

**Universidad de las Ciencias Informáticas
Facultad 7**



**Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas**

*Componente Tabular para el Subsistema Editor de Plantillas del
Sistema para la Gestión y Análisis de Información Estadística en la
Salud Pública Cubana*

Autores: Annis Martínez Dominguez
Daniel Fernández Leyva

Tutor: Ing. Karel Fernández Cedeño

Ciudad de La Habana, Junio 2010

“Año 52 de la Revolución”

Declaración de autoría:

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 23 días del mes de junio del año 2010.

Annis Martínez Domínguez

Firma de la Autora

Daniel Fernández Leyva

Firma del Autor

Ing. Karel Fernández Cedeño

Firma del Tutor

Datos de contacto:

Ing. Karel Fernández Cedeño graduado en la Universidad de las Ciencias Informáticas en el año 2008. Ha impartido la Asignatura de Ingeniería de Software, Preparación para la Prueba de Nivel de ISW y Preparación para la Prueba de Nivel de Programación. Perteneció al Área Temática Sistemas de Apoyo a la Salud donde se desempeñaba como Líder del proyecto SIEC Salud y como analista principal del Área. Actualmente se desempeña como Analista Principal del Centro de Informática Médica.

Correo electrónico: kfernandez@uci.cu.

Resumen:

El presente trabajo de diploma se realiza con el objetivo de desarrollar un componente tabular que permita sustituir el utilizado actualmente por el Subsistema Editor de Plantillas (SEP) del Sistema para la Gestión y Análisis de Información Estadística en la Salud Pública Cubana. El mismo brindará las funcionalidades necesarias para editar los modelos de flujo utilizados en la captación de información estadística en el Ministerio de Salud Pública.

Para el desarrollo de la solución propuesta se utilizó la herramienta de modelado Enterprise Architect 7.1 basada en UML 2.0 (Lenguaje Unificado de Modelado). Como Entorno de Desarrollo Integrado se utilizó Visual Studio 2008 y el lenguaje de programación utilizado fue CSharp. El sistema está concebido bajo la metodología RUP. Igualmente se analizan las tendencias actuales en Cuba y el mundo de aplicaciones informáticas existentes que se encargan del procesamiento de información estadística. Se asimilan las metodologías y tecnologías propuestas por el Departamento Sistema de Apoyo a la Salud de la Universidad de las Ciencias Informáticas para el desarrollo de aplicaciones para la salud.

A partir del uso de este componente se facilita la edición de los modelos de flujo o plantillas en dependencia de las necesidades, esto proporciona autonomía a los usuarios para establecer sus definiciones. También se podrá brindar soporte efectivo al SEP para satisfacer las necesidades de los usuarios finales. Además, permitirá el despliegue del Sistema para la Gestión y Análisis de Información Estadística en la Salud Pública Cubana y eliminará las limitaciones de comercialización.

Palabras claves: componente tabular, estadística, modelos de flujo, edición

Índice:

Introducción	1
Capítulo 1 Fundamentación Teórica	5
1.1. Conceptos generales	5
1.1.1. Sistema Nacional de Salud (SNS)	5
1.1.2. Informatización del Sistema Nacional de Salud	6
1.2. Conceptos relacionados con el dominio del problema	7
1.2.1. Estadísticas de salud	7
1.2.2. Sistemas de Información Estadística (SIE)	8
1.2.3. Sistema de Información Estadística Complementaria (SIEC) de Salud	10
1.2.4. Modelo de flujo	11
1.3. Antecedentes del sistema	11
1.3.1. Antecedentes del sistema en el ámbito internacional	11
1.3.2. Antecedentes del sistema en el ámbito nacional	15
1.4. Análisis crítico	17
1.5. Tendencias tecnológicas y metodologías actuales	17
1.5.1. Modelo Vista Controlador (MVC)	18
1.5.2. Lenguajes	18
1.5.3. Metodologías	21
1.5.4. Herramientas	22
Capítulo 2 Características del sistema	25
2.1. Objetos de automatización	25
2.2. Información que se maneja	25
2.3. Descripción del Sistema	25
2.4. Modelo de dominio	26
2.4.1. Conceptos Fundamentales del dominio	27
2.5. Especificación de los requisitos de software	28
2.5.1. Requisitos funcionales	28
2.5.2. Requisitos no funcionales	28
2.6. Modelado del Sistema	31
2.6.1. Actores del sistema	31
2.6.2. Diagrama de casos de uso del sistema	31
2.6.3. Especificaciones de los casos de uso	32
Capítulo 3 Diseño del sistema	46

3.1. Estructura del diseño _____	46
3.2. Modelo de diseño _____	47
3.3. Diagramas de clases del diseño _____	47
3.3.1. CUS Insertar columna _____	47
3.3.2. CUS Eliminar columna _____	49
3.3.3. CUS Insertar fila _____	50
3.3.4. CUS Eliminar fila _____	51
3.3.5. CUS Establecer formato _____	52
3.3.6. CUS Combinar celdas _____	53
3.3.7. CUS Separar celda _____	54
3.3.8. CUS Bloquear celda _____	55
3.4. Diagramas de secuencia _____	56
3.4.1. CUS Insertar columna – Escenario Insertar columna a la derecha _____	56
3.4.2. CUS Eliminar columna _____	57
3.4.3. CUS Establecer formato – Escenario Establecer fuente _____	57
3.4.4. CUS Exportar _____	58
3.5. Descripción de las clases _____	58
Capítulo 4 Implementación _____	60
4.1. Modelo de implementación _____	60
4.2. Componentes _____	60
4.3. Diagrama de componentes _____	60
Conclusiones _____	62
Recomendaciones _____	63
Referencias Bibliográficas _____	64
Bibliografía _____	66
Anexos _____	68
Glosario de Términos _____	70

Introducción

El futuro de la humanidad dependerá en gran medida del potencial humano, de la gestión de la producción y de los conocimientos que se alcancen. La informática en sus diferentes manifestaciones tiene asegurado un papel protagónico en este futuro. Cuba se propone su utilización justa y racional sobre principios éticos sostenibles. En los últimos tiempos en el sector de la salud se han desarrollado sistemas informáticos para el mejoramiento de los procesos médicos asistenciales, que garanticen el incremento en la calidad y seguridad de la atención médica a la población. También ha sido imprescindible incursionar en el desarrollo de aplicaciones para la gestión de la información estadística de salud.

La estadística se remonta a los comienzos de la historia desde épocas muy remotas. Como todas las ciencias no surgió de improviso, sino mediante un proceso largo de desarrollo y evolución que va desde hechos de simple recolección hasta la diversidad y rigurosa interpretación del gran volumen de información que se maneja hoy en día. Por esto se presenta como rama auxiliar de las distintas ciencias y técnicas y se encargada de estudiar los procedimientos destinados a la recolección, resumen, análisis e interpretación de un conjunto de datos. Con el advenimiento de la computación y el desarrollo de grandes sistemas informáticos, además del crecimiento progresivo de enormes volúmenes de datos, se hace necesario que la información se presente en forma de cuadro o tablas, lo que ha causado una evolución considerable en la estadística. [1]

En Cuba adquiere vital importancia el manejo y control de la información estadística y para ello se cuenta con la Oficina Nacional de Estadística (ONE), Órgano Central del Estado encargado de dirigir, ejecutar, y controlar todas las estadísticas oficiales que se reportan. En el caso de los ministerios e instituciones del Estado poseen sus direcciones de estadísticas que se encargan de satisfacer las demandas de información numéricas en dependencia de las funcionalidades que desempeñan, cada uno está regido metodológicamente por la ONE.

Entre los ministerios con que cuenta el Estado se encuentra el Ministerio de Salud Pública (MINSAP), encargado de cubrir y dirigir la aplicación de la política del Estado y del Gobierno en cuanto a la salud

pública, de ahí que constituye el organismo rector del Sistema Nacional de Salud (SNS) dado por un conjunto de métodos, técnicas, procederes y actividades gerenciales dirigidas al manejo de la información en salud, por lo que posee uno de los Sistemas de Información Estadística (SIE) de mayor complejidad, extensión y robustez dentro de los Organismos de la Administración Central del Estado (OACE).

Entre los SIE, se encuentran los Sistemas de Información Estadística Complementaria (SIEC) que están destinados a satisfacer las necesidades informativas de los órganos, organismos e instituciones estatales dentro de ellos se encuentra el Sistema de Información Estadística Complementaria de la Salud (SIEC de Salud) encargado de elaborar, captar, procesar y emitir información estadística relacionada con las actividades realizadas en cada uno de los servicios que brinda la salud pública cubana.

Una de las proyecciones de desarrollo hacia el futuro que tiene el sector de la salud para el SIEC de Salud y en conjunto para cada uno de sus subsistemas, es lograr un desempeño pertinente y más racional, la incorporación del muestreo y encuestas periódicas a profundidad, así como la extensión de la tecnología informática y redes de computadoras. Se pretende perfeccionar los procesos de almacenamiento y recuperación en soporte electrónico así como el análisis de la información por diferentes vías, especialmente por la calificación de los recursos humanos, la mayor participación de profesionales de la estadística y la incorporación de procesos automatizados que contribuyan a optimizarlo.

Para ello se trabaja de manera conjunta entre los especialistas de la Dirección Nacional de Registros Médicos y Estadística Sanitaria y la Universidad de las Ciencias Informáticas (UCI), específicamente la Facultad 7 a la cual pertenece el proyecto SIEC de Salud, que se dedica como su nombre lo indica a la informatización de cada uno de los subsistemas estadísticos del MINSAP.

El proyecto SIEC de Salud desarrolló en el curso 2008 – 2009 el Subsistema Editor de Plantillas (SEP), que pertenece al Sistema para la Gestión y Análisis de Información Estadística en la Salud Pública Cubana, este subsistema posee la responsabilidad de editar los modelos de flujo estadístico, y en su desarrollo se utilizó un Componente Tabular (FlexCell Grid Control) que cubría un alto por ciento de las funcionalidades requeridas.

Sin embargo, existe la necesidad de ampliar las funcionalidades y garantizar un proceso de soporte efectivo para el subsistema. Actualmente esto sería imposible debido a que no se brinda el soporte por

parte del proveedor para hacer cambios en el Componente Tabular utilizado, por esta razón no podrían satisfacerse futuras necesidades del Subsistema Editor de Plantillas, lo que traería como consecuencia el impedimento de asegurar soporte al usuario final del subsistema. Tampoco sería posible migrar la aplicación a plataformas libres ya que en la implementación de este componente el propietario utilizó recursos de Windows que lo hacen incompatible con otros sistemas operativos.

Cuba también se ha visto imposibilitado de comprar las licencias de software en el mercado o en caso contrario sólo se pueden obtener a través de terceros, lo que supone un alto riesgo, sin dejar de mencionar que el país pretende sustituir todas las importaciones posibles por soluciones propias.

Dada la situación anterior se plantea como **problema a resolver**: Utilización en el Subsistema Editor de Plantillas de un componente que impide brindar soporte de manera eficiente al propio subsistema. El **Objeto de Estudio** se enmarca en el proceso de gestión de la información en el Sistema de Información Estadística Complementaria de Salud. El **Campo de Acción** se centra en el proceso de gestión de los modelos de flujo en el Sistema de Información Estadística Complementaria de Salud.

Para dar solución al problema propuesto se ha definido como **Objetivo**: Desarrollar un Componente Tabular para el Subsistema Editor de Plantillas que pueda sustituir el utilizado actualmente, con un mínimo de esfuerzo en la implementación del negocio propio del SEP.

Con motivo de darle cumplimiento al objetivo planteado se han trazado las siguientes **Tareas de la investigación**:

- Evaluar el estado del arte referente a componentes tabulares existentes para la identificación de las semejanzas y diferencias funcionales que presentan con respecto a la aplicación que se propone.
- Identificar las funcionalidades del FlexCell utilizadas por el SEP para que sean implementadas en el nuevo componente.
- Probar a lo largo del ciclo de vida del software la compatibilidad con el Framework Mono para una futura migración del SEP hacia plataformas libres.
- Documentar los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño e Implementación que facilita la continuidad en el desarrollo del Componente Tabular.

- Implementar un Componente Tabular que cumpla funcionalmente con las demandas del SEP para que sustituya al componente que actualmente utiliza este subsistema.

La presente investigación consta de cuatro capítulos, los cuales están estructurados de la manera siguiente:

En el **capítulo 1 Fundamentación Teórica** se muestran los conceptos fundamentales para el entendimiento del documento, además de conceptos básicos para comprender la gestión de la información de salud. Se expone el estado del arte del tema de la investigación, tanto nacional como internacional y se realiza un estudio sobre las principales tecnologías, metodologías de desarrollo, estilos de arquitectura y lenguajes de programación enfocándose finalmente en las tecnologías escogidas para el desarrollo de la solución propuesta.

En el **capítulo 2 Características del sistema** se realiza una breve descripción de los procesos del negocio relacionados con la gestión de la información estadística de la salud. Se especifica el dominio. Además, se especifican los requisitos funcionales y no funcionales a partir de los cuales se representan los casos de uso del sistema y la descripción de los mismos, y se obtiene un prototipo no funcional que permite validar que el sistema satisface lo que se pretende lograr.

En el **capítulo 3 Diseño del sistema** primeramente se realiza el diseño del sistema donde se define el diagrama de clases del diseño por casos de uso, así como la relación existente entre ellas. Posterior a esto se muestra la interacción entre los actores y el sistema mediante los diagramas de secuencia.

En el **capítulo 4 Implementación** se presenta una colección de componentes y otros subsistemas de implementación utilizados para estructurar el modelo de implementación y dividirlos en pequeñas partes que pueden ser integradas de formas separadas. Incluye modelos claves del subsistema como el diagrama de componentes.

Capítulo 1 Fundamentación Teórica

En el presente capítulo se muestra una breve panorámica de cómo se lleva a cabo el proceso de informatización en el Sistema Nacional de Salud Cubano, igualmente se abordan conceptos básicos y definiciones esenciales para un mayor entendimiento del proceso de gestión de la información estadística en el sector de la salud del país. Se realiza un análisis sobre componentes tabulares existentes en Cuba y en el resto del mundo. Además, se analizan las tecnologías, metodologías y herramientas en las que se apoya la solución al problema.

1.1. Conceptos generales

1.1.1. Sistema Nacional de Salud (SNS)

Con el triunfo de la Revolución Cubana en enero del 1959 surgieron importantes cambios que marcaron el inicio de un impetuoso desarrollo en todas las esferas de la vida económica, social y política del pueblo cubano. De ahí que a partir de los años 60 se comenzó a trabajar en la creación del Sistema Nacional de Salud con el objetivo de llevar la acción del trabajador de la salud a los lugares más apartados. Desde entonces el sistema creado comenzó a realizar importantes reformas en la salud pública del país, cuyo basamento se potencia en la colaboración internacional y la participación de la comunidad.

El Sistema de Salud en Cuba está compuesto por:

- Atención médica preventiva, curativa y de rehabilitación.
- Asistencia a ancianos, impedidos físicos, impedidos mentales y otros minusválidos.
- Control higiénico – epidemiológico.
- Formación, especialización y educación continuada de profesionales y técnicos.
- Investigación y desarrollo de las ciencias médicas.
- Estadísticas de salud.
- Información científica – técnica.
- Promoción de salud.

- Aseguramiento de suministros no médicos y mantenimiento del sistema.
- Aseguramiento tecnológico, aseguramiento médico y electromedicina.
- Producción, distribución y comercialización de medicamentos y equipos médicos.

El SNS cubano está estructurado en cuatro niveles de dirección, los cuales se encuentran identificados con la estructura Político – Administrativa del país. Estos niveles son los siguientes:

Nivel Nacional: Representado por el MINSAP como órgano rector con funciones metodológicas, normativas, de coordinación y de control en la aplicación de las políticas del Estado y el Gobierno en cuanto a la salud pública, el desarrollo de las Ciencias Médicas y la Industria Médico - Farmacéutica.

Nivel Provincial: Representado por las Direcciones Provinciales de Salud, directamente subordinadas administrativa y financieramente a la Asamblea Provincial del Poder Popular.

Nivel Municipal: Representado por las Direcciones Municipales de Salud, dependientes administrativa y financieramente de la Asamblea Municipal del Poder Popular.

Unidades de Atención Primaria a la Salud: Constituye el primer nivel de atención médica del Sistema Nacional de Salud, siendo responsable de forma integral, de todas las acciones de salud que se realicen en las personas, presta servicios en la atención primaria de la salud junto con la estructura de dirección de los Consejos Populares, se encarga de resolver los problemas más específicos de la población. [2]

1.1.2. Informatización del Sistema Nacional de Salud

Durante los últimos años un grupo de instituciones cubanas y el propio MINSAP han desarrollado sistemas encaminados a lograr la informatización de la salud. En todos los casos el objetivo ha sido proveer al Sistema Nacional de Salud de información confiable, consistente y oportuna para la toma de decisiones y el mejoramiento de los procesos médicos asistenciales, esto garantiza el incremento en la calidad y seguridad de la atención médica a la población. El Sistema Nacional de Salud se traza una nueva estrategia para alcanzar la informatización del sector alineada al proceso de informatización de la Sociedad Cubana, para esto ubica como centro de la misma al paciente y utiliza para su construcción las

últimas Tecnologías de la Información y las Comunicaciones (TIC), para el desarrollo de aplicaciones basadas en componentes y orientadas a servicios con el empleo de software libre. [3]

Como parte del proceso evolutivo de la informatización del SNS surge Infomed como un proyecto para interconectar a los centros de información y bibliotecas médicas, el cual constituye una red de alcance nacional. Infomed es la red de personas e instituciones que trabajan y colaboran para facilitar el acceso a la información y el conocimiento para mejorar la salud de los cubanos y de los pueblos del mundo, mediante el uso intensivo y creativo de las TIC.

1.2. Conceptos relacionados con el dominio del problema

1.2.1. Estadísticas de salud

Se conocen como estadísticas de salud a un conjunto de sistemas de registros continuos, complementarios, con datos provenientes de fuentes censales, que comprende al menos las siguientes áreas: [4]

Estadísticas de Población: Información numérica acerca de la composición y principales características de las agrupaciones humanas. Son objeto de estudio de la Demografía.

Estadísticas Vitales: Información numérica cuantificable sobre hechos vitales.

Hecho Vital: Todo hecho relacionado con el comienzo y fin de la vida del individuo y con los cambios de su estado civil que puedan ocurrirle durante su existencia.

Estadísticas de Morbilidad: Información numérica sobre las enfermedades, principales padecimientos, discapacidad y secuelas de enfermedades o de hechos accidentales o intencionales que se presentan en la población.

Estadísticas de Recursos: Información numérica sobre los recursos materiales y humanos disponibles y su aprovechamiento, que posee el Sistema Nacional de Salud para su gestión.

Estadísticas de Servicios: Información numérica acerca del volumen y calidad de los servicios de salud que se prestan.

Estadísticas de Vivienda: Información cuantificable relacionada con las viviendas en que el hombre habita y sus características.

Estadísticas de Saneamiento: Datos numéricos sobre las condiciones ambientales y sanitarias del entorno donde viven, trabajan, estudian o realizan otras actividades las comunidades humanas.

Estadísticas Económicas: Información numérica necesaria para el análisis de la actividad económica de salud. Ej.: Costos, Gastos, Inversiones, Exportaciones y otras.

1.2.2. Sistemas de Información Estadística (SIE)

Los SIE proporcionan los formatos estadísticos que contienen elementos de análisis, evaluación, comparación y control que facilitan un mejor desempeño del proceso de toma de decisiones. Como todo sistema ha de tener los siguientes componentes esenciales: entradas, procesador o procesos, salidas, retroalimentación, estar bien delimitado en su alcance y contar con definiciones claras y precisas. Demanda la actividad continua y adaptable a las circunstancias y necesidades.

Clasificación de los SIE.

Por el nivel jerárquico:

- Global.
- Ramal.

Por el alcance:

- Nacional.
- Territorial.

Por los medios técnicos en que se sustente:

- Manual.
- Automatizado.
- Mixto.

Por con su finalidad o propósito:

- Recursos.
- Servicios.
- Mortalidad.
- Morbilidad.

Por su contenido:

- Generales.
- Especiales.

De acuerdo con la organización y funcionalidad de la actividad estadística en Cuba, los SIE son de tres tipos:

Sistema de Información Estadística Nacional (SIEN)

- Comprende la elaboración de estadísticas y análisis destinados a satisfacer las necesidades informativas del Estado y del Gobierno a los efectos de conocer el comportamiento de los procesos económicos, demográficos y sociales, y especialmente para el control del plan de la economía nacional y del presupuesto, los compromisos estadísticos internacionales, a la población y otras instituciones.
- El Sistema de Información Estadística Nacional se aprueba anualmente mediante resolución del Jefe de la Oficina Nacional de Estadísticas.
- En la organización del Sistema de Información Estadística Nacional, la Oficina Nacional de Estadísticas vela cuidadosamente para evitar la proliferación de estadísticas innecesarias o que no correspondan realmente a las necesidades informativas de los órganos de dirección, así como para minimizar la carga informativa a los centros informantes.

Sistema de Información Estadística Territorial (SIET)

- Comprende la elaboración de las estadísticas destinadas a satisfacer los requerimientos informativos de los órganos del Estado y Gobierno a nivel territorial, por lo que este sistema consta de quince subsistemas independientes.
- Los Sistemas de Información Estadística Territorial se aprueban mediante resolución del Jefe de la Oficina Nacional de Estadísticas.

Sistema de Información Estadística Complementaria (SIEC)

- Comprende la elaboración de las estadísticas y análisis destinados a satisfacer las necesidades informativas de los órganos, organismos e instituciones estatales a los efectos del control administrativo de sus entidades.
- Dictan las normas específicas de sus sistemas informativos en el marco de lo que establecen las normas generales del Sistema Estadístico Nacional y controlan el funcionamiento de sus sistemas informativos.
- La Oficina Nacional de Estadísticas, como órgano rector, vela por el funcionamiento adecuado del Sistema de Información Estadística Complementaria, conoce y analiza los sistemas complementarios; trabaja para que en dichos sistemas no se elaboren estadísticas que puedan ser brindadas oportunamente mediante el procesamiento de la información primaria captada por el Sistema de Información Estadística Nacional.

1.2.3. Sistema de Información Estadística Complementaria (SIEC) de Salud

El SIEC de Salud cubano es ramal, de alcance nacional y mixto, ya que contiene subsistemas cuyo soporte técnico es manual solamente o manual y automatizado.

Características funcionales de la actividad estadística de la Salud en Cuba

El SIEC de Salud Cubano utiliza fundamentalmente el método de registro continuo y la aplicación de encuestas por muestreo, posee cobertura nacional y Departamentos de Estadística y Registros Médicos en todas las unidades e instituciones de salud. La certificación de los eventos regulados para su captación

y tratamiento estadístico posterior es universal, y el diseño y funcionamiento de los Sistemas de Información Estadística que soportan la actividad estadística en salud, se basan en los conceptos y atributos de la calidad de la información tales como, factibilidad, confiabilidad, pertinencia y racionalidad. El SIEC de Salud es un sistema diverso y voluminoso por los componentes que lo integran y que se expresan por la carga estadística, número de variables, alcance geográfico, áreas del conocimiento e indicadores de salida que provee, y es uno de los mayores SIEC del país, con alto costo de ejecución.

1.2.4. Modelo de flujo

Modelo o planilla en la cual se encuentran los indicadores a captar por parte del sistema, cada uno se corresponde con un subsistema de información de los que conforman el SIEC de Salud. Estos modelos son creados y modificados por parte de la Dirección Nacional de Registros Médicos y Estadística Sanitaria, en dependencia de las necesidades de información del propio MINSAP o de otras instituciones, que utilizan esta información con el objetivo de trazar estrategias y evaluar procedimientos.

1.3. Antecedentes del sistema

1.3.1. Antecedentes del sistema en el ámbito internacional

FlexCell Grid Control

FlexCell es un control de rejilla, flexible y rápido de usar. Proporciona funciones comprensivas como: la impresión, la vista preliminar de impresión, gráficos, fusión de celdas, celda virtual, ordenamiento. Así como la importación y exportación a XML, al libro de trabajo de Excel (ninguna necesidad de apoyo en tiempo de ejecución de Excel), entre otras. Provee 6 tipos de celdas y 9 objetos programados. [5]

FlexCell Grid Control soporta la interfaz estándar de Microsoft Windows ActiveX, puede ser usado en el código de aplicaciones en Visual Basic, Visual FoxPro, Visual C++, VB.net, C#.

Características principales:

- FlexCell ofrece 11 clases: la celda, gráficos, columnas, combo box, imágenes, PageBreak, configuración de páginas, rango, Report Title, filas y selección.
- Permite fusión de celdas y la personalización visual de la celda según la necesidad del usuario.
- El contenido de la rejilla se puede copiar, cortar y pegar libremente, y el texto puede ser copiado y pegado mutuamente entre FlexCell y otros programas como Excel, Bloc de notas.

Desventaja:

- Herramienta propietaria, se requiere la compra de licencias.

Sigma Grid: Rejilla de datos editable Ajax

Sigma Grid es una rejilla de datos Ajax para mostrar y editar una tabla con desplazamiento y ordenación. Escrito en JavaScript puro. Gracias a su extensibilidad, permite la sustitución del editor de celdas integrado con componentes externos personalizados (editores definidos por el usuario) y presentar los datos de las celdas como desee el cliente (renderización definida por el usuario). La gran cantidad de atributos y manejadores de eventos le ofrecen más flexibilidad con poca codificación. [6]

Características:

- Soporta Internet Explorer 6.0 o superior, Firefox 2.0 o superior, Safari 3.0 o superior, Opera9.0 o superior.
- Edición en la celda, navegación por los datos por medio del teclado, cabeceras anidadas multilínea, columnas ordenables o congeladas.
- Exporta e importa formatos CSV/XML/JSON, para soportar más edición dentro de la hoja de cálculos o cualquier otra aplicación.

Desventaja:

- Sólo para aplicaciones web.
- No es libre.

PC-Axis

Es un programa para el procesamiento de datos estadísticos que permite describir y explotar grandes cantidades de datos. Posibilita la organización de tablas y la confección de bases de datos estadísticas locales. Utilizado para visualizar las publicaciones que el Instituto Nacional de Estadísticas de España (INE) difunde en formato electrónico. Se incluye en todos los CD-ROM editados desde el año 2000 y también se puede descargar, es desarrollado por el Instituto de Estadística Sueco. [7]

Características:

- Sencilla interfaz de usuario basada en hoja de cálculo.
- Personalización de tablas. El usuario decide qué variables y valores intervienen y en qué orden.
- Confección de gráficos estadísticos.
- Confección de mapas temáticos (se necesita instalar el programa PX-Map y las colecciones de mapas).
- Exportación de tablas a una amplia variedad de formatos (Excel, HTML) y elabora gráficos, organiza tablas, establece bases de datos estadísticas locales.
- Compatible con Windows 95/98/ME/NT/2000/XP

Desventaja:

- La versión que se distribuye gratuitamente no incluye las opciones y las funciones para la creación de archivos, por lo que el PC-Axis básico recuerda al Adobe Acrobat Reader para uso estadístico, de manera que es relativamente fácil leer y convertir a Excel los datos, pero no se puede elaborar fácilmente las tablas.
- Existe limitación en la B.D, no se pueden presentar en pantalla tablas que sobrepasen 2000 filas o 50 columnas.
- A la hora de hacer selecciones del número de filas y columnas de la tabla resultante, en caso de sobrepasar el límite, la única forma de manejar la tabla es descargarla a la PC y tratarla con el software propio.
- Para que muchos de sus software adicionales (PX-Map y PX-Web) funcionen es necesario instalar previamente PC-Axis.
- Solo compatible con plataformas Windows.

Hojas de cálculos

Una hoja de cálculos (o programa de hojas de cálculos) es un software a través del cual se pueden usar datos numéricos y realizar cálculos automáticos de números que están en una tabla. También es posible automatizar cálculos complejos al utilizar una gran cantidad de parámetros y al crear tablas llamadas hojas de trabajo. Por lo tanto, la hoja de cálculo es una herramienta multiuso que sirve tanto para actividades de oficina que implican la organización de grandes cantidades de datos, como para niveles estratégicos y de toma de decisiones al crear representaciones gráficas de la información sintetizada. Las compañías de software han creado muchas hojas de cálculo. [8]

Las más importantes son:

- Microsoft Excel: paquete de oficina Microsoft Office.
- Sun: StarOffice Calc, paquete StarOffice.
- OpenCalc: paquete OpenOffice.
- IBM/Lotus 1-2-3: paquete SmartSuite.
- Corel Quattro Pro: paquete WordPerfect.
- KSpread: paquete KOffice, paquete gratuito de Linux.

MstGrid

Es un control de cuadrícula profesional que ofrece funciones completas para la creación de interfaces de fácil uso para el usuario, puede mostrar, editar, establecer formato, organizar e imprimir los datos tabulares. Se trata de un componente ActiveX que facilita el proceso de desarrollo de las aplicaciones. [9]

Características:

- Es compatible con los estilos visuales que imitan los estilos disponibles en Office XP como azul, plata y oliva.
- Proporciona funciones globales tales como:
 - ✓ Cambiar formatos y estilos de cada celda.

- ✓ Mostrar imágenes y datos en cada celda.
- ✓ Mover, cambiar tamaño, así como ocultar las filas y columnas.
- ✓ Importación y exportación a distintos formatos.
- ✓ Mezclar y separar celdas.
- ✓ Ofrece impresión y vista previa de impresión, incluye 45 propiedades relacionadas con la configuración de impresión.

Desventajas:

- Sólo compatible con Visual Basic.
- Sólo compatible con plataformas Windows

1.3.2. Antecedentes del sistema en el ámbito nacional

En Cuba las hojas de cálculos cobran vital importancia porque son programas que contienen una amplia gama de funcionalidades que pueden ser aplicadas en distintas esferas de la vida, principalmente Microsoft Excel, que permite realizar operaciones con números organizados en una tabla. Excel es un tabulador electrónico que está diseñado para confeccionar tablas, modelos, gráficos, pero además para confeccionar libros y desplazarse por las diferentes hojas. Este programa es muy utilizado por personal de economía y estadística fundamentalmente, ya que con su uso se puede automatizar el contenido de trabajo.

1.3.2.1. Antecedentes del sistema en la Universidad de las Ciencias Informáticas

Proyecto de la ONE

En la Facultad 3 de la UCI se desarrolla un proyecto para la ONE que se encarga de la captación de todas las estadísticas oficiales del país dentro de las cuales se encuentran las estadísticas de la salud, las mismas se rigen por las metodologías de la ONE. [10]

La aplicación desarrollada por este proyecto es totalmente dependiente de la red y se realiza bajo las características siguientes:

Características:

- Se utiliza como gestor de base de datos Postgres SQL
- El servidor Web utilizado es el Apache
- Desarrollado con los lenguajes PHP y JavaScript
- Para el desarrollo se utiliza el Framework Symfony
- Utiliza Componentes de Open Jacob
- Sistema generador de reportes dinámicos

Desventajas

- Dependencia total de la conectividad.
- Limitaciones de edición de los distintos modelos de flujo.

Módulos desarrollados en la UCI

En el curso 2007-2008, en la UCI, se desarrollaron un grupo de aplicaciones las cuales se corresponden con los módulos: Actividades de Cirugía y otras Atenciones y Servicios; Emergencia, Urgencia e Ingreso en el Hogar. Además de Infección Intrahospitalaria; Consulta Externa y Certificados Médicos por Invalidez Temporal a Trabajadores. Estos son los únicos subsistemas automatizados, de los más de 130 con que cuenta el MINSAP. Los mismos manejan grandes volúmenes de datos desde el nivel de unidad de salud hasta el nivel nacional. Además, para el desarrollo de cada uno de los módulos se emplea mucho tiempo, y si uno de los modelos de flujo correspondiente al subsistema desarrollado sufre algún cambio en los requisitos del negocio esto implicaría una reingeniería del sistema por completo y por consiguiente su reimplementación. [11]

Características

- Se desarrolla una aplicación Web para cada uno de los módulos.
- Basadas en arquitectura en capas.
- Desarrolladas con PHP 5, como lenguaje de programación.
- MySQL 5 como gestor de base de datos.

Desventajas

- Dependencia total de la conectividad.
- Interfaces y negocio definidos a la medida.

Subsistema Editor de Plantillas

Durante el curso 2008 – 2009 en la Facultad 7 de la UCI se decidió implementar un Subsistema Editor de Plantillas, perteneciente al Sistema para la Gestión y Análisis de Información Estadística en la Salud Pública Cubana, para la edición de los modelos de flujo que contienen los datos estadísticos que se generan en el ámbito de la salud. En dependencia de las necesidades planteadas este subsistema proporcionaría autonomía a los usuarios para establecer sus definiciones, sin necesidad de volver a iniciar un nuevo ciclo de desarrollo en la edición de una plantilla. Para ello se utilizó el componente FlexCell Grid Control, que brinda muchas de las funcionalidades básicas que se requieren.

1.4. Análisis crítico

Los componentes tabulares descritos anteriormente proporcionan controles de rejillas flexibles y rápidos para el uso y manejo de plantillas que controlen los distintos datos estadísticos, brindan interfaces de fácil entendimiento para usuarios que pueden tener mínimos conocimientos en informática, de ahí que proveen un sin número de funcionalidades muy sugerente tales como: formatos para impresión, exportación a XML, Excel y otros.

Sin embargo, a pesar de los múltiples beneficios en la mayoría de los casos, muchas de estas aplicaciones son propietarias lo que en este caso es un inconveniente para su utilización, pues el producto no puede ser utilizado libremente y el proceso de soporte que brindan sus proveedores no satisface las necesidades del usuario final del subsistema. Además, en la implementación de muchos de estos componentes se utilizan recursos propios de Windows que impide una posible migración hacia plataformas libres.

1.5. Tendencias tecnológicas y metodologías actuales

En este epígrafe se realiza un análisis detallado de los principales conceptos y tecnologías que pueden ser apropiados para el desarrollo del sistema. Se describe la arquitectura a utilizar, la metodología para el análisis y diseño del sistema, el lenguaje de programación que se empleará y las herramientas que apoyan el desarrollo del Componente.

1.5.1. Modelo Vista Controlador (MVC)

Este patrón de arquitectura de software permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos, esto proporciona múltiples vistas sobre un mismo modelo de datos. Los tres elementos esenciales de este patrón son los siguientes:

- Modelo: Administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado, usualmente formulados desde la vista, responde a instrucciones de cambio para cambiar el estado de estos datos, habitualmente desde el controlador.
- Vista: Presenta el modelo en un formato adecuado para interactuar con el usuario, usualmente un elemento de interfaz de usuario.
- Controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Tanto la vista como el controlador dependen del modelo, el cual no depende de otros conceptos o clases. Esta separación permite construir y probar el modelo independientemente de la representación visual.

1.5.2. Lenguajes

1.5.2.1. Lenguaje Unificado de Modelado. (UML 2 .0)

UML (Lenguaje Unificado de Modelado) es un lenguaje para especificar, visualizar, construir y documentar las diferentes etapas del desarrollo de un software, así como para el modelado de procesos de negocio u otros sistemas. UML reúne una colección de las mejores prácticas en la ingeniería que han sido utilizadas con éxito para modelar sistemas grandes y complejos porque cubre tanto objetos conceptuales como los

procesos de negocio y funciones del sistema, cubre también objetos concretos como clases en un lenguaje de programación, esquemas de bases de datos y componentes reusables de software. [12]

En las versiones previas se hacía un fuerte hincapié en que UML no era un lenguaje de programación, ya que un modelo creado con él no podía ejecutarse. En UML 2.0 esta asunción cambió en forma drástica y el lenguaje se modificó de manera tal que permitiera capturar mucho más comportamiento, con el fin de permitir la creación de herramientas que soportaran la automatización y la generación de código ejecutable a partir de modelos UML.

Ventajas de UML 2.0:

- Produce un aumento en la calidad del desarrollo.
- Reduce los costos del proyecto.
- Mejora en un 50% o más los tiempos totales de desarrollo.
- Permite especificar la estructura y el comportamiento del sistema y comunicarlo a todos los integrantes del proyecto.
- Brinda la posibilidad de obtener un "plano" del sistema.
- Permite dimensionar mejor los riesgos de un proyecto, tener un mejor rendimiento antes de construir el sistema.
- Facilita la documentación de las decisiones de la arquitectura del proyecto.
- Ofrece un mejor soporte a la planificación y control del proyecto.
- Permite realizar una verificación y validación del modelo realizado.

1.5.2.2. CSharp (C#)

C# es un lenguaje orientado a objetos, con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Se puede utilizar este lenguaje para crear aplicaciones clientes para Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, entre otras.

Como lenguaje orientado a objetos C# admite los conceptos de encapsulación, herencia y polimorfismo. El proceso de generación de C# es simple en comparación con el de C y C++ y es más flexible que en Java.

No hay archivos de encabezado independientes ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos.

Entre sus principales características se destacan:

- Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
 - ✓ El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al código fuente tales como ficheros de cabecera o ficheros IDL
 - ✓ El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código.
- Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario
- Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque respecto al de otros lenguajes como C++, es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales, sino que todo el código y datos han de precisarse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.
- Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades, eventos o atributos.
- Gestión automática de memoria: Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos, pues lenguaje proporciona un mecanismo de liberación de recursos determinista a través de la instrucción using. [13]

1.5.2.3. Lenguaje de Etiquetado Extensible (XML)

XML es un Lenguaje de Etiquetado Extensible muy simple pero estricto, que desempeña un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones y permite estructurar, almacenar e intercambiar información. [14]

1.5.3. Metodologías

1.5.3.1. Proceso Unificado de Rational (RUP)

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto a modo de plantilla que explica los pasos necesarios para terminar el proyecto. RUP es más que un proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. Está formado por componentes interconectados a través de interfaces y junto con el Lenguaje Unificado de Modelado UML. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es un refinamiento realizado por Rational Software del más genérico Proceso Unificado. [15]

El ciclo de vida de RUP se caracteriza por:

- Guiado por los casos de uso
- Centrado en la arquitectura
- Iterativo e incremental

RUP para organizar el proceso de construcción cuenta con cuatro fases, seis flujos de trabajos fundamentales y tres de apoyo.

Fases de desarrollo: Inicio, Elaboración, Construcción y Desarrollo.

Flujos de trabajo fundamentales: Modelado de negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Despliegue.

Flujos de trabajo de apoyo: Administración de Configuración y Cambios, Administración del proyecto, Ambiente.

1.5.4. Herramientas

1.5.4.1. Visual Studio 2008

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. [16]

Visual Studio permite a los desarrolladores crear aplicaciones desktop, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

1.5.4.2. Mono Migration Analyzer 1.2.2 (MOMA)

Herramienta analítica para migraciones de .NET a Linux, ayuda a identificar los problemas encontrados a la hora de portar las aplicaciones realizadas en la plataforma .NET al Framework Mono. Con esta herramienta se puede identificar las zonas que aún no están soportadas por Mono y permite a los desarrolladores cuantificar el número de cambios requeridos para ejecutar aplicaciones .NET en un entorno Linux.

Algo importante a considerar es que aunque MOMA garantice resultados confiables, se debe recordar siempre que la verdadera prueba consiste en ejecutar la aplicación sobre Mono.

1.5.4.3. Enterprise Architect 7.1 (EA)

Enterprise Architect es una herramienta CASE (Computer Aided Software Engineering) para el diseño y construcción de sistemas de software. EA soporta la especificación de UML 2.0 que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto. EA es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo y proporciona una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio. [17]

Algunas de las características claves de Enterprise Architect son:

- Crea elementos del modelo UML para un amplio alcance de objetivos.
- Ubica esos elementos en diagramas y paquetes.
- Documenta los elementos que se han creado. La documentación de alta calidad puede ser rápidamente exportada desde sus modelos en formato RTF e importar a Word para una personalización y presentación final.
- Se puede realizar ingeniería reversa del código existente en varios lenguajes tales como: C++, C#, Delphi, Java, Python, PHP, VB.NET y clases de Visual Basic.
- Permite estimar el tamaño de un proyecto en esfuerzo de trabajo en horas.

1.5.4.4. SourceGrid 4.22

SourceGrid es una herramienta colaborativa escrita en C# que permite a los programadores agregar capacidades básicas de rejilla a las aplicaciones implementadas en .NET. Con SourceGrid se puede mostrar y editar información, pues se utiliza para manipular y mostrar datos en forma de tabla. Además, agrega muchas características que los grid de .NET no incluyen. [18]

Características principales:

- Es posible personalizar el aspecto gráfico, el tipo de editor y el comportamiento contextual de cada celda.
- Soporta de manera nativa muchos tipos de datos.
- Cualquier control .NET puede utilizarlo como editor con pocas líneas de código.

- La altura y el ancho pueden ser personalizado de forma independiente para cada columna y fila, también puede calcular de forma automática el tamaño de las celdas.
- Apoya las operaciones automáticas de Copiar y Pegar.
- Con un poco de extensión soporta el enlace de datos.

Ventaja:

- Es una herramienta de código abierto, cuyo funcionamiento puede ser manipulado por cualquier persona que lo desee sin restricción, con la cual se tiene pleno acceso al código.

Desventaja:

- SourceGrid no tiene un diseñador, todo se debe hacer con el código.
- Lentitud en tiempo de ejecución.

En este capítulo se abordaron los aspectos fundamentales del Sistema Nacional de Salud en Cuba y los requisitos y premisas definidos para la informatización de la salud en el país con el objetivo de analizar el objeto de estudio definido. Se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión de la presente investigación, así como los antecedentes históricos e importancia de la información estadística en Cuba. Se estudiaron sistemas vinculados al campo de acción en el ámbito nacional e internacional y se realizó un análisis de las tecnologías, lenguajes y herramientas para dar cumplimiento al proceso de desarrollo.

Capítulo 2 Características del sistema

Al describir las principales características del sistema el presente capítulo tiene como objetivo presentar la propuesta de solución para el Componente Tabular, así como analizar los requisitos funcionales y no funcionales para un mejor entendimiento del sistema. Además, debido a la poca claridad de los procesos del negocio se desarrolla un Modelo de dominio.

2.1. Objetos de automatización

Se desea automatizar la edición y actualización de los modelos de flujo (plantillas), donde se recoge la información estadística de todas las unidades de salud del país. Además, se deben calcular sumas, por cientos, promedio y totales de la información.

2.2. Información que se maneja

La información utilizada en los procesos a automatizar es numérica, estadísticas de la salud propiamente, por lo que la misma debe ser tratada y analizada correctamente para que los resultados de la salud expresados estadísticamente sean exactos.

El SIEC de Salud cuenta con un grupo de modelos de flujo para la recopilación de la información, que constituyen el principal documento que deben generar cada uno de los estadísticos de la salud del país. En estos modelos de flujo se recoge información referente a datos de cantidades de pacientes tratados en diferentes áreas, así como promedios y consolidados de esta información.

2.3. Descripción del Sistema

Se propone una aplicación a partir de la que toda la información estadística es captada mediante los modelos de flujo, los cuales responden a las necesidades de información de cada uno de los subsistemas de información del SIEC de Salud, y que son creados por un conjunto de especialistas de la rama, ellos definen el formato y las reglas bajo las cuales será asentada la información captada, hecho por el cual surge el Subsistema Editor de Plantillas que responde a esta necesidad, y para ser utilizado por este se

desarrolla el Componente Tabular que tiene como objetivo brindar funcionalidades al SEP que le permitan realizar la edición de los distintos modelos de flujo o plantillas.

2.4. Modelo de dominio

El Modelo de dominio es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software. Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión).

El modelo de dominio se representa en UML con un Diagrama de Clases en los que se muestra:

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

En el Modelo de dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos. Cualquiera sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo de dominio contempla cualquiera de las soluciones analizadas. El modelo de dominio es global, se realiza para todos los casos de uso y no para uno en particular. [19]

El Componente Tabular forma parte de un sistema que no responde a las exigencias de un cliente, sino que se desarrolla basándose en las experiencias adquiridas en el desarrollo de aplicaciones como el Editor de Plantillas. Está dirigido al procesamiento de información estadística, y está diseñado para que satisfaga las necesidades de procesamiento de información estadística sin seguir necesariamente la estructura de ningún organismo o entidad en específico.

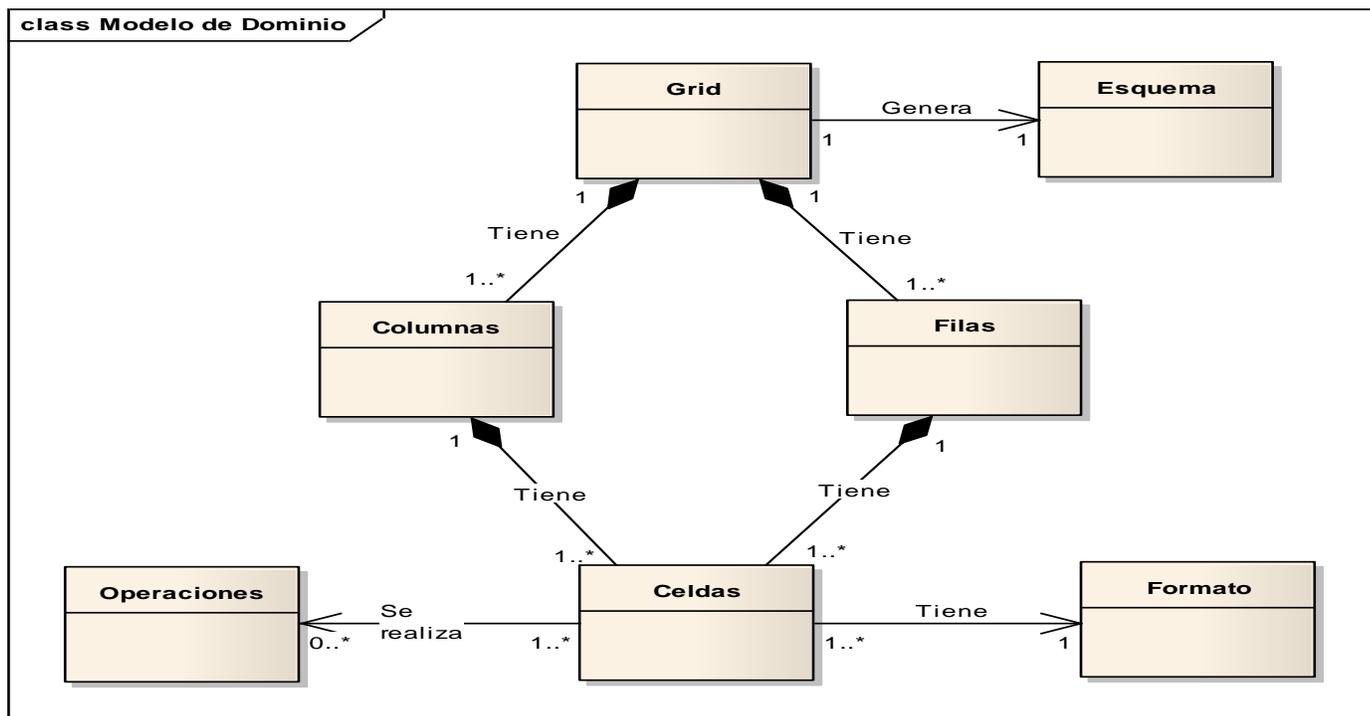


Fig. 1. Modelo de dominio.

2.4.1. Conceptos Fundamentales del dominio

Grid: representa una plantilla, en la cual se realizan las distintas operaciones para editar los datos.

Esquema: representa el formato XML de la rejilla.

Columna: representa un rango de celdas alineadas verticalmente.

Filas: representa un rango de celdas alineadas horizontalmente.

Celdas: constituye la unidad básica del grid.

Formato: el formato no es más que el estilo visual de la información que contienen las celdas.

Operaciones: representa todas las operaciones que se realizan sobre el grid o rejilla.

2.5. Especificación de los requisitos de software

2.5.1. Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

RF1 Insertar columna a la izquierda.

RF2 Insertar columna a la derecha.

RF3 Eliminar columna.

RF4 Insertar fila abajo.

RF5 Insertar fila arriba.

RF6 Eliminar fila.

RF7 Seleccionar celda.

RF8 Establecer color.

RF9 Establecer fuente.

RF10 Establecer alineación.

RF11 Combinar celdas.

RF12 Separar celda.

RF13 Bloquear celda.

RF14 Desbloquear celda.

RF15 Exportar.

RF16 Importar.

RF17 Inicializar grid.

RF18 Establecer tipo de celda

RF19 Seleccionar columna.

RF20 Seleccionar fila.

2.5.2. Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Usabilidad

RNF1 El Componente podrá ser utilizado por el Subsistema Editor de Plantillas.

RNF2 El Componente debe garantizar un acceso fácil y rápido y podrá ser utilizado por usuarios con pocos conocimientos informáticos.

Seguridad

➤ **Integridad**

RNF3 Se realizarán validaciones del tipo de datos que pueda permitir una celda, esto reduce a 0 la posibilidad de registrar datos erróneos.

➤ **Eficiencia**

RNF4 El Componente deberá ser rápido ante las solicitudes de los usuarios, el tiempo de respuesta para cada solicitud del usuario deberá ser menos de 5 segundos.

Soporte

RNF5 Una vez terminado el Componente se realizarán procesos de capacitación y mantenimiento de software.

Restricciones de diseño

RNF6 Para el desarrollo se define C# como lenguaje de programación.

RNF7 Se define Visual Studio como entorno de desarrollo integrado.

RNF8 Para realizar el modelado del software se utilizará la herramienta Enterprise Architect en su versión 7.1.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

RNF9 Se dispondrá de un Manual de Usuario que indicará como interactuar con las funcionalidades del Componente.

RNF10 Se dispondrá de la documentación del sistema realizada con la metodología de desarrollo RUP y así generar todo los artefactos correspondientes.

Interfaces de usuario

RNF11 La información será mostrada de manera comprensible y fácil de leer.

Interfaces Hardware

RNF12 Requerimientos mínimos, Ordenador Pentium o superior, 128 MB de memoria RAM, Disco Duro de 20 GB, copia de seguridad.

Interfaces Software

RNF13 Garantizar que el Componente se utilice sobre Sistema Operativo Windows XP Profesional Service Pack 2 o GNU/LINUX.

RNF14 La aplicación se implementará sobre .NET Framework 2.0 o superior.

Estándares Aplicables

RNF15 Para el desarrollo y la documentación del Componente se deben cumplir los estándares y pautas definidas por el Departamento SAS.

RNF16 Para las descripciones de casos de uso, y mensajes que debe emitir el sistema, se deben seguir las pautas de análisis definidas en el Departamento SAS.

2.6. Modelado del Sistema

El modelado del sistema describe cuáles son las personas o grupos de personas, instituciones o sistemas relacionados ya existentes que interactúan con los procesos que el sistema deberá realizar. Así como la relación que guardan los mismos.

2.6.1. Actores del sistema

A continuación se define y describen los actores del sistema.

Actor	Descripción
Diseñador	El diseñador es el encargado de utilizar o ejecutar todos los métodos u operaciones del Componente Tabular, pertinentes a la creación y edición de las plantillas.

2.6.2. Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema representa gráficamente los procesos y su interacción con los actores.

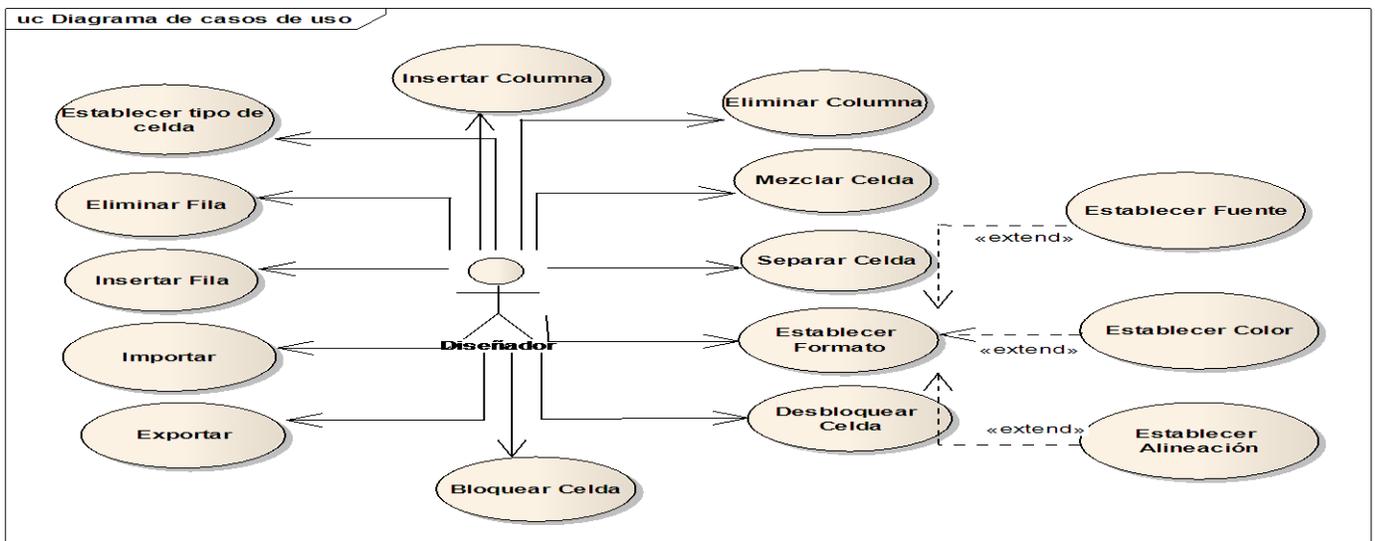


Fig. 2. Diagrama de casos de uso del sistema

2.6.3. Especificaciones de los casos de uso

2.6.3.1. Descripción del caso de uso: Insertar columna

Caso de Uso:	Insertar columna.	
Actor:	Diseñador.	
Resumen:	El caso de uso inicia cuando el Diseñador desea Insertar una columna, para insertar la columna a la izquierda o a la derecha debe estar seleccionada la celda o columna. El caso de uso finaliza cuando la columna ha sido insertada.	
Precondiciones:	Se debe seleccionar una celda o columna. Seleccionar la opción Insertar Columna. El grid o rejilla deberá estar inicializado.	
Referencias	RF1, RF2, RF7, RF17, RF19.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección "Insertar Columna"		
Acción del Actor	Respuesta del Sistema	
1. El Diseñador escoge la celda o columna en el grid, luego presiona clic derecho y selecciona la opción Insertar columna.	2. Se muestran las opciones de insertar columna a la derecha o izquierda: <ul style="list-style-type: none"> • En caso de insertar la columna a la izquierda ver Alternativa 1 "Insertar columna a la izquierda". • En caso de insertar la columna a la derecha ver Alternativa 2 "Insertar columna a la derecha". 	
Flujos Alternos		
Alternativa 1 "Insertar columna a la izquierda".		
Acción del Actor	Respuesta del Sistema	

1. El Diseñador selecciona la opción insertar columna a la izquierda.	2. Inserta la columna a la izquierda de la columna o celda seleccionada.
Prototipo de Interfaz	
Alternativa 2 “Insertar columna a la derecha”.	
Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona la opción insertar columna a la derecha.	2. Inserta la columna a la derecha de la columna o celda seleccionada.
Prototipo de Interfaz	
Poscondiciones	La columna es insertada.

2.6.3.2. Descripción del caso de uso: Eliminar columna

Caso de Uso:	Eliminar columna
Actor:	Diseñador
Resumen:	El caso de uso inicia cuando el diseñador desea eliminar una o varias columnas, debe estar seleccionada la celda o columna, la columna eliminada es la que está seleccionada en el momento de realizar esta acción. El caso de uso finaliza cuando se elimina la columna seleccionada.
Precondiciones:	Se debe seleccionar una celda o columna. Se debe seleccionar la opción Eliminar columna. El grid o rejilla deberá estar inicializado.
Referencias	RF3, RF7, RF17, RF19.
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Eliminar columnas”	
Acción del Actor	Respuesta del Sistema
1. El Diseñador escoge la celda o columna en el grid, luego presiona clic derecho y selecciona	2. Muestra el mensaje de advertencia “Se eliminará la columna seleccionada. Al seleccionar Aceptar se perderán todos

la opción Insertar columna.	los datos. ¿Desea continuar?” y permite: <ul style="list-style-type: none"> • Aceptar la eliminación de la columna. • Cancelar la operación. Ver Alternativa 1: “Cancelar operación.”
3. Selecciona la opción de aceptar la eliminación de la columna.	4. Se elimina la columna.
Flujo Alternos.	
Alternativa 1: “Cancelar operación.”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Cancelar operación.	2. Regresa a la vista anterior.
Poscondiciones	La columna es eliminada.

2.6.3.3. Descripción del caso de uso: Insertar fila

Caso de Uso:	Insertar fila
Actor:	Diseñador
Resumen:	El caso de uso inicia cuando el diseñador desea Insertar una Fila, para insertar la fila abajo o arriba debe estar seleccionada la celda o fila. El caso de uso finaliza cuando se inserta la fila.
Precondiciones:	Se debe seleccionar una celda o fila. Se debe seleccionar la opción Insertar fila. El grid o rejilla deberá estar inicializado.
Referencias	RF4, RF5, RF7, RF17, RF20.
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Insertar fila”	
Acción del Actor	Respuesta del Sistema

1. El Diseñador escoge la fila o celda en el grid, presiona clic derecho y selecciona la opción Insertar fila.	2. Se muestran las opciones de insertar fila abajo y arriba. <ul style="list-style-type: none"> • En caso de insertar la fila debajo de la celda seleccionada ver Alternativa 1 “Insertar fila abajo”. • En caso de insertar la fila arriba de la celda seleccionada ver Alternativa 2 “Insertar fila arriba”.
--	--

Flujos Alternos

Alternativa 1 “Insertar fila abajo”.

Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona la opción insertar fila abajo.	2. Inserta la fila debajo de la fila o celda seleccionada.

Prototipo de Interfaz

Alternativa 2 “Insertar fila arriba”.

Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona la opción insertar fila arriba.	2. Inserta la fila arriba de la fila o celda seleccionada.

Prototipo de Interfaz

Poscondiciones	La fila es insertada.
----------------	-----------------------

2.6.3.4. Descripción del caso de uso: Eliminar fila

Caso de Uso:	Eliminar fila.
Actor:	Diseñador.
Resumen:	El caso de uso inicia cuando el diseñador desea una eliminar fila o varias filas, debe estar seleccionada la celda o fila, la fila eliminada es la que está seleccionada en el momento de realizar esta acción. El caso de uso finaliza cuando se elimina la fila seleccionada.

Precondiciones:	Se debe seleccionar una celda o fila. Se debe seleccionar la opción Eliminar fila. El grid o rejilla deberá estar inicializado.
Referencias	RF6, RF7, RF17, RF20.
Prioridad	Crítico.
Flujo Normal de Eventos	
Sección "Eliminar fila"	
Acción del Actor	Respuesta del Sistema
1. El Diseñador escoge la fila o celda en el grid, presiona clic derecho y selecciona la opción eliminar fila.	2. Muestra el mensaje de advertencia "Se eliminará la fila seleccionada. Al seleccionar Aceptar se perderán todos los datos. ¿Desea continuar?" y permite: <ul style="list-style-type: none"> • Aceptar la eliminación de la fila. • Cancelar la operación. Ver Alternativa 1: "Cancelar operación."
3. Selecciona la opción de aceptar la eliminación de fila.	4. Se elimina la fila.
Flujo Alternos.	
Alternativa 1: "Cancelar operación."	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Cancelar operación.	2. Regresa a la vista anterior.
Poscondiciones	La fila es eliminada.

2.6.3.4. Descripción del caso de uso: Establecer formato

Caso de Uso:	Establecer formato
Actor:	Diseñador
Resumen:	El caso de uso inicia cuando el Diseñador desea modificar el estilo de la

	información que se encuentra en una plantilla. Para ello debe estar seleccionada la celda o rango de celda. El caso de uso finaliza cuando se termine de ejecutar la acción.
Precondiciones:	Se debe seleccionar una celda o varias celdas. Se debe seleccionar la opción. El grid o rejilla deberá estar inicializado.
Referencias	RF7, RF8, RF9, RF10, RF17, RF19, RF20.
Prioridad	Crítico

Flujo Normal de Eventos

Sección “Establecer formato”

Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona el menú Editar.	2. Se muestran las opciones de formato. <ul style="list-style-type: none"> • En caso de establecer la Fuente de la información seleccionada ver Alternativa 1 “Establecer fuente”. • En caso de establecer Color de la información seleccionada ver Alternativa 2 “Establecer color”. • En caso de establecer Alineación de la información seleccionada ver Alternativa 4 “Establecer alineación”.

Flujos Alternos

Alternativa 1 “Establecer fuente”.

Acción del Actor	Respuesta del Sistema
1. El Diseñador escoge la celda o rango de celdas a la cual desea darle formato.	
2. El Diseñador selecciona la opción Fuente en el menú Estilos	3. Se muestra una interfaz con los tipos de letras, las opciones de aplicar el formato de negrita, subrayado y cursiva al texto

	seleccionado y tamaño que se le puede aplicar.
4. El Diseñador selecciona las propiedades de fuente con las que desea darle formato a la información	5. Cambia el formato de la información seleccionada por el Diseñador.
Prototipo de Interfaz	
Alternativa 2 “Establecer color”.	
Acción del Actor	Respuesta del Sistema
1. El Diseñador escoge la celda o rango de celdas a la cual desea cambiar el color.	
2. El Diseñador selecciona la opción Color en el menú Estilos	3. Se muestra una interfaz con los colores que puede aplicar a la información seleccionada.
4. El Diseñador selecciona el color que desea darle a la información	5. Cambia la información con el color seleccionado por el Diseñador
Prototipo de Interfaz	
Alternativa 4 “Establecer alineación”.	
Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona la celda o rango de celdas a la cual desea cambiar la alineación.	
2. El Diseñador selecciona en el menú Alineado el tipo de alineación que desea darle a la celda o rango de celdas.	3. Cambia la alineación de la celda o rango de celdas seleccionadas.
Prototipo de Interfaz	
Poscondiciones	Cambia el formato de la celda o rango de celdas seleccionadas.

2.6.3.5. Descripción del caso de uso: Combinar celdas

Caso de Uso:	Combinar celdas
Actores:	Diseñador
Resumen:	El caso de uso inicia cuando el diseñador desea combinar un rango de celdas, las celdas que se combinan son aquellas que están seleccionadas en ese momento. El caso de uso finaliza cuando se ejecuta la operación.
Precondiciones:	<p>Seleccionar rango de celdas.</p> <p>Se debe seleccionar la opción Combinar celdas.</p> <p>El grid o rejilla deberá estar inicializado.</p>
Referencias	RF7, RF11, RF17, RF19, RF20.
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Combinar celdas”	
Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona rango de celdas que desea combinar en el grid.	
2. El Diseñador presiona clic derecho y selecciona la opción Mezclar.	<p>3. En caso de haber más de un valor en el rango de celdas seleccionado, se muestra un mensaje “La selección contiene valores de datos múltiples. Al combinar en una única celda se conservarán solamente los datos del extremo superior izquierdo.”, muestras dos opciones:</p> <ul style="list-style-type: none"> • Aceptar. • Cancelar. <p>En caso de seleccionar la opción cancelar ver Alternativa 1 “Cancelar mezclar”</p> <p>En caso de existir un solo valor o no haber valores ver Alternativa 2 “Rango sin valores o un solo valor”</p>

	<p>Si el rango de celda seleccionado no está correcto ver Alternativa 3 “Rango incorrecto”.</p> <p>Si el rango de celda seleccionado contiene alguna celda bloqueada ver Alternativa 4 “Rango con celdas bloqueadas”.</p>
4. Selecciona la opción aceptar.	5. Combina el rango de celdas que están seleccionadas.
Prototipo de Interfaz	
Flujos Alternos	
Alternativa 1 “Cancelar mezclar”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Cancelar operación.	2. Cancela la operación mezclar y conserva los datos de las celdas o rango de celdas.
Prototipo de Interfaz	
Alternativa 2 “Rango sin valores o un solo valor”	
Acción del Actor	Respuesta del Sistema
	1. Combina el rango de celdas que están seleccionadas y mantiene el valor que exista dentro del rango.
Prototipo de Interfaz	
Alternativa 3 “Rango incorrecto”	
Acción del Actor	Respuesta del Sistema
	1. Muestra un mensaje “No se puede cambiar parte de una celda combinada.”
Prototipo de Interfaz	
Alternativa 4 “Rango con celdas bloqueadas”	
Acción del Actor	Respuesta del Sistema
	1. Muestra un mensaje “No se puede combinar el

	rango seleccionado, contiene celdas bloqueadas.”
Prototipo de Interfaz	
Poscondiciones	Se mezcla el rango de celdas.

2.6.3.7. Descripción del caso de uso: Separar celdas

Caso de Uso:	Separar celda
Actores:	Diseñador
Resumen:	El caso de uso inicia cuando el diseñador desea separar un rango de celdas, las celdas que se separan son aquellas que están seleccionadas en ese momento. El caso de uso finaliza cuando se ejecuta la operación.
Precondiciones:	Para separar un rango de celda tiene que haber sido previamente mezclado. Seleccionar celda o rango de celdas. Se debe seleccionar la opción Separar celdas. El grid o rejilla deberá estar inicializado.
Referencias	RF7, RF12, RF17, RF19, RF20.
Prioridad	Crítico

Flujo Normal de Eventos

Sección “Separar celda”

Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona rango de celdas que desea separar en el grid.	
2. El Diseñador presiona clic derecho y selecciona la opción separar.	3. Separa el rango de celdas que están seleccionadas.

Prototipo de Interfaz

Poscondiciones	Se separa el rango de celdas.
-----------------------	-------------------------------

2.6.3.8. Descripción del caso de uso: Bloquear celda

Caso de Uso:	Bloquear celda.
Actores:	Diseñador.
Resumen:	El caso de uso inicia cuando el diseñador necesite Bloquear una celda o rango de celdas, las celdas que se bloquean son aquellas que están seleccionadas en ese momento. El caso de uso finaliza cuando se ejecuta la operación.
Precondiciones:	Se debe seleccionar una celda o un rango de celdas. Se debe seleccionar la opción Bloquear celdas. El grid o rejilla deberá estar inicializado.
Referencias	RF7, RF13, RF17, RF19, RF20.
Prioridad	Crítico
Flujo Normal de Eventos.	
Sección "Bloquear celda".	
Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona la celda o rango de celdas que desea bloquear en el grid.	
2. El Diseñador presiona clic derecho y selecciona la opción Bloquear.	3. Bloquea la celda o rango de celdas que están seleccionada.
Prototipo de Interfaz	
Poscondiciones	Se bloquea el rango de celdas.

2.6.3.9. Descripción del caso de uso: Desbloquear celda

Caso de Uso:	Desbloquear celda.
Actores:	Diseñador.
Resumen:	El caso de uso inicia cuando el diseñador necesite Desbloquear una celda o rango de celdas, las celdas que se desbloquean son aquellas que están seleccionadas en ese momento. El caso de uso finaliza cuando se ejecuta la operación.
Precondiciones:	Para desbloquear una celda o rango de celdas tiene que haber sido previamente

	<p>bloqueado.</p> <p>Se debe seleccionar una celda o un rango de celdas.</p> <p>Se debe seleccionar la opción Desbloquear celdas.</p> <p>El grid o rejilla deberá estar inicializado.</p>
Referencias	RF7, RF14, RF17, RF19, RF20.
Prioridad	Crítico
Flujo Normal de Eventos.	
Sección "Desbloquear celda".	
Acción del Actor	Respuesta del Sistema
1. El Diseñador selecciona una celda o rango de celdas que desea desbloquear en el grid.	
2. El Diseñador presiona clic derecho y selecciona la opción Desbloquear.	3. Desbloquea la celda o rango de celdas que están seleccionada.
Prototipo de Interfaz	
Poscondiciones	Se desbloquea el rango de celdas.

2.6.3.10. Descripción del caso de uso: Exportar

Caso de Uso:	Exportar
Actores:	Diseñador.
Resumen:	El caso de uso inicia cuando el Diseñador selecciona en el menú Exportar, porque no exista conexión y desee exportar la plantilla a algún soporte extraíble (Memorias, Disquetes) o porque desea continuar la edición de alguna plantilla desde otro lugar. El Diseñador también puede seleccionar la opción Exportar para guardar en el directorio local alguna plantilla que aún no esté lista para publicar. El caso de uso finaliza cuando la plantilla es enviada al directorio local o al soporte extraíble que el Diseñador desee.
Precondiciones:	El grid o rejilla deberá estar inicializado. Seleccionar la opción Exportar.
Referencias	RF15, RF17.

Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Exportar”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Exportar del menú Archivo.	2. Se muestra una interfaz donde el Diseñador debe introducir la periodicidad y fecha en la que se comenzará a llenar la plantilla.
3. El Diseñador introduce los datos.	4. Se muestra una interfaz donde se encuentran las ubicaciones donde el Diseñador puede exportar las plantillas. (Dentro de las ubicaciones que se muestran están los soportes extraíbles y el directorio local).
5. Selecciona la ubicación en la que desea exportar la plantilla.	6. La plantilla se envía a la ubicación seleccionada.
Prototipo de Interfaz	
Poscondiciones	La plantilla es exportada.

2.6.3.11. Descripción del caso de uso: Importar

Caso de Uso:	Importar
Actores:	Diseñador
Resumen:	El caso de uso inicia cuando el Diseñador necesita obtener de un soporte extraíble o de alguna ubicación del ordenador una plantilla. El caso de uso finaliza cuando la plantilla es importada.
Precondiciones:	Seleccionar la opción Importar. El grid o rejilla deberá estar inicializado.
Referencias	RF16, RF17.
Prioridad	Crítico
Flujo Normal de Eventos	

Sección "Importar"	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción Importar del menú Archivo.	2. Se muestra una interfaz con las ubicaciones donde se pueden encontrar las plantillas. (Dentro de las ubicaciones que se muestran están los soportes extraíbles y el directorio local).
3. Selecciona la ubicación donde se encuentra la plantilla.	4. Se muestra listado de las plantillas
5. Selecciona la plantilla que desea importar.	6. Se muestra en una interfaz la plantilla deseada.
Prototipo de Interfaz	
Poscondiciones	La plantilla es importada.

En este capítulo se identificaron los principales conceptos del Componente Tabular. Además, se mostraron las descripciones de cada una de las clases que intervienen en el modelo de dominio. Se presentaron los requisitos funcionales y no funcionales que indican las capacidades o condiciones que el sistema debe cumplir y las propiedades o cualidades que el producto debe tener respectivamente. También se seleccionaron los requisitos o grupos de requisitos funcionales con los que se va a conformar los casos de uso del sistema al igual que sus actores, y se describen los casos de uso.

Capítulo 3 Diseño del sistema

Con el diseño del sistema se pretende profundizar mediante diagramas todo el flujo de los distintos procesos. A través de los diagramas de interacción se describe gráficamente la interacción entre los actores y el sistema, también se describen las distintas clases asociadas al diseño del sistema.

3.1. Estructura del diseño

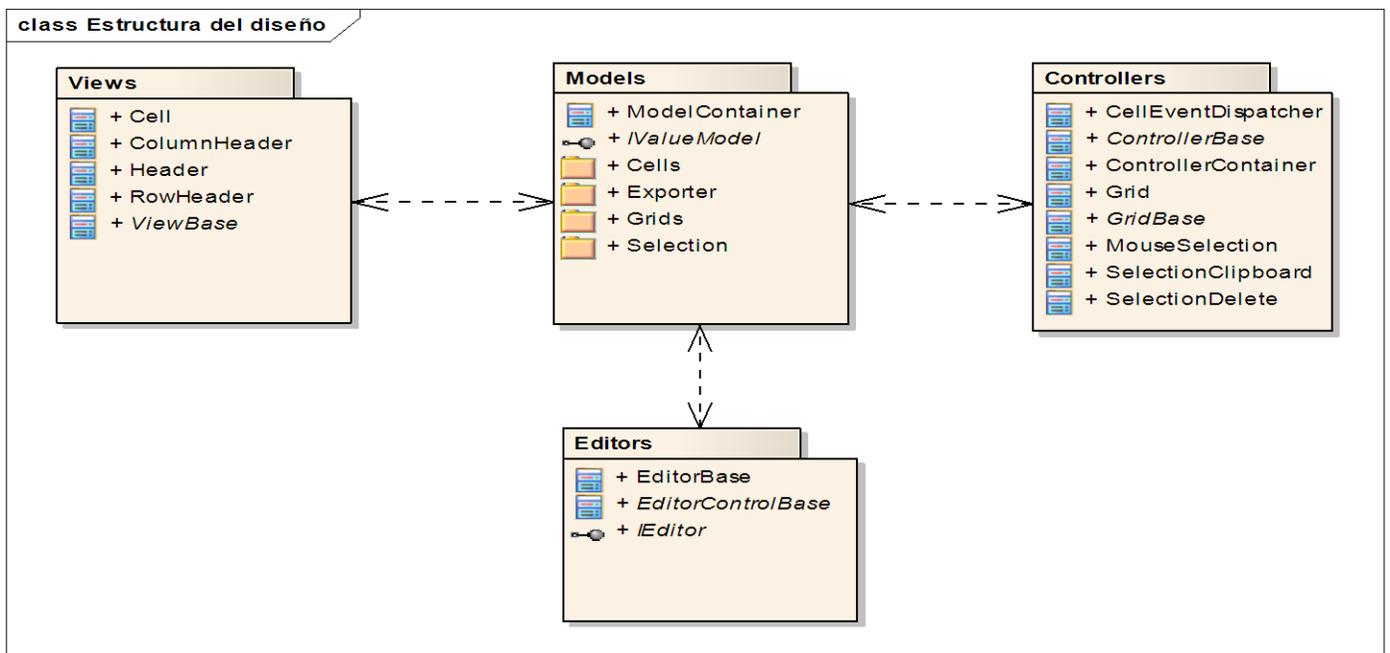


Fig. 3. Estructura del diseño

Cada celda del grid está compuesta de cuatro partes fundamentales basadas en un patrón Modelo-Vista-Controlador modificado:

Modelo: Clase que maneja el valor de la celda y es accesible por los otros componentes. Su propósito es tener los datos independientes del objeto celda.

Vista: Clase que contiene todas las propiedades visuales de la celda. Tiene como objetivo separar el código de elementos de diseño del resto del código y poder compartir la misma vista entre varias celdas.

Controlador: Define el comportamiento de cada celda, para esto tiene en cuenta los eventos que hay asociados a la misma.

Editor: Clase utilizada para personalizar propiedades de edición. Usualmente interactúa con la clase Modelo para manejar las conversiones necesarias, particularmente las conversiones de cadenas.

3.2. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en los requisitos no funcionales y otras restricciones del entorno de implementación que tienen impacto en el sistema, por tanto, define las clases del diseño que conformarán el sistema a implementar.

3.3. Diagramas de clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases del software y contienen las clases, atributos, métodos, navegabilidad y dependencias existentes entre ellas. A continuación se representan los diagramas de clase del diseño por casos de uso del sistema.

3.3.1. CUS Insertar columna



Fig. 3. Diagrama de clases del diseño del CUS Insertar Columna.

3.3.2. CUS Eliminar columna

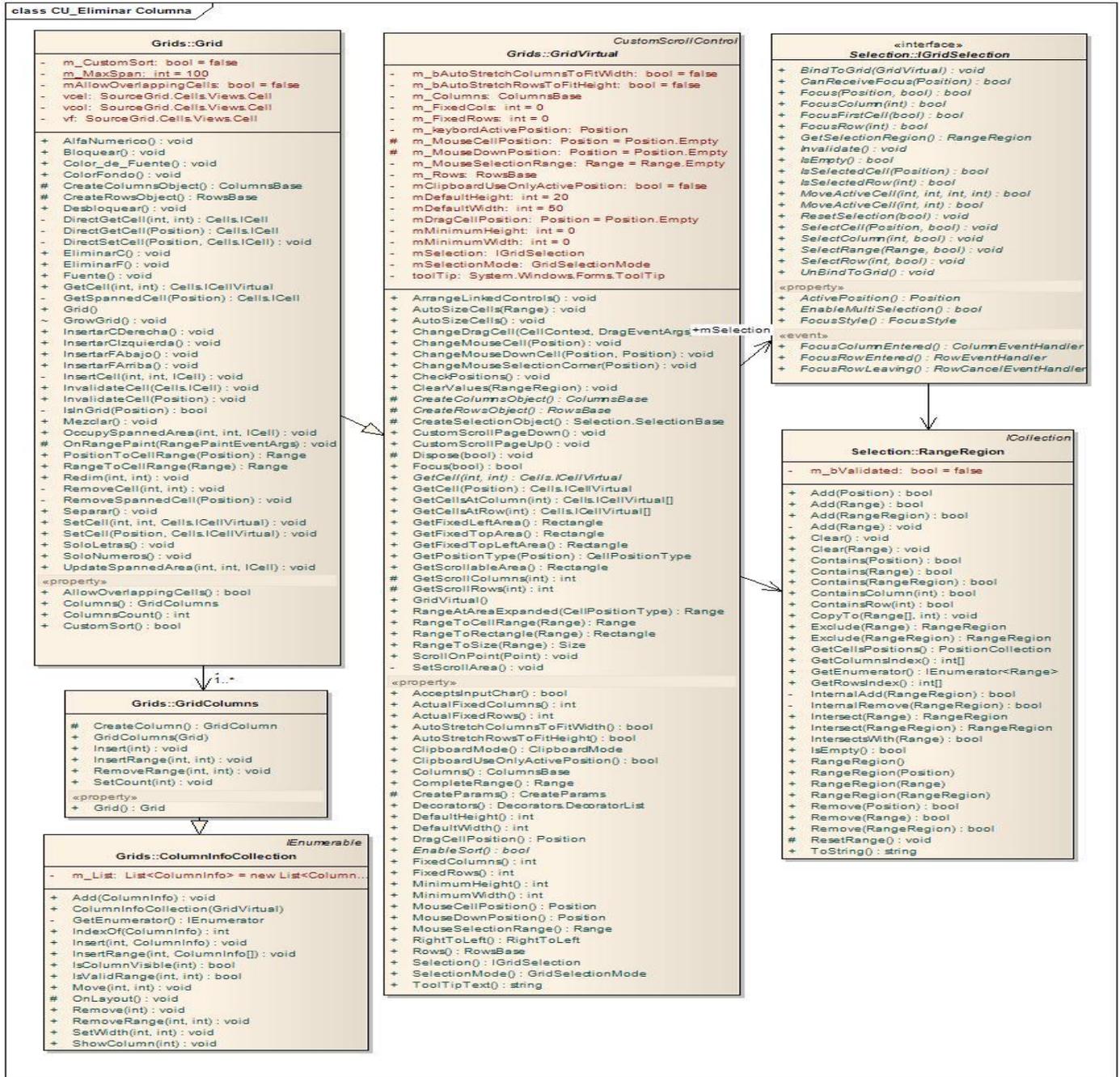


Fig. 4. Diagrama de clases del diseño del CUS Eliminar Columna.

3.3.3. CUS Insertar fila

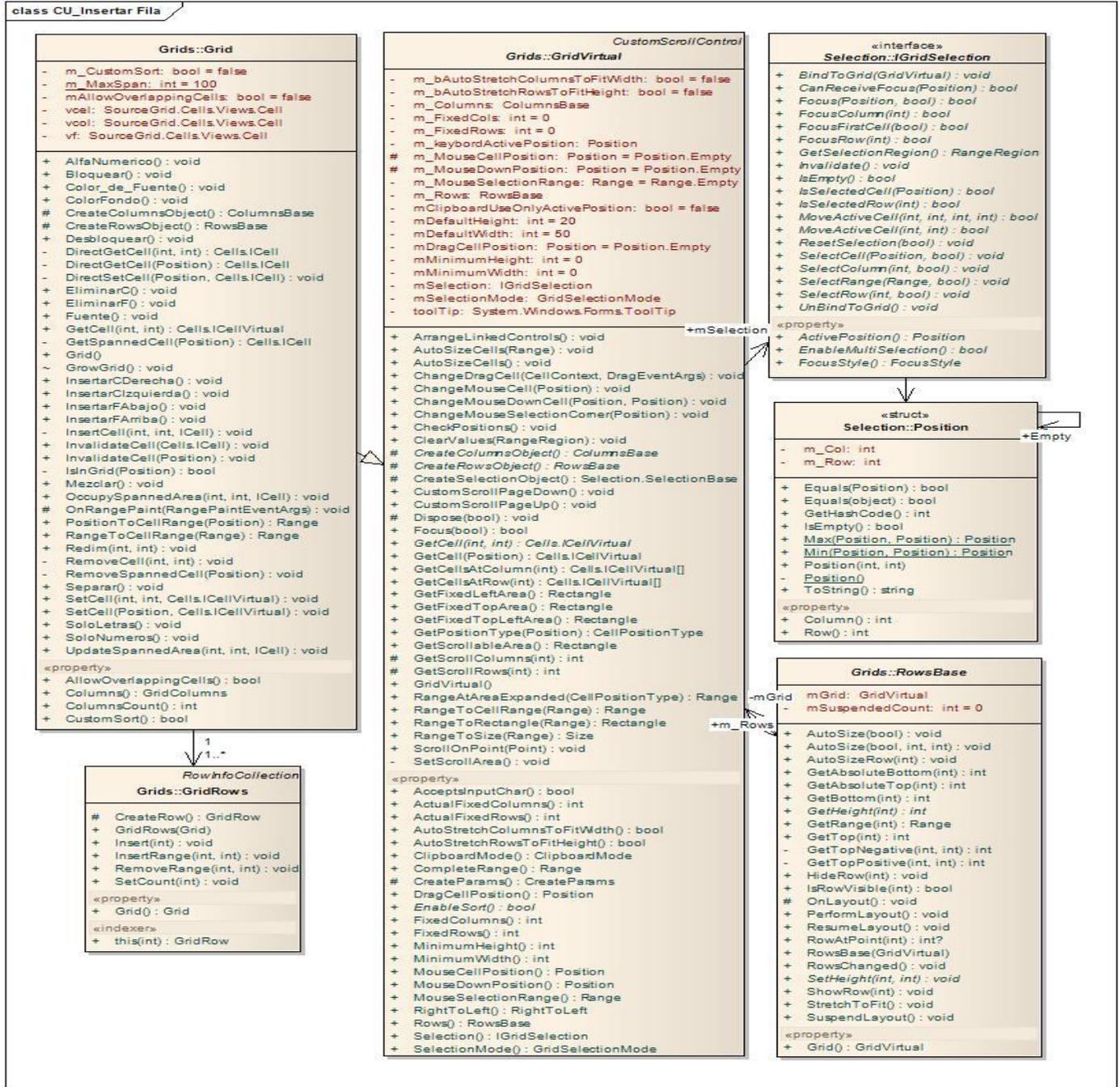


Fig. 5. Diagrama de clases del diseño CUS Insertar Fila

3.3.4. CUS Eliminar fila



Fig. 7. Diagrama de clases del diseño CUS Eliminar Fila

3.3.5. CUS Establecer formato

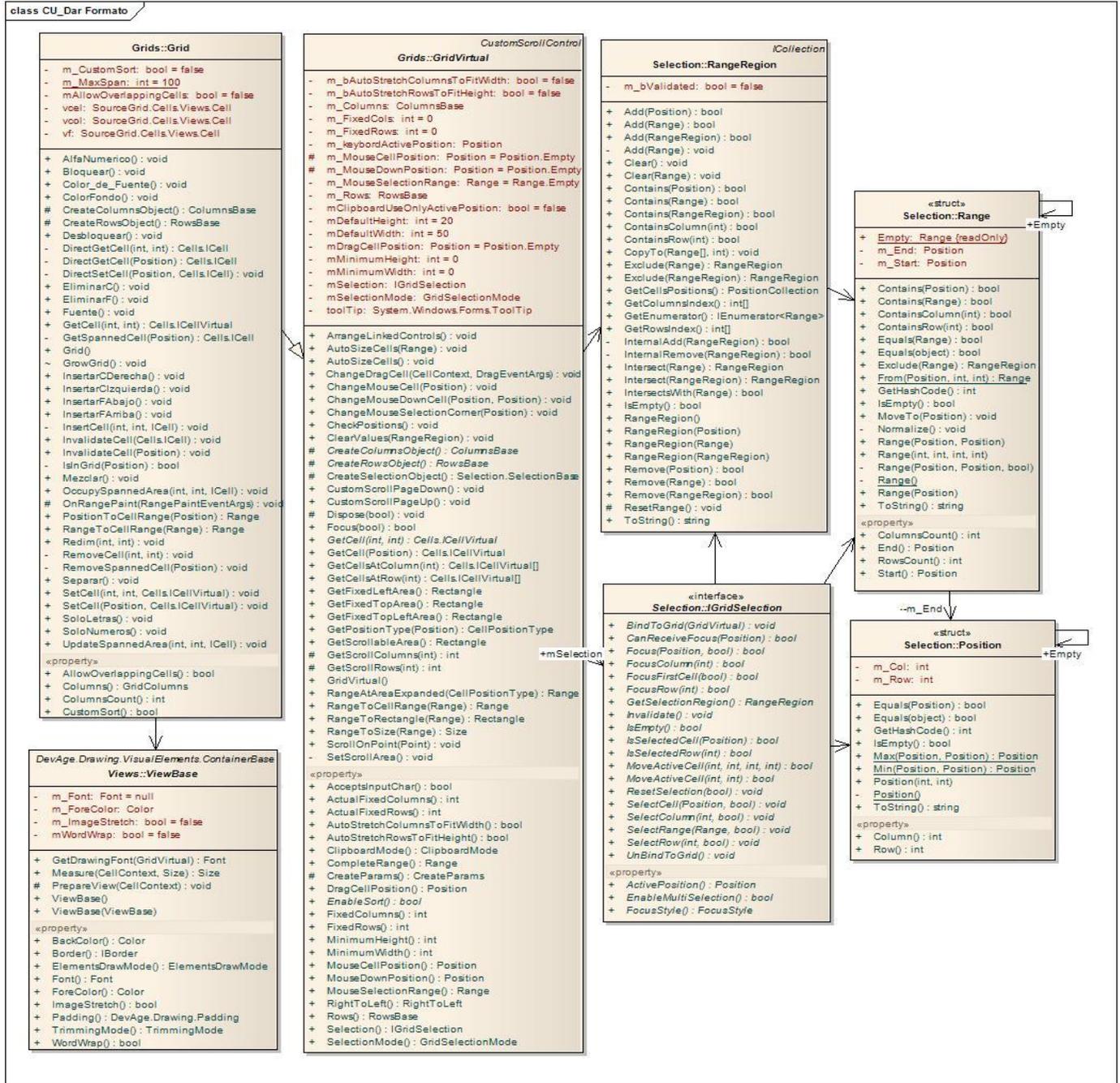


Fig. 8. Diagrama de clases del diseño CUS Establecer formato

3.3.6. CUS Combinar celdas



Fig. 9. Diagrama de clases del diseño CUS Combinar celdas

3.3.7. CUS Separar celda

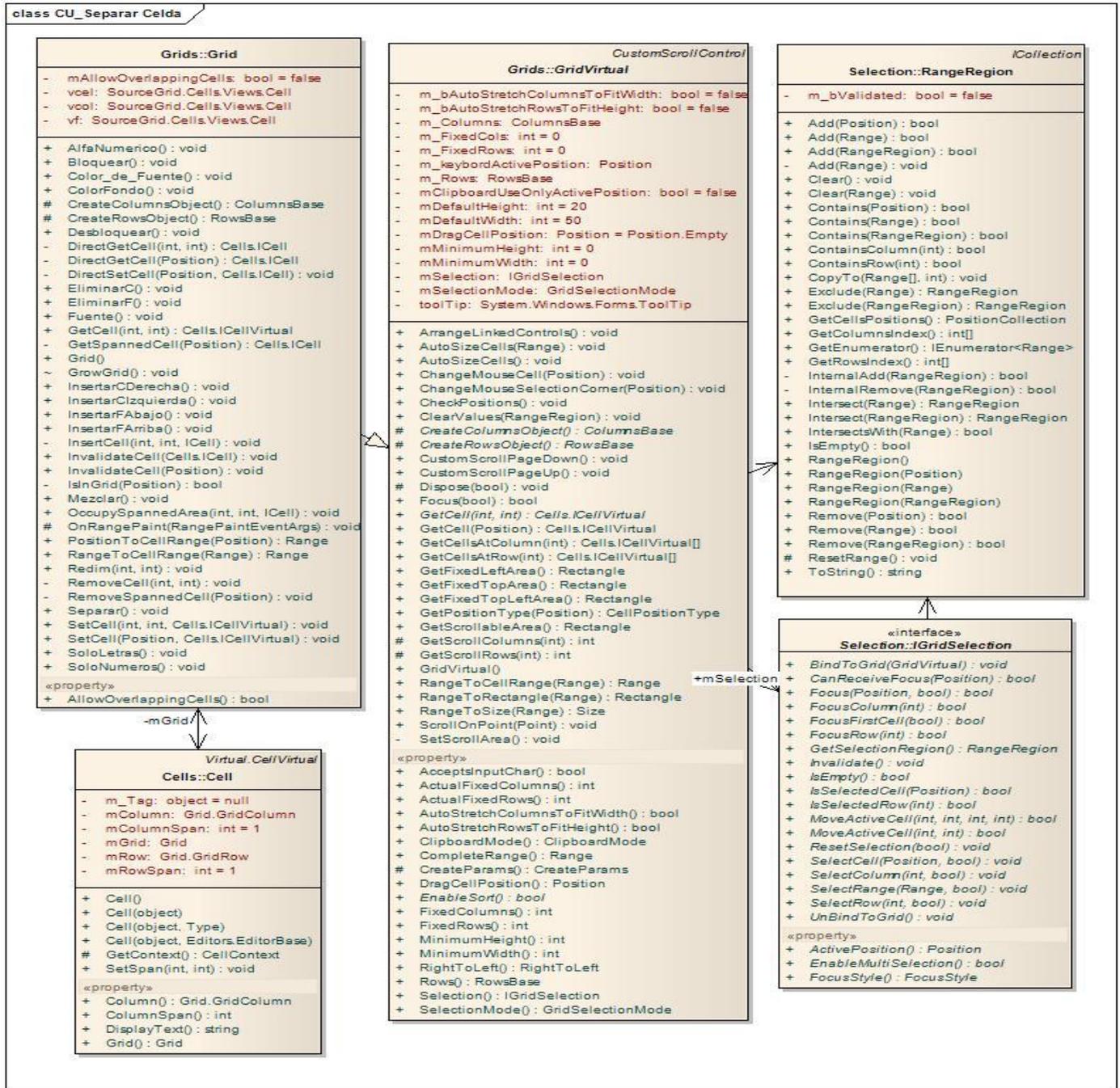


Fig. 10. Diagrama de clases del diseño CUS Separar celda

3.3.8. CUS Bloquear celda

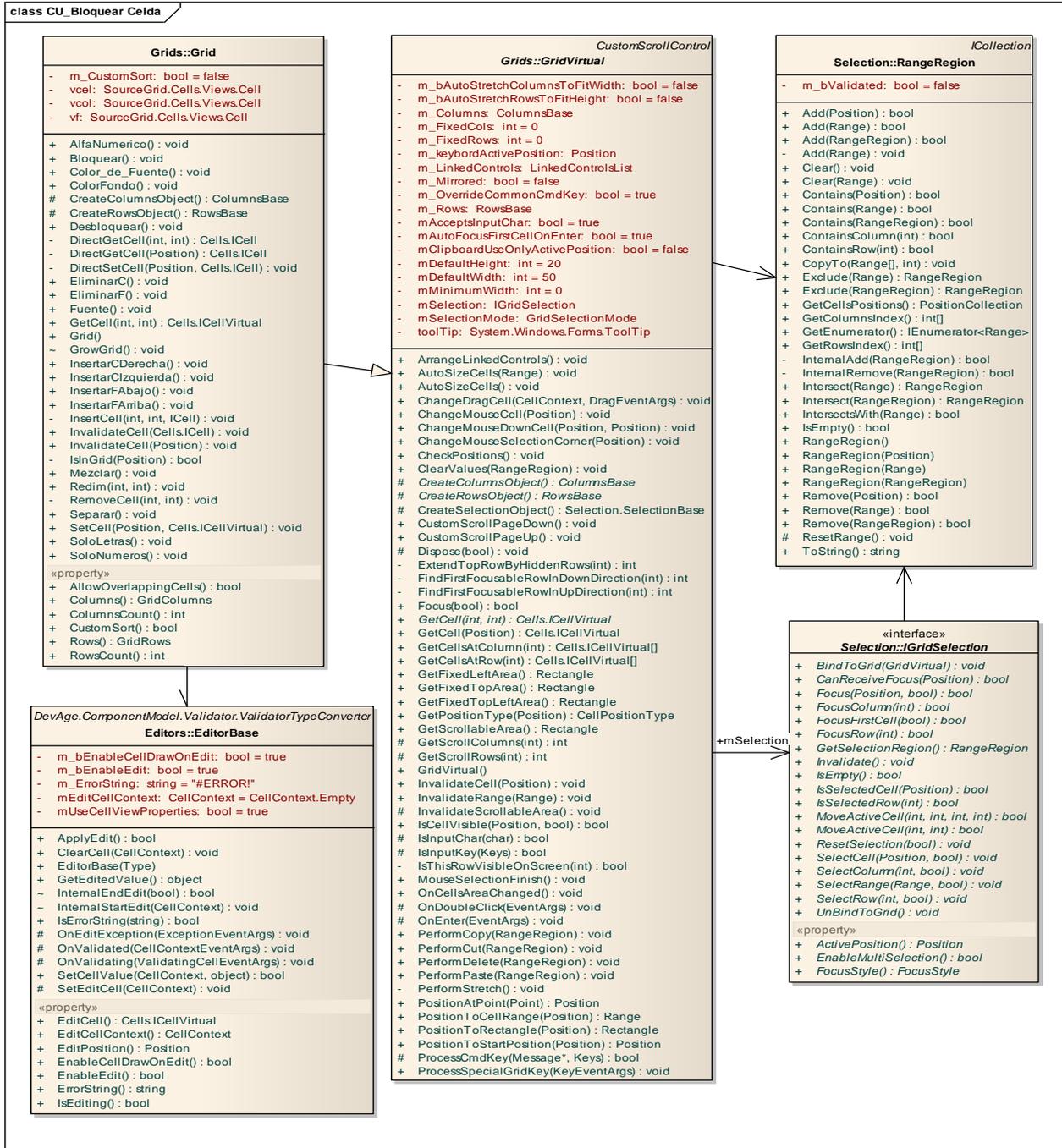


Fig. 11. Diagrama de clases del diseño CUS Bloquear celda

3.4. Diagramas de secuencia

Los diagramas de secuencia muestran los objetos y sus relaciones, incluyen los mensajes que pueden ser enviados entre ellos, además, se observan las interacciones entre objetos ordenadas en una secuencia de tiempo. Seguidamente se muestran los diagramas de secuencia por algunos casos de uso del sistema, los restantes se muestran en el [Anexo 1](#).

3.4.1. CUS Insertar columna – Escenario Insertar columna a la derecha

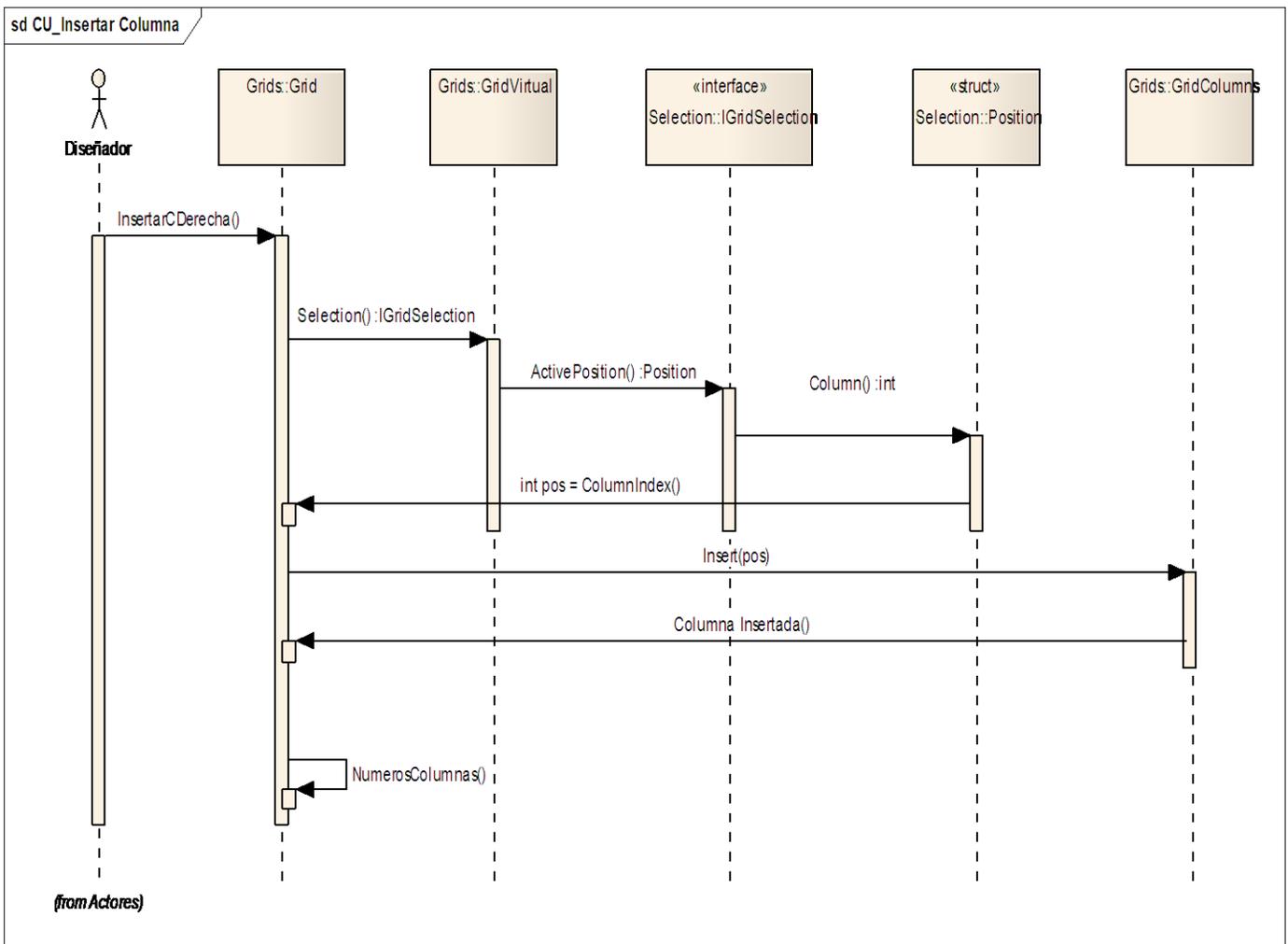


Fig. 62. Diagrama de secuencia CUS Insertar Columna – Escenario Insertar columna a la derecha.

3.4.2. CUS Eliminar columna

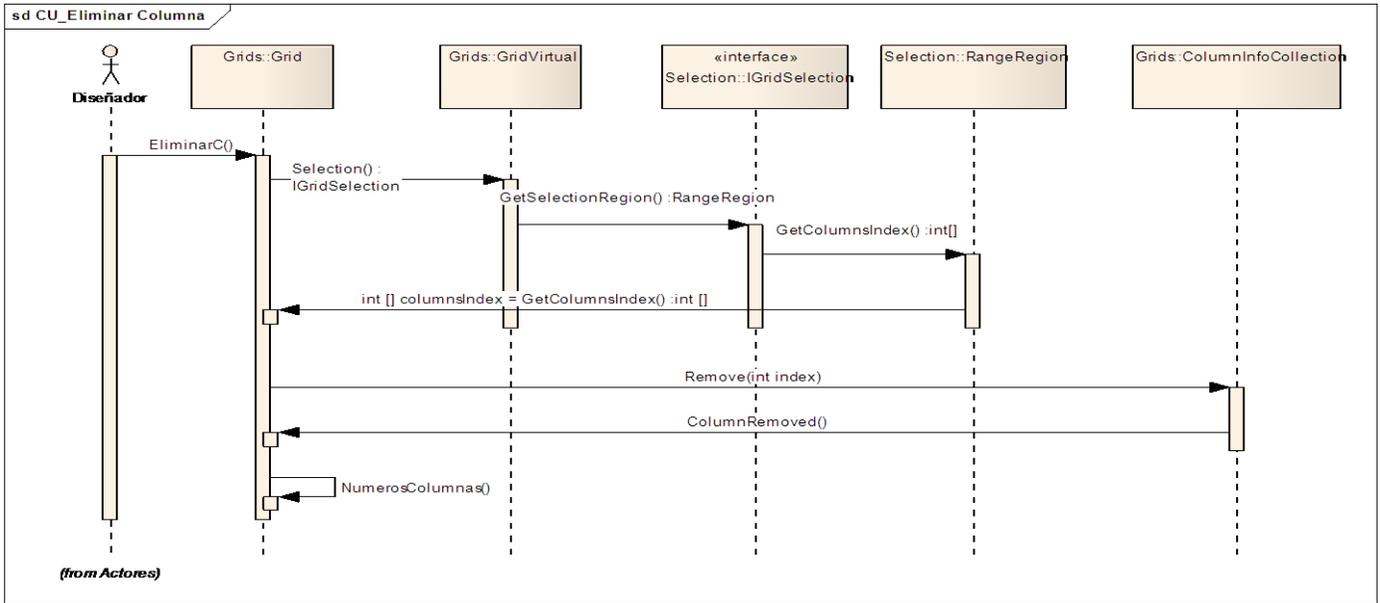


Fig. 73. Diagrama de secuencia CUS Eliminar Columna

3.4.3. CUS Establecer formato – Escenario Establecer fuente

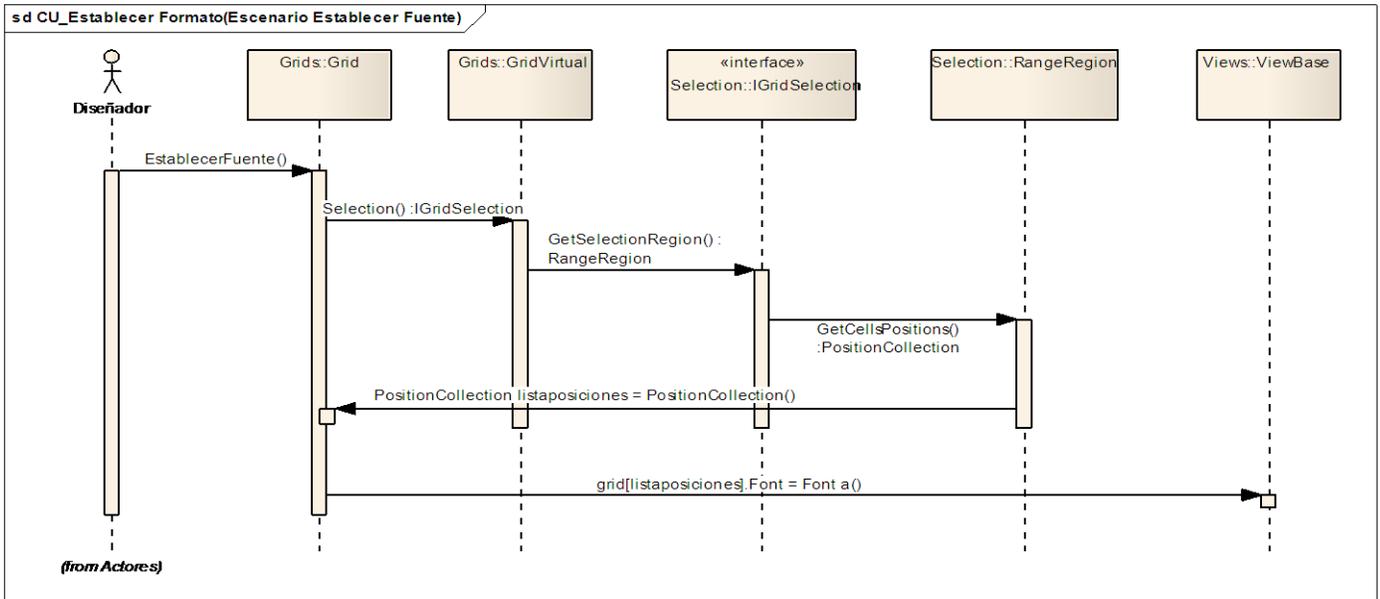


Fig. 84. Diagrama de secuencia CUS Establecer formato – Escenario Establecer fuente.

3.4.4. CUS Exportar

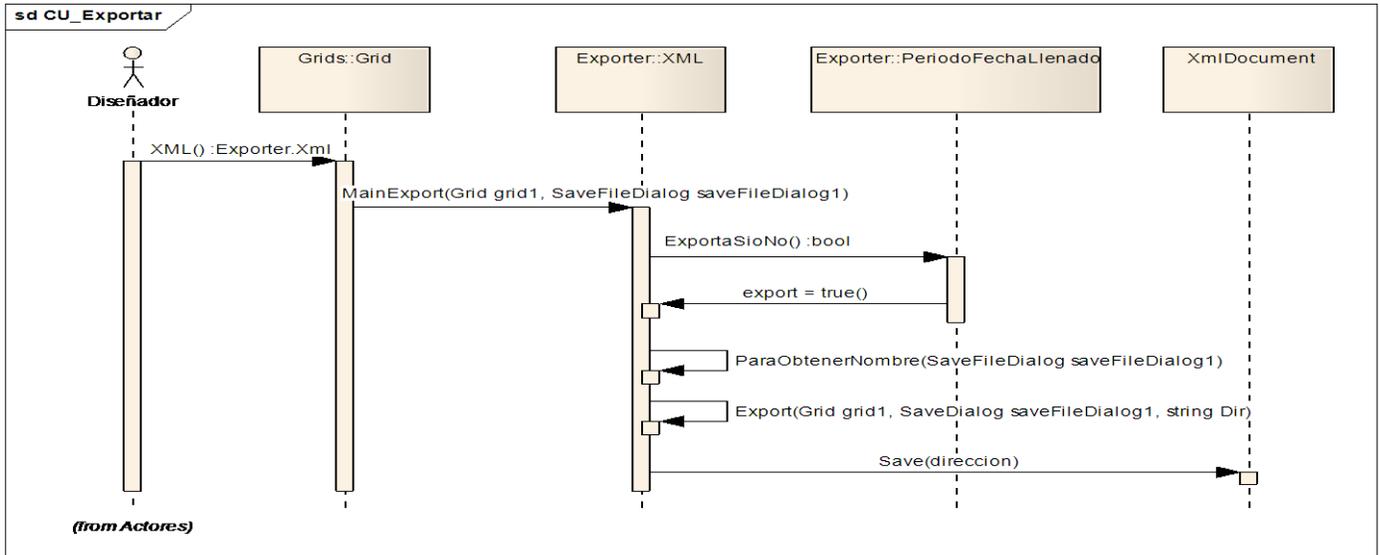


Fig. 95. Diagrama de secuencia CUS Establecer formato

3.5. Descripción de las clases

Modelos

Grid: Representa al Componente Tabular, en esta clase se implementan la mayoría de los métodos que serán utilizados por el Subsistema Editor de Plantillas.

Cell: Es la unidad básica de la clase Grid.

SelectionBase: Representa una celda o rango de celdas que están seleccionadas en el Grid, que permite personalizar las propiedades de esta selección.

GridRows: Maneja la información referente a cada una de las filas.

GridColumn: Maneja la información referente a cada una de las columnas.

Position: Representa la posición de una celda en el grid, para ello guarda la fila y la columna a la que pertenece la celda.

Range: Representa un rango de celdas en el grid. Este rango siempre tiene la posición de inicio en la esquina superior izquierda y la posición final en la esquina inferior derecha.

IValueModel: Una interfaz donde está contenido el valor de cada celda.

Vista

View.Cell: Clase que maneja los aspectos visuales de la celda y puede ser compartida entre muchas de ellas.

Controlador

CellEventDispatcher: Controla cada evento de una celda específica, puede ser compartido entre múltiples celdas y es usualmente el controlador por defecto usado en la clase Grid.

Editor

IEditor: Interfaz usada en operaciones de edición, conversiones de cadenas y formato de datos. Puede ser asignada a la propiedad Editor de la celda.

En el presente capítulo se obtuvo el diagrama de clases del diseño para cada caso de uso del sistema, donde se representaron las relaciones que se establecen entre las clases. Además, quedaron definidos los diagramas de interacción y al modelar el sistema se tuvieron en cuenta todos los requisitos previstos y las restricciones que estos deben cumplir.

Capítulo 4 Implementación

El objetivo principal de la implementación es desarrollar la arquitectura y el sistema en conjunto. En el presente capítulo se pretende describir el modelo de implementación con el propósito de definir la organización del código, planificar las integraciones del sistema necesarias en cada iteración, además de implementar las clases y subsistemas encontrados durante el diseño. Se incluye en el desarrollo del capítulo el diagrama de componentes.

4.1. Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño y las clases se implementan en términos de componentes, ficheros de código fuente, ejecutables, entre otros. Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación, al describir los componentes y construir su organización y dependencia entre los nodos físicos en que funcionará la aplicación.

4.2. Componentes

Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Representa una unidad de código (fuente, binario o ejecutable) que permite mostrar las dependencias en tiempo de compilación y ejecución. Las instancias de componentes de software muestran unidades de software en tiempo de ejecución y generalmente ayudan a identificar sus dependencias y su localización en nodos.

4.3. Diagrama de componentes

Un diagrama de componentes ilustra los fragmentos de software que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clases, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución,

estos constituyen bloques de construcción, así eventualmente un componente puede comprender una gran porción de un sistema.

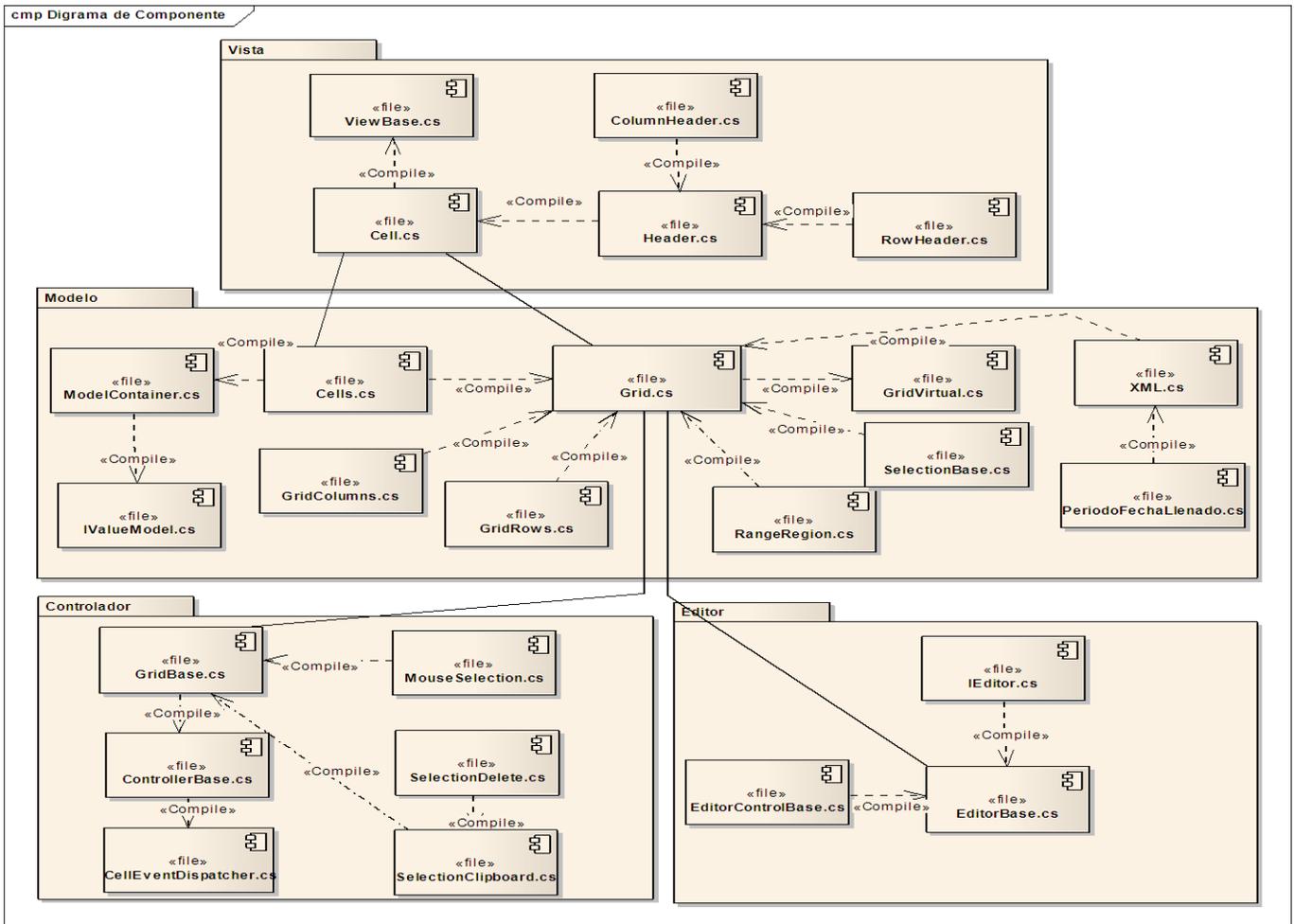


Fig.16. Diagrama de componentes.

En este capítulo se obtuvo el diagrama de componentes, con este se demuestra cómo se encuentra dividido por paquetes de clases el desarrollo del Componente Tabular, agrupando las clases en cuatro grupos fundamentales: modelo, vista, controlador y editor. Se especificaron las relaciones que existen entre los componentes, que indican el grado de interrelación que existe entre ellos.

Conclusiones

Con la investigación se logró desarrollar el Componente Tabular para el Subsistema Editor de Plantillas, dando cumplimiento al objetivo y tareas planteadas. A continuación se plantean los principales resultados:

- El análisis del estado del arte reveló que existen componentes tabulares potentes, pero en su generalidad son herramientas propietarias. Tienen altos precios en el mercado y no satisfacen las necesidades primarias del SEP. Finalmente se utilizó un componente de código abierto para implementar las funcionalidades requeridas por el SEP.
- Después de analizar las posibles herramientas y tecnologías para el desarrollo de la aplicación se decidió implementar el componente basándose en un patrón de arquitectura Modelo Vista Controlador modificado. Se utilizó CSharp como lenguaje de programación y Visual Studio 2008 como IDE (Entorno de Desarrollo Integrado) sobre .NET Framework.
- Durante el desarrollo de la aplicación se realizaron pruebas de compatibilidad con el Framework Mono las cuales resultaron positivas, pues se limitó en la implementación el uso de funcionalidades y componentes que estuvieran relacionados con la plataforma Windows. Esto ayudará a minimizar las dificultades en el momento de efectuar una migración a entornos basados en el sistema operativo GNU/Linux.
- Como resultado del modelado de los flujos de trabajo Modelamiento del negocio, Requerimientos, Análisis y Diseño e Implementación que propone la metodología RUP, se obtuvieron varios artefactos necesarios para obtener una visión detallada del sistema informatizado que permitirá la continuidad en el desarrollo del componente.
- Se implementó un componente que satisface las funcionalidades primarias del SEP y que a la vez ha permitido ampliarlas, con esto se logra cumplir las demandas de los especialistas y sustituir el componente utilizado por el Editor de Plantillas en su versión de prueba.

Recomendaciones

Al equipo de desarrollo del Departamento de Sistemas de Apoyo a la Salud:

- Sustituir el Componente Tabular del Subsistema Captura y Reparó de la Información por el ya implementado.
- Lograr que el Componente Tabular permita importar los distintos modelos de flujo que existen en formato Excel.
- Agregar funcionalidades que brinden más opciones a la hora de diseñar los modelos.

A la Dirección Nacional de Registros Médicos y Estadísticas Sanitarias:

- Mantener comunicación estrecha con el equipo de desarrollo para facilitar el proceso de retroalimentación y mejora de las funcionalidades y características del sistema.

Referencias Bibliográficas

1. GestioPolis. Concepto e historia de la estadística. [Online] [Citado: 16 de Enero 2010.] <http://www.gestipolis.com/recursos/experto/catsexp/pagans/eco/21/estadistica.htm>
2. Lloréns Figueroa, José. La lucha por la Salud Cubana [En línea] [Citado: 26 Enero 2010.] http://books.google.es/books?id=z4tP9h9UT-IC&pg=PA70&lpg=PA70&dq=organizacion+de++las+unidades+basicas+de+salud++en+cuba&source=bl&ots=dalnpDbZ55&sig=dcBPdJuY6dtzwt-gdc34yGopgj8&hl=es&ei=sBEITIOmF4K88gbLtK3AAQ&sa=X&oi=book_result&ct=result&resnum=4&ved=0CCgQ6AEwAw#v=onepage&q&f=false
3. Fernández Cedeño, Karel & Ortega Valenzuela, Trany Rafael. Sistema de Información Estadística Complementario de Salud. Módulo: Actividades de Cirugía y otras Atenciones y Servicios. 2008.
4. DEIS, Departamento de Estadísticas e Información de Salud [online]. [Citado: Febrero 2, 2010.] <http://deis.minsal.cl/index.asp>
5. Sitio de descargas de software. [Online] [Citado: Febrero 15, 2010.] http://www.freedownloadmanager.org/es/downloads/Control_de_Rejilla_de_FlexCell_500_p/
6. Sitio de descarga de Software. [Online] [Citado: Febrero 20, 2010.] http://www.freedownloadmanager.org/es/downloads/search.php?string=Sigma+grid&search=All&match=Any&search_btn=Buscar+%3E%3E%3E.
7. Instituto Nacional de Estadísticas. [Online] [Citado: Marzo 16, 2010.] <http://www.ine.es/prodyser/pcaxis/pcaxis.htm>.
8. Kioskea.net. [Online] [Citado: Marzo 4, 2010.] <http://es.kioskea.net/contents/tableur/tablintro.php3>
9. MySoftTool Studio. [Online] [Citado: Marzo 16, 2010.] <http://mysofttool.com/>.
10. Perez, Yunelis & Rosabal, Antonio. 2009. Sistema para la Gestión y Análisis de la Información Estadística en la Salud Pública Cubana: Subsistema Editor de Plantillas. Ciudad de la Habana : s.n., 2009.
11. Ídem 10
12. Lenguaje de modelamiento unificado. [Online] [Citado: Marzo 10, 2010.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info21/uml.htm>

13. *Introducción al lenguaje C# y .NET Framework*. MSDN. [Online] [Cited: Febrero 10, 2010.] <http://msdn.microsoft.com/es-es/library/z1zx9t92%28VS.80%29.aspx>.
14. *World Wide Web*. [Online] [Cited: Febrero 10, 2010.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>
15. Ramos Carrillo, Anay. *Eumed.net*. [Online] [Cited: Febrero 22, 2010.] <http://www.eumed.net/libros/2009c/587/indice.htm>.
16. *El arte de programar*. [Online] [Cited: Marzo 10, 2010.] <http://www.elartedeprogramar.cl/csharp-developed-visual-studio-2008-monodevelop/>.
17. *Enterprise Architect*. [Online] [Citado: Marzo 30, 2010.] <http://www.sparxsystems.com.ar/>
18. *The Code Project*. [Online] 1999. [Citado: Febrero 12, 2010.] <http://www.codeproject.com/KB/grid/csharpgridcontrol.aspx>
19. *BenchPrincipal*. [Online] [Cited: Marzo 20, 2010.] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.

Bibliografía

1. BenchPrincipal. [Online] [Cited: Marzo 20, 2010.] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.
2. DEIS, Departamento de Estadísticas e Información de Salud [online]. [Citado: Febrero 2, 2010.] <http://deis.minsal.cl/index.asp>
3. El arte de programar. [Online] [Cited: Marzo 10, 2010.] <http://www.elartedeprogramar.cl/csharp-developed-visual-studio-2008-monodevelop/>.
4. Enterprise Architect . [Online] [Citado: Marzo 30, 2010.] <http://www.sparxsystems.com.ar/>
5. Fernández Cedeño, Karel & Ortega Valenzuela, Trany Rafael. Sistema de Información Estadística Complementario de Salud. Módulo: Actividades de Cirugía y otras Atenciones y Servicios. 2008.
6. GestioPolis. Concepto e historia de la estadística. [Online] [Citado: 16 de Enero 2010.] <http://www.gestiopolis.com/recursos/experto/catsexp/pagans/eco/21/estadistica.htm>
7. Guía de Usuario de Enterprise Architect. [En línea] [Citado: 8 de Febrero de 2010.] <http://www.sparxsystems.com.ar/EASUserGuide/ea.html>.
8. Infomed, Portal de Salud de Cuba [Online]. [En línea] [Citado: 10 de Febrero de 2010.] <http://www.sld.cu/sitios/dne/temas.php?idv=8337>.
9. Instituto Nacional de Estadísticas. [Online] [Citado: Marzo 16, 2010.] <http://www.ine.es/prodyser/pcaxis/pcaxis.htm>.
10. Introducción al lenguaje C# y .NET Framework. MSDN. [Online] [Cited: Febrero 10, 2010.] <http://msdn.microsoft.com/es-es/library/z1zx9t92%28VS.80%29.aspx>.
11. Introducción a UML 2.0. [En línea] [Citado: 23 de Enero de 2010.] http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15#Conociendo_UML_2_0.
12. Kioskea.net. [Online] [Citado: Marzo 4, 2010.] <http://es.kioskea.net/contents/tableur/tablintro.php3>
13. LENGUAJE DE MODELAMIENTO UNIFICADO (UML). [En línea] [Citado: 20 de Enero de 2010.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info21/uml.htm>.
14. Lloréns Figueroa, José. La lucha por la Salud Cubana [En línea] [Citado: 26 Enero 2010.] <http://books.google.es/books?id=z4tP9h9UT-IC&pg=PA70&lpg=PA70&dq=organizacion+de++las+unidades+basicas+de+salud++en+cuba&source>

=bl&ots=dalnpDbZ55&sig=dcBPdJuY6dtzwt-

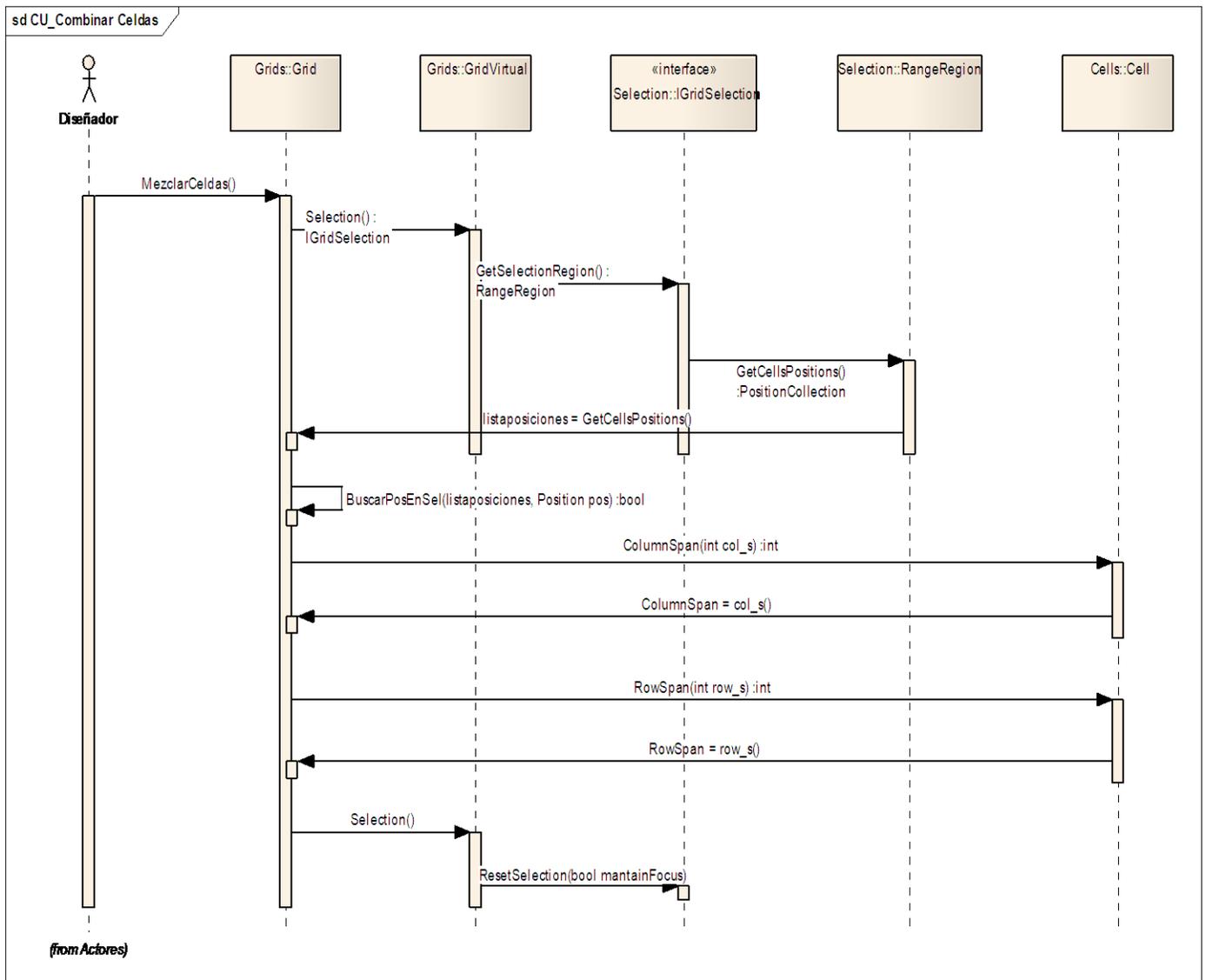
gdc34yGopgj8&hl=es&ei=sBEITIOmF4K88gbLtK3AAQ&sa=X&oi=book_result&ct=result&resnum=4&ved=0CCgQ6AEwAw#v=onepage&q&f=false

15. Modelo de Dominio. [En línea] [Citado: 5 de Febrero de 2010.] http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
16. MySoftTool Studio. [Online] [Citado: Marzo 16, 2010.] <http://mysofttool.com/>
17. Perez, Yunelis & Rosabal, Antonio. 2009. Sistema para la Gestión y Análisis de la Información Estadística en la Salud Pública Cubana: Subsistema Editor de Plantillas. Ciudad de la Habana : s.n., 2009.
18. Ramos Carrillo, Anay. Eumed.net. [Online] [Cited: Febrero 22, 2010.] <http://www.eumed.net/libros/2009c/587/indice.htm>.
19. Servicios Web en C# .Net. [En línea] [Citado: 1 de Febrero de 2010.] <http://www.locualo.net/programacion/servicios-web-net/00000025.aspx>.
20. Sitio de descarga del Software. [Online] [Citado: Febrero 18, 2010.] <http://www.freedownloadmanager.org/es/downloads/search.php?search=All&string=grid&match=Any&startpage=1>.
21. Sitio de descargas de software. [Online] [Citado: Febrero 15, 2010.] http://www.freedownloadmanager.org/es/downloads/Control_de_Rejilla_de_FlexCell_500_p/
22. Sitio de descarga de Software. [Online] [Citado: Febrero 20, 2010.] http://www.freedownloadmanager.org/es/downloads/search.php?string=Sigma+grid&search=All&match=Any&search_btn=Buscar+%3E%3E%3E.
23. Sitio de la Juventud Cubana en saludo al Aniversario 50 de la Revolución. [En línea] [Citado: 15 de febrero de 2010.] http://aniversario50.cubava.cu/index.php?option=com_content&view=article&id=198%3AAsalud&catid=59%3Ajusticia-social&Itemid=56&lang=es.
24. The Code Project. [Online] 1999. [Citado: Febrero 12, 2010.] <http://www.codeproject.com/KB/grid/csharpgridcontrol.aspx>
25. World Wide Web. [Online] [Cited: Febrero 10, 2010.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>

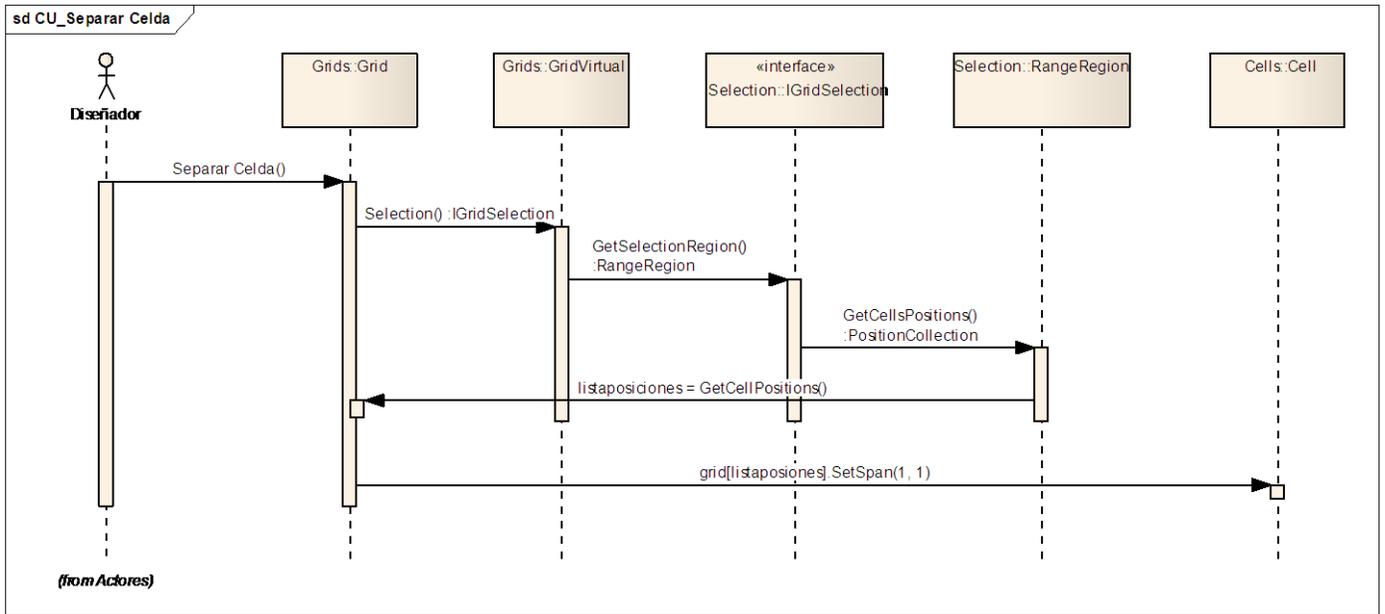
Anexos

Anexo 1 Diagramas de secuencia

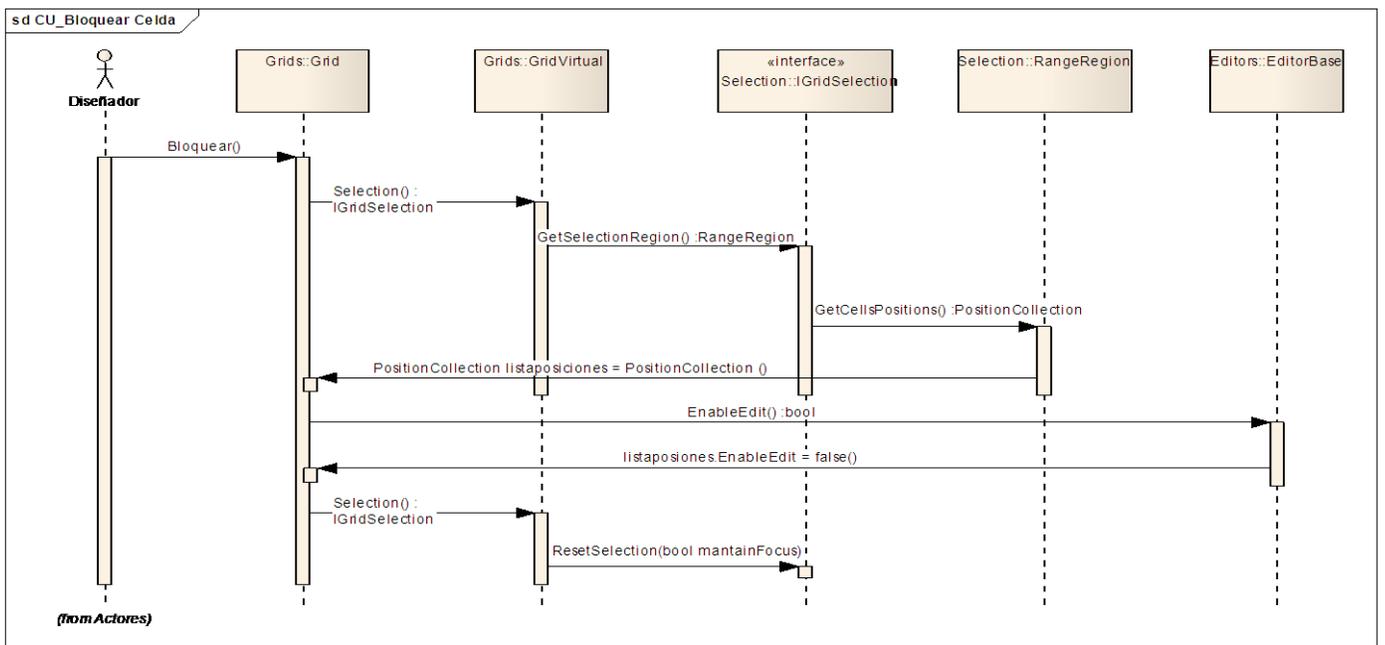
CUS Combinar Celdas



CUS Separar Celdas



CUS Bloquear celdas



Glosario de Términos

Actor: Cualquier individuo, grupo, entidad, organización, máquina o sistema de información externo.

Caso de Uso: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias. [RUP]

Clases: Conjunto de objetos que comparten atributos, operaciones, relaciones y semántica, las mismas representan los conceptos fundamentales del sistema.

Diagramas: Es la representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.

Diagrama de clases del modelo del dominio: Representación de los conceptos de importancia en el área de la aplicación, así como de las relaciones entre estos.

Editor de Plantillas: Se encarga de gestionar los modelos de flujo.

Enterprise Architect: Enterprise Architect es una herramienta CASE (Computer Aided Software Engineering) para el diseño y construcción de sistemas de software. EA soporta la especificación de UML 2.0, que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto.

EA es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, y proporciona una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio. Puede modelar procesos de negocio, sitios web, interfaces de usuario, redes, configuraciones de hardware, mensajes y más. Tiene las características precisas para diseñar y administrar el desarrollo e implementación de un sistema.

Plantilla o Modelo de flujo: Modelo o planilla en la cual se encuentran los indicadores a captar por parte del sistema, cada uno se corresponde con un subsistema de información de los que conforman el Sistema de Información Estadística Complementaria de Salud. Estos modelos son creados y modificados por parte de la Dirección Nacional de Registros Médicos y Estadística Sanitaria en dependencia de las

necesidades de información del propio Ministerio de Salud o de otras instituciones que utilizan esta información con el objetivo de trazar estrategias y evaluar procedimientos.

Prototipo: Maqueta visual funcional o no de la futura aplicación. Este puede ser una imagen o una aplicación que simule funcionalidades del software.

RF (requisitos funcionales): Los Requisitos Funcionales son capacidades o condiciones que el sistema debe cumplir.

RFN (requisitos no funcionales): Los Requisitos No Funcionales son propiedades o cualidades que el sistema debe tener. Normalmente, están vinculados a requisitos funcionales, es decir, una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.