed by the University of Information Science, which consists of four modules, including Managing Studies module. This module allows the creation of studies, the making of general schedule, the definition of rules for validating fields of the Data Collection Notebook sheets, however, it presents a set of limitations making that the work of specialists resulting complex. In Clinics system, the management groups of subjects is performed in different studies, in example, the system only manages a schedule for study and thus one group of subjects. This situation causes the analysis of a clinical trial is difficult for specialists, because the searching of results is performed cumbersome. After conducting tests start, it's not possible to add new rules or modify the existing ones. If changes are made changes, the specialist must enter the data again, making the process costly for both the test and the specialist. As part of the System for Data Management of*

*Clinical Tests (XAVIA SIDEC) the Design module is developed to resolve the weaknesses identified. The materials and methods applied: as methodology the Agile Unified Process, as database manager PostgreSQL, and JBoss server as application server, Eclipse was used as development tool and Java like programming language. All artifacts were created with Visual Paradigm. The development of the solution has several associated benefits, including: obtaining the corresponding results of subjects with similar features, allow a better analysis of the performed study. It is flexible because it allows to modify and add rules after the conducting tests have started, reduces development time and avoids the delay of the same. It also provides an immediate and updated access of the studies's status. It generates significant savings in monetary capital, ensuring the acquisition of proprietary software without incurring expenses for licenses. It contributes to the policy of technological sovereignty carried out after the implementation of the organization lines.*

**Keywords: Administration of clinical trials; Clinical trials; System Data Management Clinical Trials.**

## Introducción

El Centro de Inmunología Molecular (CIM), institución biotecnológica cubana dedicada a la investigación básica, desarrollo y fabricación de productos a partir del cultivo de células de mamíferos. Su principal área de investigación y desarrollo es la búsqueda de nuevos productos para el diagnóstico y tratamiento del cáncer y enfermedades relacionadas con el sistema inmune (CIMAB, 2016). Para valorar la calidad y seguridad de estos nuevos productos en los seres humanos, el centro realiza ensayos clínicos en hospitales altamente especializados.

Como medio de apoyo a la gestión de ensayos, la institución utiliza desde el 2012 el sistema XAVIA Clínicas, desarrollado por estudiantes y profesores del Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas (UCI). El mismo cuenta con los módulos Administrar Sistema, Enviar Datos, Extraer Datos y Gestionar Estudio. Este último permite el diseño de los ensayos clínicos, facilitando entre otros: la creación de estudios, los cuales se realizan sobre pacientes sometidos a investigación clínica, el diseño de los Cuadernos de Recogida de Datos (en lo adelante hojas CRD) donde se recogen los datos de los pacientes, y la confección, en un cronograma general, de los momentos de seguimiento programados y no programados, que describen un tipo de seguimiento que sucede durante la realización del estudio y en los cuales se llenan las hojas CRD. Permite, además, definir reglas para la validación de los campos de las hojas CRD, sin embargo, presenta un conjunto de insuficiencias haciendo que el trabajo de los especialistas resulte complejo.

Los sujetos incluidos en un estudio se dividen en diferentes grupos siguiendo sus características y para cada grupo el proceso de recogida de datos debe seguir un cronograma diferente. Actualmente en el sistema Clínicas, para gestionar varios grupos de sujetos se tienen que hacer en estudios diferentes, es decir, el sistema solo gestiona por estudio un cronograma y por ende un solo grupo de sujetos.

Esto provoca que el análisis de un ensayo clínico sea difícil para los especialistas, debido a que la búsqueda de sus resultados se realiza de forma engorrosa.

Otro inconveniente que presenta, es que una vez que el diseño del estudio es aprobado por el investigador promotor y se inicia la conducción de los ensayos, no es posible agregar nuevas reglas o modificar las existentes. En caso de realizar cambios, el especialista debe rediseñar el estudio e introducir los datos nuevamente trayendo como consecuencia pérdidas de tiempo y atraso del ensayo clínico.

La presente investigación se traza como objetivo el desarrollo del módulo Diseño del Sistema para el Manejo de Datos de Ensayos Clínicos (XAVIA SIDEC), solventando las insuficiencias detectadas en el módulo Gestionar Estudios del sistema Clínicas.

## **Metodología**

Para el desarrollo del módulo Diseño del sistema XAVIA SIDEC primeramente el equipo de proyecto se reunió con el cliente para definir los requisitos. Los métodos empíricos que se utilizaron fue-ron la Observación y la Entrevista, para entender el proceso de gestión de los ensayos clínicos en el CIM. Además, se utilizaron como métodos teóricos el Análisis - Síntesis e Histórico - Lógico, para el análisis de la literatura en el estudio de la metodología de desarrollo, de las tecnologías y herramientas que se pudieran utilizar para el desarrollo del módulo, y se investigó sobre la historia y antecedentes relacionados con el manejo de los ensayos clínicos en Cuba.

### **Metodología AUP-UCI**

Proceso Unificado Ágil (AUP, por sus siglas en inglés) de Scott Ambler es una versión simplificada del Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés). Esta describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que se mantienen válidos en RUP.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se utilizó como metodología de desarrollo la variación del Proceso Unificado Ágil en unión con el Modelo CMMI -DEV v1.3 definida para la actividad productiva de la UCI.

Descripción de las fases:

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, se unifican las restantes tres fases de AUP en una sola, a la que se llamará Ejecución y se agrega la fase de Cierre.

- Inicio: se llevan a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- Ejecución: se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- Cierre: se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Las disciplinas definidas en la variación AUP para la UCI son:

- Modelado de negocio: destinada a comprender los procesos de negocio de una organización. Para modelar el negocio se proponen las siguientes variantes: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC).

- **Requisitos:** comprende la administración y gestión de los requisitos funcionales y no funcionales del producto, agrupados en cuatro escenarios condicionados por el Modelado de negocio. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)].
- **Análisis y diseño:** se modela el sistema y su forma (incluida su arquitectura) para que soporte los requisitos, incluyendo los requisitos no funcionales.
- **Implementación:** a partir de los resultados del Análisis y Diseño se construye el sistema.
- **Pruebas internas:** se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.
- **Pruebas de liberación:** pruebas diseñadas y ejecutadas por una entidad certificadora externa, de la calidad a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
- **Pruebas de Aceptación:** es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Todas las disciplinas antes definidas se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales.

Escenario para la disciplina Requisitos:

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos de la UCI, quedando de la siguiente forma:

- Escenario No. 1

Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

$$\text{CUN} + \text{MC} = \text{CUS}$$

- Escenario No. 2

Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$$\text{MC} = \text{CUS}$$

- Escenario No. 3

Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

$$\text{DPN} + \text{MC} = \text{DRP}$$

- Escenario No. 4

Proyectos que no modelen negocio solo pueden modelar el sistema con HU (UCI, 2015).

HU

Por las características del sistema desarrollado se definió que el escenario a utilizar es el No. 2, proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

### **Arquitectura**

Para el desarrollo del sistema se utilizó la Plataforma Java Enterprise Edition 5.0 utilizando como ser-vidor de aplicaciones JBoss Server 4.2.2. Este servidor de aplicaciones se encuentra certificado con J2EE (Java 2 Enterprise Edition) y soporta sistemas de gran complejidad y alta concurrencia. Al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte e implementa todo el paquete de servicios de J2EE (Barrios, 2003). Como Sistema

Gestor de Base de Datos se utilizó PostgreSQL 8.4 y el marco de trabajo Jboss Seam 2.1.1 para la lógica del negocio. El sistema está basado sobre el patrón arquitectónico Modelo-Vista-Controlador.

Patrón arquitectónico:

Para el diseño arquitectónico, el sistema fue implementado según el patrón Modelo-Vista-Controlador (MVC), que se adapta muy bien a las necesidades del negocio y es aprobado internacionalmente para el desarrollo de aplicaciones Web.

La arquitectura Modelo-Vista-Controlador fue introducida como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk. Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vis-tas (Catalani, 2007).

- Modelos: es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas.
- Vistas: las vistas, como su nombre nos hacen entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación. En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos.
- Controladores: contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una búsqueda de información. En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no

es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo (Álvarez, 2014).

### **Herramientas para el desarrollo del sistema**

Para el desarrollo del sistema se utilizó como herramienta de modelado Visual Paradigm la cual proporciona un potente modelado mediante el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), como lenguaje de programación Java el cual es un lenguaje orientado a objetos, multihilo y dinámico; como entorno de desarrollo integrado Eclipse, es de código abierto, basado en Java que facilita la escritura de código correcto, disminuye el tiempo de depuración e incrementa la productividad del desarrollador.

### **Resultados y discusión**

Con el objetivo de detectar las limitaciones del módulo Gestionar Estudios del sistema XAVIA Clínicas se realizó un análisis del mismo tanto por el proyecto Ensayos Clínicos de la UCI, así como por parte de los especialistas del Centro de Inmunología Molecular vinculados al manejo de ensayos clínicos. Concluido el estudio, se determinó que una parte de las funcionalidades del módulo Gestionar Estudios satisfacen el proceso de diseño de los ensayos, sin embargo, se identificaron otras funcionalidades de las cuales el módulo carece.

Con el desarrollo del módulo Diseño del Sistema para el Manejo de Datos de Ensayos Clínicos se obtiene un nuevo sistema que supera las insuficiencias existentes en su versión precedente. El módulo Diseño está formado por seis submódulos:

- Gestionar Estudios
- Gestionar Grupo de Sujetos
- Gestionar Cronograma General
- Gestionar Momentos de Seguimientos (MS)

- Gestionar Hojas CRD
- Gestionar Reglas

Primeramente, el investigador promotor diseña el estudio, teniendo en cuenta la Guía para la Elaboración del Protocolo de Ensayo Clínico, que contiene los datos generales del ensayo. Una vez realizado esto, el gerente de datos puede crear el estudio, así como gestionar los grupos de sujetos que participarán en el ensayo. Un grupo de sujeto es la agrupación de varios pacientes según características específicas de cada uno, como pueden ser la edad, sexo o patología. El gerente de datos crea además el cronograma general de ejecución del estudio para cada grupo de sujetos. Luego puede gestionar los Momentos de Seguimiento (MS) para cada una de las etapas del cronograma (Evaluación, Tratamiento o Seguimiento) y crear las hojas CRD asociadas a estos MS.

Un MS es un tipo de seguimiento (programado o no programado) que sucede durante la realización del estudio, en el cuál se realizan exámenes a los sujetos. En la creación de un MS programado se planifican los momentos por visita (día de visita, que el paciente debe asistir a la consulta) o por período (cada varios días durante un período de tiempo en semanas, meses o años). En el caso de los MS no programados, se planifican una vez cuando existan los síntomas o cada vez que haya síntomas. En ambos casos, se procede a seleccionar hojas CRD para los momentos.

Las hojas CRD recogen toda la información clínica de los sujetos del ensayo y son cargadas al sistema como un documento EXCEL que cumpla con el formato requerido. La plantilla Excel está constituida por 4 hojas o páginas y todos los términos aparecen en inglés. Las mismas son:

- CRF: donde se define el nombre del CRD, su descripción y el número de la versión actual.
- Sections: es donde se definen los submodelos dentro de la hoja de CRD que estamos diseñando que puede ser análoga a un modelo.

- **Groups:** es una manera de agrupar las variables del modelo; es útil para la creación de tablas.
- **Items:** son las variables que se definen para cada una de las secciones creadas anteriormente.

Posteriormente debe gestionar las reglas de validación y derivación para las variables de las hojas CRD para las visitas definidas en el cronograma de ejecución. Una vez aprobado el cronograma y las reglas de validación y derivación, el Investigador Promotor aprobará el diseño del estudio y se dará paso a la gestión de los datos para los sujetos.

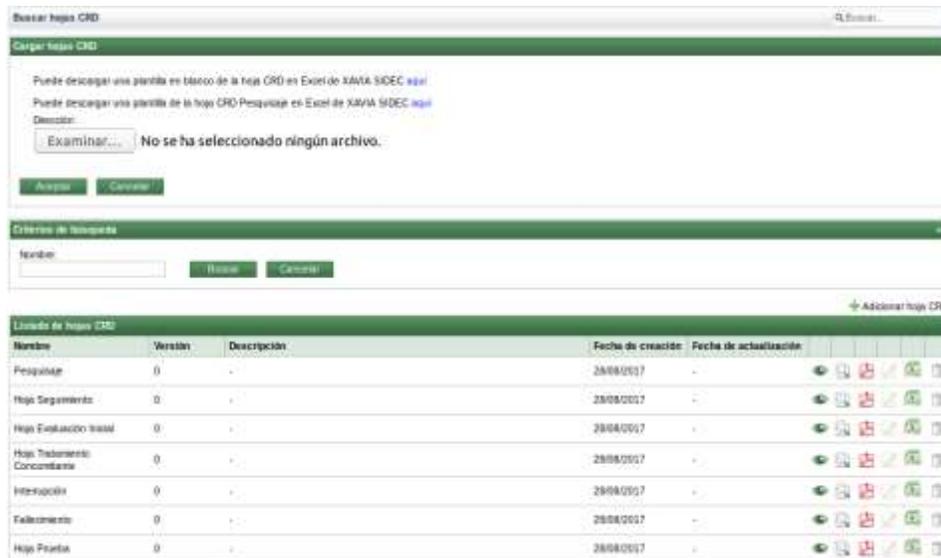


Figura 1. Interfaz que permite gestionar las hojas CRD en el módulo Diseño del SIDE.

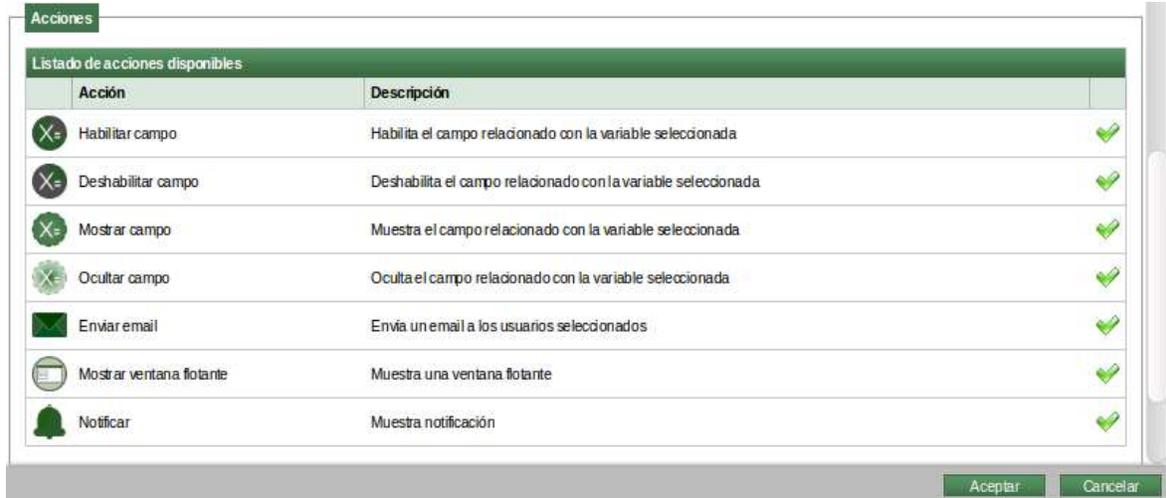


Figura 2. Interfaz que permite seleccionar las acciones que se ejecutarán si se cumple la regla.

El desarrollo de la solución tiene varios beneficios asociados, entre ellos: la obtención de resultados correspondientes a sujetos con características similares, permite un mejor análisis del estudio realizado. La flexibilidad al permitir modificar y agregar reglas después de iniciada la conducción del ensayo, reduce el tiempo de desarrollo y evita el atraso de los mismos. Además, proporciona un acceso inmediato y actualizado del estado de los estudios. Agiliza el proceso de diseño de los Cuadernos de Recogida de Datos.

Genera ahorros significativos en capital monetario, garantizando la adquisición de un software de desarrollo propio sin incurrir en gastos por concepto de licencias. Contribuye a la política de soberanía tecno-lógica llevada a cabo a partir de la implementación de los lineamientos del partido.

## Conclusiones

Se describieron las principales características del módulo Diseño del sistema XAVIA SIDEC, el cual es capaz de resolver las insuficiencias presentes en el módulo Gestionar Estudio del sistema XAVIA Clínicas.

El sistema permite diseñar el ensayo clínico por grupos de sujetos, lo cual posibilita un mejor análisis del estudio realizado. Además, estandariza la información manejada en los ensayos clínicos y agiliza el proceso de diseño de los

Cuadernos de Recogida de Datos. El sistema se integra a las políticas de informatización de la salud pública cubana, reflejada en los lineamientos del partido.

## **Bibliografía**

1. CIMAB. Centro de Inmunología Molecular. [En línea] [Citado el: 1 de julio de 2016.] <http://www.cimab-sa.com/index.php?action=cim>
2. PROGRAMA DE MEJORA. “Metodología de desarrollo para la Actividad productiva de la UCI”, Universidad de las Ciencias Informáticas, La Habana, Cuba, 2015.
3. Barrios Núñez., Juan Manuel, Investigación de la Plataforma J2EE y su aplicación práctica, [En línea] [Citado el: 12 de julio de 2016.] "<http://users.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf>," Universidad de Chile, 2003.
4. Catalani, Exequiel, Desarrollo, [En línea] <https://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>, 2007.
5. Álvarez., Miguel Ángel, Desarrollo Web, [En línea] <http://www.desarrolloweb.com/articulos/que-es-mvc.html>, 2014.