

Universidad de las Ciencias Informáticas

Facultad 6



**Interfaz Web de Captura de Datos
en Dispositivos Móviles en el
Sistema Ligero de Captura de Datos.**

**Trabajo de Diploma para optar por el título de
Ingeniero Informático**

Autor(es): "Juliet Mora Casares"

"Lorenzo Martinez Mena"

Tutor(es): "Ing. Frank González Fernández"

Co-tutor: "Ing. Dioleisys Fontela González"

"La Habana, Junio 2012"

"Año 54 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 firmo la presente a los ____ días del mes de _____ del año _____.

<nombre autor>

<nombre tutor>

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Frank González Fernández
Ingeniero en Ciencias Informáticas
E-mail: ffernandez@uci.cu

Tutor: Ing. Dioletsys Fontela Gonzáles
Ingeniero en Ciencias Informáticas
E-mail: dfontela@uci.cu

Agradecimientos

AGRADECIMIENTOS

Muchas han sido las personas que han contribuido al desarrollo de esta investigación, agradecemos a todos los profesionales que desde nuestros inicios en la carrera han trabajado intensamente en nuestra formación, a los docentes por su instrucción y educación adecuada y con profunda científicidad, a los tutores, que desde los cimientos de la tesis nos orientaron adecuadamente para el logro eficiente de la presente obra que hoy presentamos, con exigencia y constancia en toda la dirección asumida aportando calidad a nuestra formación.

Al colectivo en general de la escuela, que ha incidido de manera directa o indirecta en nuestra realización como profesional de las ciencias informáticas.

No podemos obviar a la familia, que nos ha dado su apoyo espiritual y económico para nuestro adecuado desenvolvimiento en este largo camino de constante estudio y dedicación.

Lorenzo.

Agradezco a mis padres porque sin ellos no estaría aquí, porque han luchado por mí y por mi superación y por estar siempre presentes. A mi abuela, mi tío, y mi hermano por quererme, apoyarme y confiar en mí. A mi abuelo por incentivarme desde pequeña a luchar por mi futuro y al que le hubiera encantado ver este sueño hecho realidad. A mis tutores por soportarme y por todas las peleas que permitieron el éxito de este trabajo. Al profesor Lobo por haberme dado la luz cuando todo era oscuridad y permitir que éste trabajo se haya llevado a cabo. Al profesor Garnache por ser otro tutor más para nosotros y no solo para nosotros sino para todos los muchachos del centro. A mis amigas Ana Rosa, Marla, Margarita que han estado estos 5 años a mi lado, y en especial a mi inseparable amiga Claudia por tantos ratos divertidos y por dejarme dormir en los turnos de clases. A todo mi piquete, mis amigos del café, por tantas noches en vela, tantas maldades, por su apoyo y por haber hecho este último año diferente. A mi compañera de fiesta Mary sin la cual no hubiera tenido ánimos cada mañana para despertarme a trabajar. A aquellos profesores que de verdad me enseñaron e hicieron por mi superación. En fin a todos aquellos que siempre me han querido y me han apoyado.

Juliet.

Dedicatoria

DEDICATORIA

Dedico esta obra a todas las personas que me ayudaron para la realización de la presente investigación y que me apoyaron en cada decisión tomada, en especial:

A mi madre, que ha compartido conmigo desde los primeros latidos de mi corazón, con extraordinaria y encantadora dulzura siempre ha sido mi guía, ejemplo de persona, la cual se mantiene a mi lado en todos los momentos de mi vida, dando todo de sí para hacerme feliz.

A mis Tutores, que con paciencia y dedicación orientaron adecuadamente toda la evolución de la investigación, desarrollando y profundizando nuestros conocimientos y habilidades en el campo de las ciencias científicas.

A mis familiares por su apoyo incondicional durante estos cinco años de intenso estudio.

A todos los profesores, compañeros de estudios y profesionales en general que durante la carrera han influido en mi crecimiento profesional.

A todos aquellos que directa o indirectamente han sido partícipe de mi realización personal. Muchas Gracias.

Lorenzo.

A mi mamá por estar en todo momento para mí, por su apoyo y amor incondicional, por todo su esfuerzo, por hacerme una persona de bien, y por ser yo antes que ella. A mi familia, quienes una que otra noche soñaron mis anhelos, aquí se los dedico convertidos en realidad.

Juliet.

Resumen

RESUMEN

Después de un análisis exhaustivo del Sistema Integrado de Gestión Estadística se presentan diferentes factores que traen como consecuencia la orientación del sistema a dispositivos móviles. Esto se debe a que con el avance de las Nuevas Tecnologías de la Informática y las Comunicaciones (NTICs) el desarrollo de la Web tiende a ser cada vez más ligero y portable, elemento que sumado a factores como la conectividad, traen consigo el diseño de aplicaciones más ligeras para entornos cada vez más cambiantes. Debido a las dificultades anteriormente mencionadas, el Centro de Tecnologías de Gestión de Datos (DATEC) de la Universidad de las Ciencias Informáticas (UCI), se ha dado a la tarea de desarrollar un sistema para la captura de datos en dispositivos móviles. El presente trabajo centra su desarrollo en una interfaz web de captura de datos en dispositivos móviles en el Sistema Ligero de Captura de Datos (SIGELite). El sistema implementado permite realizar las siguientes funcionalidades: exportar, importar, buscar y eliminar plantillas de formulario, así como también eliminar, modificar, exportar e importar recursos (formularios). Del lado del cliente es utilizado el lenguaje de programación J2ME. Para garantizar la portabilidad de la información se desarrolla la aplicación bajo un ambiente multiplataforma, fundamentalmente con herramientas de código abierto.

PALABRAS CLAVES

Sistema Integrado de Gestión Estadística, dispositivos móviles, portable, formularios.

Índice

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
Introducción al Capítulo 1	5
1.1 Tecnología móvil.....	5
1.2 Comunicaciones Inalámbricas	6
1.3 Sistemas de Captura de Datos	6
1.3.1 Sistemas de Captura de Datos para la Web.....	7
1.3.2 Sistemas de Captura de Datos en Dispositivos Móviles	7
1.4 Metodologías de desarrollo de software.....	8
1.4.1 Proceso Unificado Abierto (OpenUP)	9
1.5 Lenguaje Modelado UML (Unified Modeling Language, UML 2.0).....	10
1.6 Herramientas CASE para la modelación del sistema	10
1.6.1 Visual Paradigm 8.0	10
1.7 Lenguajes de Programación	11
1.7.1 Lenguaje de programación del lado del servidor PHP	11
1.7.2 Lenguaje de programación del lado del cliente J2ME.....	11
1.9 Entorno de Desarrollo Integrado (IDE)	13
1.9.1 NetBeans 6.9	13
Conclusiones del Capítulo 1	14

Índice

CAPÍTULO 2: ANÁLISIS Y DISEÑO	15
Introducción al Capítulo 2	15
2.1 Descripción del sistema	15
2.2 Modelo de Dominio	15
2.3 Diagrama conceptual del modelo de dominio.....	16
2.4 Definición de clases del modelo de dominio.....	16
2.5 Especificación de los Requisitos del Sistema.....	17
2.5.1 Requerimientos Funcionales	17
2.5.2 Requerimientos no Funcionales	18
2.6 Modelo de Caso de Uso del Sistema	20
2.6.1 Definición de los actores del sistema	20
2.6.2 Diagrama de Caso de Uso del sistema	21
2.6.3 Descripción textual de los Casos de Uso del sistema.....	21
2.7 Diseño del Sistema	26
2.8 Modelo del Diseño	26
2.9 Diagramas de clases del diseño.....	27
2.10 Patrones de Diseño.....	27
2.11 Diagramas de interacción del diseño.....	29
2.12 Diagrama de Despliegue.....	30
Conclusiones del Capítulo 2	31
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA	32
Introducción al Capítulo 3	32
4.1 Modelo de implementación	32
4.1.1 Diagrama de Componentes.....	32

Índice

4.2 Código Fuente	33
4.2.1 Estándares de codificación.....	33
4.3 Pruebas	35
4.3.1 Pruebas unitarias	35
4.3.2 Prueba de aceptación	38
4.3.3 Prueba de sistema	39
4.4 Diseño y ejecución de los Casos de Prueba de Caja Negra.....	39
4.4.1 Caso de Prueba Administrar Plantilla de Formulario	39
4.4.2 Caso de Prueba Transferir Recurso	42
4.5 Diseño y ejecución del método Caja Blanca	43
4.5.1 Técnica del camino básico del método de Caja Blanca para el método Conectar	44
4.5.2 Técnica del camino básico del método de Caja Blanca para el método Importar Plantilla ...	46
4.6 Interfaces Principales de la Aplicación	48
Conclusiones del Capítulo 3	49
CONCLUSIONES	50
RECOMENDACIONES	51
BIBLIOGRAFÍA	52
REFERENCIAS BIBLIOGRÁFICAS	54
GLOSARIO DE TÉRMINOS	56
ANEXOS	59
ÍNDICE DE FIGURAS	
Figura 1. Componentes de la tecnología Java ME	13
Figura 2: Diagrama del Modelo de Domino.....	16

Índice

Figura 3. Diagrama de Casos de Uso del Sistema.....	21
Figura 4. Diagrama de clases del diseño del CU Administrar Recurso.	27
Figura 5. Diagrama de secuencia del CU Administrar Recurso sección Modificar Recurso.	30
Figura 6. Diagrama de Despliegue.....	30
Figura 7. Diagrama de Componentes.	33
Figura 8. Ejemplo del código fuente de la aplicación.	34
Figura 9. Aplicación de la Técnica del Camino Básico al método Conectar.	45
Figura 10. Aplicación de la Técnica del Camino Básico al método Importar Plantilla.	47
Figura 11. Interfaz Listar Plantilla.....	48
Figura 12. Interfaz Listar Recurso.....	49

ÍNDICE DE TABLAS

Tabla 1. Actores del componente propuesto.....	21
Tabla 2. Descripción textual del caso de uso Administrar Recurso.	26
Tabla 3. Descripción de las variables del caso de prueba Administrar Plantilla de Formulario.....	39
Tabla 4. Sección a probar Buscar Plantilla de Formulario.....	40
Tabla 5. Sección a probar Eliminar Plantilla de Formulario.	41
Tabla 6. Sección a probar Listar Plantilla de Formulario.	42
Tabla 7. Sección a probar Importar recurso.....	42
Tabla 8. Sección a probar Exportar recurso.....	43

Introducción

INTRODUCCIÓN

La humanidad hoy en día se encuentra indisolublemente ligada al desarrollo vertiginoso de las tecnologías y las comunicaciones, elementos que se mezclan para convertirse en partes indispensables de las actividades cotidianas. En las comunicaciones, la telefonía móvil ha sido una de las tecnologías que ha evolucionado en la última década por las facilidades que brinda.

Con el incremento en los últimos años de la telefonía móvil se ha fomentado el intercambio de información a partir de las necesidades cada vez más crecientes de los usuarios, las cuales como respuesta han devenido en diferentes dispositivos, tales como: Asistente Digital Personal (PDA), cámaras de fotos, agendas telefónicas, calculadoras, Sistemas de Posicionamiento Global (GPS), relojes, entre otros. Las funcionalidades anteriores traen como consecuencia la sustitución gradual de las líneas fijas tradicionales por las líneas móviles, paralelamente a esto, los productores de terminales van incorporando a los celulares tecnologías relacionadas con la transmisión de datos, las cuales posibilitan el uso de celulares para brindar nuevos servicios de captura de información, proporcionándole al usuario el acceso a disímiles sitios web sin necesidad de utilizar una computadora de escritorio o portátil como intermediaria.

Las nuevas Tecnologías de la Información y las Comunicaciones (TICs) han significado una verdadera revolución en el ámbito social. Cuba no se encuentra ajena a estos avances tecnológicos e independientemente de que los mismos no pueden llegar a todas partes de su territorio, se presta un interés especial al desarrollo de la informática y las comunicaciones.

La Universidad de las Ciencias Informáticas (UCI), es un centro docente-productivo que tiene entre sus objetivos fundamentales informatizar el país y desarrollar la industria cubana del software. Dicha universidad cuenta con una red de centros productivos entre los cuales se encuentra el Centro de Tecnologías de Gestión de Datos (DATEC). En este centro existen diferentes Líneas de Productos de Software (LPS), entre las que se halla la línea Integración de Soluciones, que tiene entre sus principales proyectos; el Sistema Integrado de Gestión Estadística (SIGE) cuyo principal objetivo es capturar la información estadística del país a nivel empresarial.

La estadística en Cuba es organizada, propuesta y ejecutada por la Oficina Nacional de Estadísticas e Información(ONEI), entidad adscrita al Ministerio de Economía y Planificación de la República de Cuba, dado el carácter esencialmente central de la administración del Estado, toda empresa, organismo, organización o institución identificado como centro informante y poseedor de información de interés para la toma de decisiones de los Órganos de la Administración Central del Estado, se

Introducción

encuentra en la obligación de reportar información estadística bajo convenio con la ONEI. Para cumplir con su misión la ONEI ha priorizado a partir del 2006 la introducción de las más novedosas técnicas de almacenamiento y procesamiento de datos. En este contexto, en octubre de ese mismo año, se inició en conjunto con la UCI el desarrollo de una aplicación de soporte al marco metodológico del trabajo estadístico. Surge así el Proyecto de Informatización de la ONEI que desarrolla el SIGE.

A raíz de un análisis exhaustivo en el SIGE se deduce que existen diferentes factores que repercuten en el correcto funcionamiento del mismo. Debido al avance de las Nuevas Tecnologías de la Informática y las Comunicaciones (NTICs) en la web el desarrollo tiende a ser cada vez más ligero y portable, paralelo a esto existen agentes externos que dificultan el funcionamiento del sistema por cuestiones propias de la pobre infraestructura tecnológica existente en el país. En Cuba hoy no existe una infraestructura adecuada para implantar SIGE a todos los niveles: municipal, provincial y nacional. Todo esto, sumado al volumen de información a captar, que a grandes rasgos constituye la información estadística de todas las empresas del país, hace que el sistema implantado presente deficiencias en aquellos entornos que tengan como característica la situación anterior.

Otro factor a tener en cuenta, son los recursos de hardware que se necesitan para su instalación y despliegue, los cuales no pueden ser cubiertos en su totalidad debido a la gran cantidad de actividades informáticas que se realizan en esta entidad y las adscritas a ella. En este sentido se pueden especificar: servicios de correo, Protocolo de Transferencia de Archivos (FTP, por sus siglas en inglés) y clientes ligeros, los que en general demandan servidores para su puesta en marcha. A esto se suma la necesidad de equipos de última generación para suplir las necesidades cada vez más crecientes del sistema debido a su constante desarrollo.

Durante la realización de las encuestas se presentan errores humanos debido a que la persona que registra la información en las plantillas de encuesta no es la misma que la digita en SIGE. Otras insuficiencias pueden venir acarreadas por el deterioro de los materiales ofimáticos involucrados en el proceso de captura de información, lo que influye en la calidad de la misma así como su posible pérdida. Todo este proceso, afecta la toma de decisiones de los especialistas en relación con las actividades de planificación de los recursos en un período de tiempo.

A partir de esta situación problemática surge el siguiente **Problema de la Investigación**: ¿Cómo contribuir a la captura de la información en el Sistema Ligero de Captura de Datos?

Introducción

La investigación tiene como **objeto de estudio** el desarrollo de aplicaciones para dispositivos móviles y como **campo de acción** el desarrollo de interfaces web para la captura de datos en dispositivos móviles.

Para darle solución al problema planteado se define como **objetivo general** desarrollar una interfaz web para dispositivos móviles que permita la captura de datos a partir de los formularios definidos en el Sistema Ligero de Captura de Datos (SIGELITE).

Del objetivo general se desglosan los siguientes **objetivos específicos**:

- Analizar el marco conceptual relacionado con las interfaces web de captura de datos en dispositivos móviles.
- Diseñar la interfaz web de captura de datos en dispositivos móviles.
- Implementar la interfaz web de captura de datos en dispositivos móviles.
- Probar el funcionamiento de la interfaz web de captura de datos en dispositivos móviles.

Para cumplir con el objetivo propuesto y resolver la situación problemática mencionada, se plantean las siguientes **tareas de la investigación**:

- Revisión bibliográfica de las herramientas de captura de datos en dispositivos móviles.
- Definición de las herramientas y metodología a utilizar.
- Realización del modelo de dominio para la interfaz web de captura de datos en dispositivos móviles.
- Definición de los requisitos funcionales y no funcionales.
- Descripción de los casos de uso del sistema.
- Análisis de la interfaz web de captura de datos en dispositivos móviles.
- Diseño de la interfaz web de captura de datos en dispositivos móviles.
- Implementación de la interfaz web de captura de datos en dispositivos móviles.
- Definición de los casos de prueba.
- Ejecución de las pruebas definidas para la interfaz web de captura de datos en dispositivos móviles.

Introducción

Posibles resultados:

- Obtención de los artefactos del Análisis, Diseño, Implementación y Prueba de la Interfaz Web de captura de datos en dispositivos móviles.
- Interfaz Web de captura de datos en dispositivos móviles a partir de los formularios generados en SIGE.

El presente trabajo consta de cuatro capítulos en los cuales se describe todo el proceso por el que transita la investigación.

Capítulo 1 Fundamentación Teórica: En este capítulo se presenta la definición del marco teórico de la investigación, se exponen los temas referentes a los sistemas de captura de datos en dispositivos móviles y se define la metodología, herramientas y tecnologías a utilizar para el desarrollo de la interfaz web de captura de datos en dispositivos móviles.

Capítulo 2 Análisis y Diseño: En este capítulo se describe detalladamente las características del sistema a desarrollar. Se especifica el Modelo de Dominio como vía para describir los principales conceptos del negocio. Son especificados los requisitos funcionales y no funcionales, así como también el diagrama de casos de uso del sistema y las descripciones de los mismos. Se detallan los diagramas que representan las funcionalidades del sistema. Se modelan y detallan los diagramas que representan las funcionalidades del sistema, aplicando los patrones de diseño seleccionados.

Capítulo 3 Implementación y Prueba: En este capítulo se describe como fue implementado el componente, así como las garantías de su funcionamiento de acuerdo con lo previsto.

Capítulo 1: Fundamentación Teórica

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción al Capítulo 1

En este capítulo se enuncian los principales conceptos teóricos que constituyen la base de la investigación. Se abordan los temas relacionados con la captura de datos. Se exponen las herramientas y lenguajes de programación a utilizar para el desarrollo de la interfaz web de captura de datos en dispositivos móviles.

1.1 Tecnología móvil

Las tecnologías inalámbricas han experimentado un vertiginoso desarrollo en los últimos años, una de las más significativas es la tecnología móvil; que a grandes rasgos es un sistema que permite al usuario establecer y recibir una comunicación de voz o datos dentro de la zona de cobertura de la estación base (conjunto de antenas receptoras y transmisoras) con la que interactúa el móvil una vez encendido. A pesar de que fue concebida inicialmente para transmitir señales de voz debido a las limitaciones tecnológicas de la época, la tecnología móvil de hoy en día es capaz de brindar otros tipos de servicios tales como almacenamiento de datos, reproducción de audio y video, navegación en Internet, realización de compras online y envío de mensajes electrónicos, entre otros servicios.

Las comunicaciones entre móviles se generan a bajas potencias lo que trae como consecuencia que el móvil deba estar cerca de una estación base para captar correctamente la señal. Por esta razón, para comunicaciones entre teléfonos móviles que abarquen grandes distancias, debe existir una red con muchas estaciones base para que cada una de ellas brinde una cobertura de radio sobre pequeñas áreas geográficas comúnmente conocidas como zonas de cobertura, "celda o célula" (de ahí la expresión teléfonos celulares) donde la cobertura de cada celda debe comunicarse parcialmente con su vecina para asegurar que no haya ninguna interrupción en la comunicación.

Los teléfonos celulares poseen transmisores de bajo poder. Muchos de ellos presentan dos intensidades de señal: 0.6 watts y 3.0 watts (en comparación con la mayoría de los radios de banda civil transmiten a 4 watts). Paralelamente, las estaciones centrales también transmiten a bajo poder, lo que se materializa en dos ventajas fundamentales:

- Las transmisiones de la estación central y de los teléfonos en la misma zona de cobertura no salen de ésta, lo que permite la reutilización de las frecuencias asociadas a la zona.

Capítulo 1: *Fundamentación Teórica*

- El consumo de energía del teléfono celular, que generalmente funciona con baterías, es relativamente bajo. Una baja energía significa baterías más pequeñas, lo cual hace posible que los teléfonos celulares sean más pequeños.(HIDALGO, 2008)

1.2 Comunicaciones Inalámbricas

Uno de los mayores logros de la computación lo trajo el desarrollo de Ethernet en los años 70, una red capaz de conectar varias máquinas en una red local de forma eficiente. Esta idea fue creciendo, mejores métodos de transmisión aparecieron, aunque todos ellos limitados por la necesidad de una conexión física (cable) para poder comunicarse entre distintos dispositivos. Una nueva forma de transmisión era necesaria, una que eliminara los problemas que acarreaban una reestructuración de un local de trabajo o una aplicación del mismo. Al igual que el telégrafo evolucionó con el devenir del tiempo a teléfonos móviles los cuales eliminaron el cableado tradicional, las redes locales necesitaban un proceso similar, de aquí que surgiera la idea de crear redes locales inalámbricas, WLAN (del inglés: Wireless Local Area Network) partiendo de que el concepto de eliminar la conexión física no era nuevo sino que databa de principios del siglo pasado. Las redes inalámbricas traen consigo un conjunto considerable de ventajas tales como:

- Disponibilidad de la información, pues un trabajador puede acceder a la misma sin necesidad de trasladarse de lugar.
- Facilidad de ampliación de las redes locales existentes, debido a que en la mayoría de los casos resulta más económico que rediseñar toda la estructura, reduciendo considerablemente tanto la inversión económica como el tiempo de reestructuración.
- Comodidad, facilidad de instalación y mantenimiento.(HIDALGO, 2008)

1.3 Sistemas de Captura de Datos

En Cuba, a pesar de las difíciles condiciones económicas se han realizado grandes esfuerzos para ir a la par del desarrollo de la telefonía celular en el mundo y la diversidad de los servicios disponibles en esta tecnología. En el país existe desde hace algún tiempo como proveedor de servicios Cubacel, sucursal de la Empresa de Telecomunicaciones de Cuba (ETECSA), la cual en materia de desarrollo de aplicaciones de captura de información no ha tenido grandes resultados.

Capítulo 1: Fundamentación Teórica

1.3.1 Sistemas de Captura de Datos para la Web

Sistema Integrado de Gestión Estadística

El Sistema Integrado de Gestión Estadística (SIGE) fue desarrollado de conjunto con la ONEI de forma tal que contempla el marco metodológico y operativo de esta. Como sistema está integrado por módulos altamente especializados que permiten la gestión de las unidades de observación y la captación de la información estadística. Este sistema está compuesto por los siguientes módulos: Gestión de Configuración, Diseñador de Formularios, Entrada de Datos y Generador de Reportes, este último componente es externo al sistema y permite la visualización de la información almacenada. SIGE tiene como objetivo principal captar la información estadística a nivel empresarial a partir de formularios matriciales. Los formularios matriciales se definen a partir de Indicadores (Herramientas para clasificar y definir, de forma más precisa, objetivos e impactos. Constituyen medidas verificables de cambio o resultado diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a las metas establecidas) y Aspectos (elementos relacionados a los indicadores que enmarcan la dimensión de estos, entiéndase características más específicas). Otro mecanismo para la obtención de la información lo constituyen las encuestas, las cuales son usadas para la captación de información frecuentemente con diferentes tipos de información. SIGE como política se orienta a estándares abiertos lo cual concuerda con la política del país hacia una soberanía tecnológica. (DELGADO, 2011)

Sistema Ligero de Captura de Datos

El Sistema Ligero de Captura de Datos (SIGELITE), es una aplicación que tiene como finalidad la captura de datos a partir de plantillas definidas en formato XML por SIGE. SIGELITE permite la gestión de plantillas de formularios y encuestas, así como también de recursos (formularios y encuestas), los cuales no se guardan en una base de datos, sino que se almacenan en el servidor en archivos físicos en formato XML.

1.3.2 Sistemas de Captura de Datos en Dispositivos Móviles

SysSurvey

SysSurvey es una solución especializada en soportar procesos estratégicos para la creación y desarrollo de formularios en la captura de datos por medio de dispositivos móviles o computadores de mano tipo Pocket o Palm. La aplicación SysSurvey es una herramienta que permite realizar la captura

Capítulo 1: Fundamentación Teórica

de datos censales a través de un Dispositivo Móvil de Captura (DMC), permitiendo al encuestador ubicarse geográficamente en terreno para llevar a cabo la captura de datos.(MOLINA, 2005)

QBSys

Este sistema es un módulo de captura de datos para dispositivos móviles que permite la automatización de la tarea de relevamiento de estados, captura de sucesos, alertas sobre acciones a tomar y documentación de seguimiento de tareas. El objetivo es el manejo de dicha información para la obtención de reportes y emisión de avisos automáticos ante determinadas anomalías detectadas en los puntos de control.

El módulo se basa en un sistema móvil que permite la recolección de datos mientras se ejecuta un recorrido operativo. Estos datos serán cargados de forma automática en una base de datos y una aplicación permitirá finalmente la utilización de la información recolectada y la implementación de alarmas para la visualización de situaciones de excepción.(ROSARIO, 2000)

Después de analizar las herramientas anteriores se concluye que las mismas, aunque han logrado un producto de calidad con disímiles facilidades de uso y seguridad, no se ajustan a las necesidades que conllevaron a la realización del presente trabajo, debido a que las mismas presentan una licencia comercial lo cual haría de obligatorio cumplimiento la compra de la solución bajo normas detalladas y condiciones de uso, y el lenguaje de programación utilizado no es compatible con las herramientas del SIGELITE. También es válido recordar que las soluciones desarrolladas a nivel internacional son de propósito comercial, y con el presente trabajo la UCI pretende contribuir a la soberanía tecnológica.

1.4 Metodologías de desarrollo de software

Las metodologías son un conjunto de procedimientos, técnicas, herramientas y ayudas para la documentación del proceso de desarrollo de software. Guían el proceso de desarrollo de software, e imponen una disciplina sobre el desarrollo del mismo con el fin de hacerlo más predecible y eficiente. Las características de cada proyecto exigen que sea configurable, condiciones que llevan a decir que no existe una metodología de software universal. Existen en el mundo diferentes metodologías para llevar a cabo el desarrollo de software, las mismas se dividen en dos grupos: metodologías pesadas o tradicionales y las metodologías ágiles.(BARZANALLANA, 2006)

Capítulo 1: Fundamentación Teórica

1.4.1 Proceso Unificado Abierto (OpenUP)

OpenUP es una metodología de desarrollo de software ágil e incremental. Es un proceso iterativo para el desarrollo de software que es mínimo porque solo incluye el contenido del proceso fundamental, completo porque puede ser manifestado como proceso entero para construir un sistema y extensible porque puede ser utilizado como base para agregar o adaptar más procesos. OpenUP es apropiado para desarrollar un software pequeño, que utilice pocos recursos y un tiempo de desarrollo relativamente corto, lo que permite aminorar las probabilidades de fracaso de los proyectos y aumentar la probabilidad de éxito. En esta metodología no se generan documentos, diagramas e iteraciones innecesarias como los requeridos en la metodología RUP (Proceso Racional Unificado). (KAREN, 2008)

El ciclo de vida de un proyecto según la metodología OpenUP se divide en 4 fases fundamentales:

- 1) Concepción: en esta fase se pretende determinar los objetivos y establecer el alcance del proyecto.
- 2) Elaboración: el propósito de esta fase es establecer la línea base de la arquitectura del sistema y proporcionar una base estable para el desarrollo de la siguiente fase.
- 3) Construcción: esta fase tiene como propósito completar el desarrollo del sistema basado en la arquitectura definida, enfocándose en el diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo.
- 4) Transición: en esta fase se asegura que el software esté listo para entregarse a los usuarios.

En el equipo de trabajo se decide utilizar la metodología OpenUP teniendo en cuenta que el tiempo de desarrollo del software es relativamente corto y el equipo de trabajo es pequeño. Además la metodología mantiene las principales características de RUP, obviando las partes opcionales que está propone, lo que permite que el proceso sea más simple. La metodología XP, al igual que la metodología RUP, tampoco es aplicable al escenario debido a que esta exige tener como parte del equipo al usuario final, característica no presente en el proyecto pues el cliente es un agente externo al equipo de desarrollo. Por otra parte, la calidad del software de acuerdo a esta metodología no se basa en la documentación sino en la comunicación fluida con el cliente ya que después de cada fase este recibe una parte funcional, por lo que estará informado continuamente sobre el proyecto y tendrá la posibilidad de intervenir si el objetivo se desvía de sus necesidades.

Capítulo 1: Fundamentación Teórica

1.5 Lenguaje Modelado UML (Unified Modeling Language, UML 2.0)

UML es un lenguaje de modelado que permite visualizar, especificar y documentar cada una de las partes que comprende el desarrollo del software. Permite modelar elementos conceptuales como son los procesos de negocio, las funciones de sistema, clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables, así como distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Dentro de sus objetivos tiene: brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes. Prescribe un conjunto de notaciones y diagramas estándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que significan estos diagramas y símbolos.(RUMBAUGH, 2000)

Se decidió utilizar el UML pues en la actualidad es uno de los lenguajes de modelado más conocido y utilizado. Además, permite visualizar, especificar, construir y documentar un software y la metodología de desarrollo seleccionada (OpenUP) lo propone como lenguaje de notación.

1.6 Herramientas CASE para la modelación del sistema

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.(PERISSÉ, 2001) Estas herramientas permiten organizar y manejar la información de un proyecto informático, permitiéndole a los participantes de un proyecto, que los sistemas (especialmente los complejos), se tornen más flexibles y comprensibles.

1.6.1 Visual Paradigm 8.0

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite modelar todos los tipos de diagramas de clases, permite realizar código inverso, generar código desde diagramas y generar documentación.(MORA, 2007)

Una de sus ventajas sobre las demás herramientas CASE es su condición multiplataforma, por lo que tiene la capacidad de ejecutarse sobre diferentes Sistemas Operativos. Es fácil de usar y presenta una agradable interfaz para interactuar con el usuario. Se integra con el Visio para importar imágenes del mismo y realizar los diagramas de despliegue, y genera documentación para el proyecto en HTML, MS Word y PDF.

Capítulo 1: Fundamentación Teórica

La herramienta CASE seleccionada fue Visual Paradigm tomando como punto de partida que permite el diseño del producto de forma rápida y con calidad, brindando facilidades para el diseño de los diagramas, así como su documentación. Además la Universidad cuenta con la licencia para su uso no siendo el caso de la herramienta Rational Rose. Otro de los factores influyentes es que entre sus principales características y facilidades se encuentran: que es multiplataforma, se utiliza para realizar productos de calidad, soporta varios idiomas, generación de código para Java, exportación como HTML y se integra con herramientas Java.

1.7 Lenguajes de Programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; permitiendo al desarrollador comunicarse con los dispositivos de hardware y software existentes. A continuación se describen algunos de estos lenguajes utilizados en el desarrollo de la aplicación.(KERNIGHAN, 1991)

1.7.1 Lenguaje de programación del lado del servidor PHP

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación. Dentro de las ventajas que presenta el uso de PHP destacan las siguientes: es de fácil aprendizaje, es un lenguaje muy rápido, soporta la programación orientada a objetos y puede ser utilizado tanto sobre Linux como Windows. Tiene capacidad de conexión con la mayoría de los gestores de bases de datos tales como: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros. Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones. Es libre y no requiere definición de tipos de variables ni manejo detallado del bajo nivel.(GARCÍA, 2000)

1.7.2 Lenguaje de programación del lado del cliente J2ME

Por el lado de desarrollo de aplicaciones para celulares se tienen lenguajes como C++, C# y Java en versiones especiales para los equipos en cuestión. En el presente trabajo se centrará la investigación en el lenguaje Java, ya que ocupa una abrumadora mayoría en el mercado y está demostrando ser la mejor opción disponible para todos los programadores interesados en el tema, además de las características de seguridad y estabilidad inherentes a cualquier producto Java.

La Plataforma Java se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real. Por ejemplo, los usuarios finales suelen interactuar con la máquina virtual de Java y el conjunto estándar de bibliotecas.

Capítulo 1: Fundamentación Teórica

Además, las aplicaciones Java pueden usarse de forma variada. La base de la plataforma Java es una máquina virtual, la cual puede ser implementada en los más populares sistemas operativos y en gran variedad de hardware. Por lo que se puede tener aplicaciones binarias Java operando consistentemente a través de diferentes implementaciones. Juntos, el lenguaje de programación Java, la máquina virtual y las Apis, forman la plataforma Java la cual, en su versión 2, se puede encontrar en tres ediciones:

1. Java 2 Standard Edition (J2SE). Diseñada para computadoras de escritorio, puede trabajar en sistemas operativos como: Windows, Linux, MacOS y Solaris.
2. Java 2 Enterprise Edition (J2EE). Plataforma para aplicaciones multiusuario o empresariales. Se basa en J2SE y agrega Apis para trabajo en el servidor.
3. Java 2 Micro Edition (J2ME). Conjunto de tecnologías y especificaciones desarrolladas para dispositivos pequeños como los teléfonos celulares y PDA (agendas electrónicas). Utiliza derivados de componentes J2SE, como son una máquina virtual más pequeña y un conjunto de Apis menos potentes.

La Plataforma Java, Micro Edición (del inglés: Java 2 Micro Edition) o J2ME es la única plataforma en la que se puede desarrollar midlets para los celulares, esto significa que no hay un similar en plataformas. Esta versión de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (del inglés: Kilo Virtual Machine) que requiere sólo unos pocos Kilobytes de memoria para funcionar; en vez del uso de la JVM (del inglés: Java Virtual Machine) clásica, y la inclusión de un pequeño y rápido recolector de basura.

J2ME se basa en los conceptos de configuración y perfil. Una configuración describe las características mínimas en cuanto a la configuración de hardware y software. La configuración que usa J2ME es la CLDC (del inglés: Connected Limited Device Configuration). Concretamente CLDC define:

1. Cuáles son las características del lenguaje Java incluidas.
2. Qué funcionalidad será incluida en la máquina virtual Java.
3. Las Apis necesarias para el desarrollo de aplicaciones en móviles.

Capítulo 1: Fundamentación Teórica

4. Los requerimientos hardware de los dispositivos.

J2ME está formado por la configuración CLDC y por el perfil MID (del inglés: Mobile Information Device Profile) (conocido por MIDP o MID Profile). CLDC es una especificación general para un amplio abanico de dispositivos. Un perfil define las características del dispositivo de forma más específica. MIDP define las Apis y características de hardware y software necesarias para el caso concreto de los teléfonos móviles.(ABOUT JAVA ME, 2005)

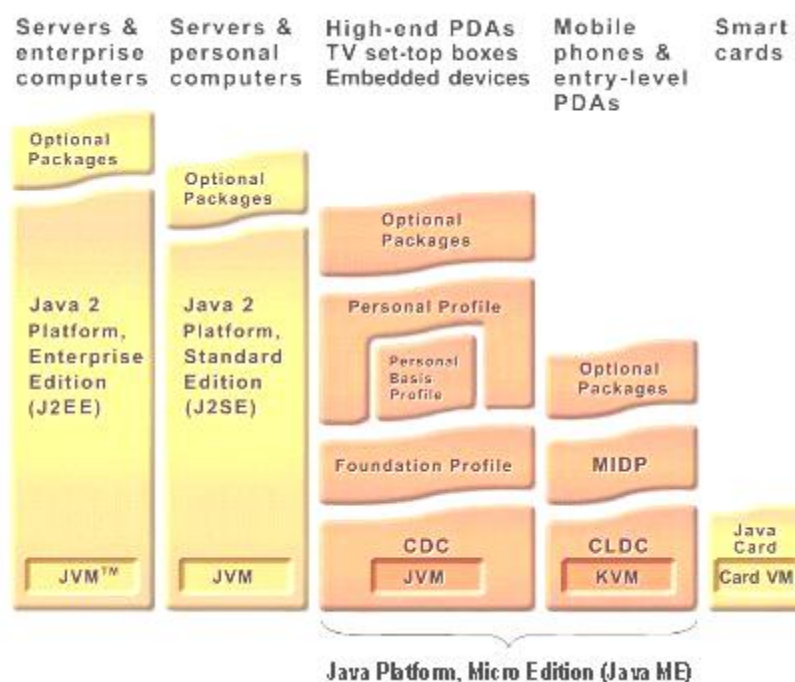


Figura 1. Componentes de la tecnología Java ME

1.9 Entorno de Desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado (IDE, Integrated Development Enviroments) es un programa informático compuesto por herramientas de programación. Consiste en un compilador, un editor de código, un depurador y un constructor de interfaz gráfica (GUI), empaquetados como un programa de aplicación. Provee un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, Visual Basic y Delphi.(LUCIANO, 2008)

1.9.1 NetBeans 6.9

NetBeans ayuda a la flexibilidad en el trabajo por sus posibilidades de modificación por parte de los desarrolladores. Ofrece además una serie de ventajas que lo hacen realmente codiciado como son: la

Capítulo 1: Fundamentación Teórica

creación de aplicaciones multiplataforma, permite que las aplicaciones creadas por el IDE se adhieran a los estándares de la industria, proporciona facilidades y garantías para la migración, facilidad de uso, cumplimiento de regulaciones, y flexibilidad entre plataformas.(BOUDREAU, 2002)

Este IDE propicia la integración de los diferentes módulos para la creación de programas en lenguajes como: Java, PHP, C++, Python, Ruby y JSP. Incluye soporte para desarrollar y probar aplicaciones para dispositivos móviles que utilicen Java 2 Micro Edition (J2ME), MIDP (Mobile Information Device Profile) 2.0, CLDC (Connected, Limited Device Configuration) 1.1. Favorece el desarrollo de interfaces gráficas de usuario con el diseñador visual móvil (VMD).

Por defecto, Netbeans incluye emuladores básicos para probar el funcionamiento de aplicaciones orientadas a dispositivos móviles. Sin embargo, cada fabricante de dispositivos provee emuladores más específicos para emular la ejecución de sus aplicaciones obteniendo un entorno de prueba robusto. Así por ejemplo, si la aplicación está siendo desarrollada para celulares Nokia 6100, se tiene la posibilidad de descargar un emulador de dicha plataforma para poseer una ejecución mucho más real.

Para la realización del sistema, se selecciona el emulador 3 del paquete de movilidad del netbeans (ver anexo 1) por encontrarse integrado a la herramienta IDE escogida para desarrollar la interfaz web de captura de datos en el Sistema Ligero de Captura de Datos.

Conclusiones del Capítulo 1

En este capítulo se realizó un análisis de los elementos fundamentales de las diferentes herramientas de captura de datos. Se analizaron las técnicas y herramientas fundamentales para dar solución a la problemática existente. Con el objetivo de obtener un proceso disciplinado sobre el desarrollo de software y con el fin de hacerlo más predecible y eficiente se empleará como metodología de desarrollo OpenUP, como lenguaje de modelado UML y como herramienta de modelado Visual Paradigm 8.0 ya que la universidad cuenta con la licencia para su uso. Como lenguajes de programación, del lado del servidor se escogió PHP en su versión 5 y J2ME del lado del cliente. En aras de lograr la organización del código se seleccionó el entorno integrado de desarrollo Netbeans 6.9 con soporte para los lenguajes de programación a utilizar.

Capítulo 2: Análisis y Diseño

CAPÍTULO 2: ANÁLISIS Y DISEÑO

Introducción al Capítulo 2

En este capítulo se enfatiza en la propuesta de solución para el problema de la investigación. Con la ayuda de definiciones y conceptos, se explicará el modelo de dominio para representar la estructura que se desea constituir en el sistema a desarrollar. Se especifican los requisitos funcionales y no funcionales de la aplicación a desarrollar y se realiza una descripción de los actores que interactúan con las herramientas y de los casos de usos identificados. Además se transforman los requisitos en detalles que muestran cómo implementar el sistema, dando paso al diseño de la aplicación generando los artefactos necesarios para cada fase. Posteriormente se diseñaron los diagramas de clases y de interacción para los casos de uso arquitectónicamente significativos y se definió con precisión la estructura física que presentará el sistema mediante el modelo de despliegue así como la descripción de los patrones de diseño utilizados.

2.1 Descripción del sistema

El sistema a desarrollar es una Interfaz Web que permite la captación de información en dispositivos móviles a partir de las plantillas de formularios especificadas en el Sistema Ligero de Captura de Datos (SIGELITE). Se desarrollará una interfaz web capaz de llevar a cabo las funcionalidades implementadas en SIGELite para entornos desktop. Dicha interfaz permitirá realizar una administración de las plantillas y recursos del sistema, así como también tareas de exportación e importación tanto de plantillas como de recursos a formato XML. Todo ello debe ser mostrado en una Interfaz Web en dispositivo móvil amigable y asequible para el usuario.

2.2 Modelo de Dominio

Un modelo de dominio es un artefacto de la disciplina de análisis construido con las reglas de UML durante la fase de concepción. Presenta uno o más diagramas de clases que no contienen conceptos propios de un sistema de software sino de la propia realidad física. Es un subconjunto del modelo de negocio y se realiza cuando no están claros los procesos o cuando no se identifican claramente los actores y trabajadores del negocio. Permite la captura de los objetos más importantes del negocio o los eventos que suceden en el entorno donde estará el sistema. (2009)

Capítulo 2: Análisis y Diseño

2.3 Diagrama conceptual del modelo de dominio

El modelo de dominio es fundamental para comprender y describir los conceptos más importantes dentro del contexto del sistema, es una representación visual del entorno real del proyecto. La siguiente figura muestra el Modelo de Dominio para el negocio de captura de datos.

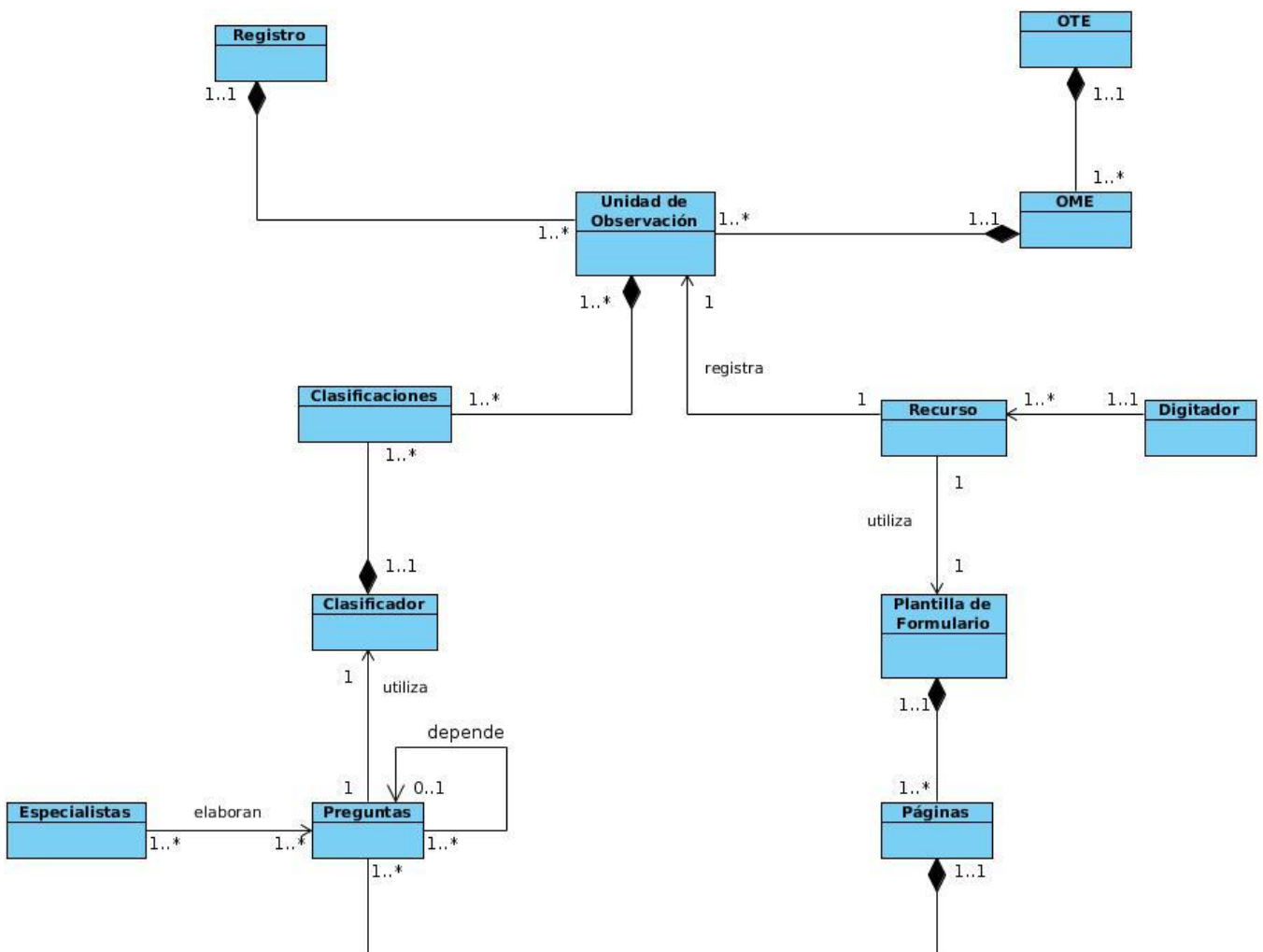


Figura 2: Diagrama del Modelo de Dominio

2.4 Definición de clases del modelo de dominio

Páginas: Asociadas a la plantilla de formulario.

Capítulo 2: Análisis y Diseño

Preguntas: Diseñadas por los especialistas y contenidas en las páginas de la plantilla de formulario.

Plantilla de Formulario: Soporte para la captura de información.

OTE: Oficina Territorial de Estadística. Es una oficina que está por debajo de la Oficina Nacional Estadística y es a nivel de provincia.

OME: Oficina Municipal de Estadística. Es una oficina que está por debajo de la Oficina Territorial Estadística y está a nivel de municipios.

Registro: Agrupan a las unidades de observación de acuerdo a la actividad económica que desempeñan.

Unidad de Observación: Constituye un elemento base para la captación de formularios ya que se registra en la plantilla de formulario para el reporte de información.

Clasificador: Asociados a la definición de las preguntas.

Clasificaciones: Se asocian al clasificador como elemento más específico de la definición del mismo.

Recurso: Información capturada a partir de una plantilla de formulario definida.

Digitador: Es el encargado de capturar la información utilizando los recursos.

2.5 Especificación de los Requisitos del Sistema

Los requerimientos de un sistema describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento. Los requerimientos no son más que propiedades o restricciones determinadas de forma precisa que deben satisfacerse. Se pueden clasificar en: funcionales y no funcionales.(AGUT, 2008)

2.5.1 Requerimientos Funcionales

Los requerimientos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. En algunos casos, los requisitos funcionales declaran explícitamente lo que el sistema no debe hacer, es decir definen las funciones que el sistema será capaz de realizar.(Tecnología y Synergix, 2009)

Se identificaron los siguientes requisitos funcionales:

RF1- Exportar plantilla de formulario.

Capítulo 2: Análisis y Diseño

- RF2- Importar plantilla de formulario.
- RF3- Eliminar plantilla de formularios.
- RF4- Buscar plantillas de formularios.
- RF5- Listar plantillas de formularios.
- RF6- Captar plantilla de formulario.
- RF7- Modificar recurso.
- RF8- Listar recursos.
- RF9- Buscar recursos.
- RF10- Eliminar recurso.
- RF11- Exportar recurso.
- RF12- Importar recurso.
- RF13- Autenticar Usuario.

2.5.2 Requerimientos no Funcionales

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Estas propiedades o cualidades se refieren a las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales son importantes para el posterior éxito del producto; normalmente están vinculados a los requerimientos funcionales.

Los requisitos no funcionales les muestra a los clientes y usuarios si el sistema cumple con las funcionalidades requeridas, que tan usable es, cuán seguro, marcando la diferencia entre un producto bien aceptado y uno con poca aceptación.(Tecnología y Synergix, 2009)

Para el componente propuesto se definen los siguientes requisitos no funcionales:

Usabilidad

Debe poseer una interfaz agradable para el cliente de acuerdo a los estándares de diseño, además de ser intuitivo y tener un alto nivel de usabilidad. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de dispositivos móviles y de un ambiente web en sentido general. El sistema debe tener acceso al menú general desde cualquiera de sus páginas.

Capítulo 2: Análisis y Diseño

Seguridad

La información asociada a la aplicación deberá estar protegida de acceso no autorizado haciendo uso de mecanismos de autenticación que garanticen la protección de los datos. La autenticación se realizará mediante nombre de usuario y contraseña, ambos contenidos en un fichero de configuración del servidor, en el cual la contraseña aparecerá encriptada usando MD5 (del inglés: Message-Digest Algorithm).

Confiabilidad

La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en la base de datos desde donde se extraigan las plantillas de los formularios y los recursos.

Apariencia o Interfaz externa

El sistema deberá poseer una interfaz web sencilla, amigable, lo más clara e intuitiva posible para el usuario además de estar bien detallada y organizada la información que se mostrará en la interfaz. Usar la menor cantidad de imágenes y en el formato más idóneo de acuerdo a las características del móvil, además de tener buena calidad visual y ser comprensibles. El sistema debe comunicarse a través del protocolo de comunicación HTTP y debe validar la entrada de información del usuario, e identificar los errores que se detecten para que el usuario los corrija.

Portabilidad

Debe ser portable a los modelos de teléfonos móviles que soporten aplicaciones Java, por lo que el diseño y la implementación no deben ser dependientes del dispositivo hardware en el que se implante.

Hardware

Requerimientos mínimos del servidor:

- Ordenador Pentium II o superior, con 1.5 GHz de velocidad de microprocesador.
- Memoria RAM mínimo 128MB.
- Dispositivo extraíble con 1Gb de capacidad para instalar el sistema.

Capítulo 2: Análisis y Diseño

Software

Requerimientos mínimos para la máquina servidora:

- Sistema operativo Linux o Windows.
- Servidor de aplicaciones Apache, versión 2.
- Lenguaje de programación PHP y librerías, versión 5.3.3.
- Sistema Gestor de Base de Datos SQLITE, versión 3.7.3.

2.6 Modelo de Caso de Uso del Sistema

El modelo de casos de uso del sistema representa las relaciones existentes entre actores y casos de uso. Los actores son terceros fuera del sistema que interactúan con él (puede ser cualquier persona, individuo, grupo, entidad, organización, máquina o sistema de información externo; con los que el sistema interactúa) y los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

2.6.1 Definición de los actores del sistema

Los actores representan un tipo de usuario del sistema. Se entiende como usuario cualquier elemento externo que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático, unidades organizativas o empresas. Un actor usa un caso de uso para desempeñar alguna porción de trabajo que es de valor para el negocio. El conjunto de casos de uso al que un actor tiene acceso define su rol global en el sistema y el alcance de su acción. Además son generalmente responsables de realizar actividades que serán automatizadas en el sistema.

Se identificaron como actores para la interfaz web propuesta:

Actor	Descripción
Usuario	Es el responsable de importar, exportar, administrar y captar plantilla de formulario, administrar recurso, exportar e importar recurso.
SIGELite	Es el responsable de devolver las funcionalidades que se implementaron en el sistema SIGELite, para mostrarlas en

Capítulo 2: Análisis y Diseño

	el dispositivo móvil.
--	-----------------------

Tabla 1. Actores del componente propuesto

2.6.2 Diagrama de Caso de Uso del sistema

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa y cada actor está unido a los casos de uso en los que participa mediante una línea. Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

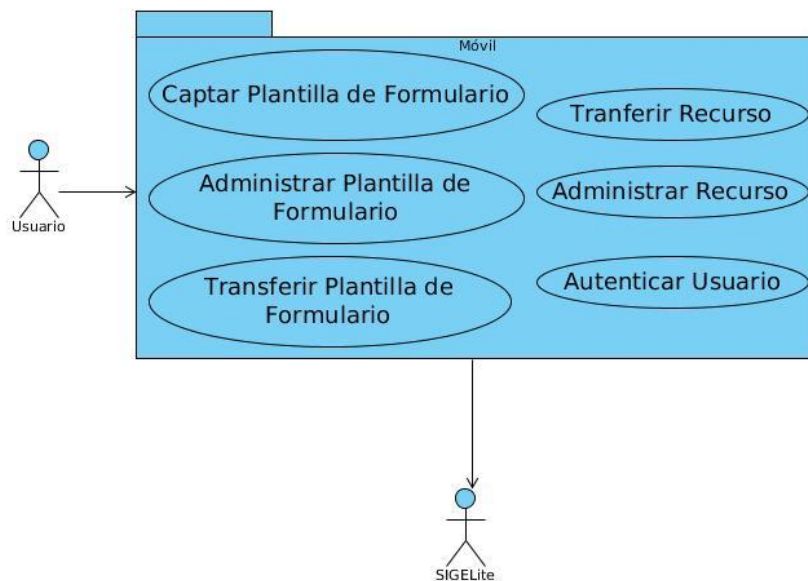


Figura 3. Diagrama de Casos de Uso del Sistema.

2.6.3 Descripción textual de los Casos de Uso del sistema

Los casos de uso deben ser detallados por una descripción textual que muestra la funcionalidad que se implementará en la interfaz web propuesta. A continuación se muestra la descripción de uno de los casos de usos más importantes del sistema.

Capítulo 2: Análisis y Diseño

CU_Administrar Recurso.

Objetivo	Modificar, Eliminar, Buscar o Listar un recurso.	
Actores	Usuario, SIGELite	
Resumen	<p>El caso de uso se inicia cuando el Usuario desea realizar una de las siguientes acciones:</p> <p>Eliminar recurso: El usuario desea eliminar un recurso y selecciona para ello un recurso de la lista de recursos existentes. El sistema se conecta a SIGELite y se elimina dicho recurso finalizando así el caso de uso.</p> <p>Buscar recurso: El usuario desea buscar un recurso. El sistema muestra una interfaz para que el usuario especifique los parámetros de búsqueda, el sistema se conecta a SIGELite y muestra todos los recursos que coincidan con los parámetros introducidos finalizando así el caso de uso.</p> <p>Listar recursos: El usuario desea listar los recursos, el sistema se conecta con SIGELite y muestra un listado de todos los recursos existentes.</p>	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Deben existir recursos en el sistema.	
Postcondiciones	Se Modifican, Buscan, Eliminan y Listan los recursos según la operación realizada.	
Flujo Normal de Eventos		
	Actor	Sistema
1.	El usuario selecciona la opción que desea realizar con el recurso, dígame:	

Capítulo 2: Análisis y Diseño

	<ol style="list-style-type: none"> 1. Buscar recursos. 2. Eliminar recurso. 3. Listar recursos. 4. Modificar recurso. 	
2.		<p>El sistema en dependencia de la opción seleccionada muestra la interfaz correspondiente:</p> <ol style="list-style-type: none"> 1. Buscar recursos. Ver Sección 1: “Buscar recursos”. 2. Eliminar recurso. Ver Sección 2: “Eliminar recurso”. 3. Listar recursos. Ver Sección 3: “Listar recursos”. 4. Modificar recurso. Ver Sección 4: “Modificar recurso”.
Sección 1: “Buscar recursos”		
	Actor	Sistema
1.	Selecciona la opción buscar recursos.	
2.		Muestra una interfaz para especificar los parámetros de búsqueda.
3.	Llena los campos con los criterios deseados y posteriormente presiona el botón aceptar.	
4.		Se conecta a SIGELite consumiendo el servicio buscar recurso. Muestra en la interfaz principal “Recurso” la o las

Capítulo 2: Análisis y Diseño

		plantillas que cumplan con el criterio o los criterios definido(s) por el actor anteriormente.
Flujos alternos		
3a. El usuario no desea buscar el recurso.		
	Actor	Sistema
1.	Selecciona el botón cancelar.	
2.		Cancela la ejecución de la operación y vuelve al entorno de trabajo.
Sección 2: “Eliminar recurso”		
	Actor	Sistema
1.	Selecciona el recurso y posteriormente selecciona el botón eliminar recurso.	
2.		Muestra una ventana para confirmar si desea eliminar el recurso.
3.	Selecciona la opción aceptar si desea continuar la operación.	
4.		Se conecta a SIGELite eliminando el recurso especificado.
Flujos alternos		
3a. El usuario no desea eliminar el recurso.		
	Actor	Sistema

Capítulo 2: Análisis y Diseño

1.	No desea eliminar el recurso y selecciona el botón cancelar.	
2.		Cancela la ejecución de la operación y vuelve al entorno de trabajo.
Sección 3: “Listar recursos”		
	Actor	Sistema
1.	Selecciona la opción Recursos.	
2.		Se conecta a SIGELite y muestra una ventana con el listado de recursos existentes.
Sección 4: “Modificar recurso”		
	Actor	Sistema
1.	Selecciona el recurso y posteriormente el botón modificar recurso.	
2.		Se conecta a SIGELite y muestra en una nueva ventana el recurso con todos los datos que contiene.
3.	Modifica los campos del formulario y presiona la opción guardar.	
4.		Modifica el formulario correctamente y guarda el recurso.
Flujos alternos		

Capítulo 2: Análisis y Diseño

3a. El usuario no desea modificar el recurso.		
	Actor	Sistema
1.	No desea modificar el formulario y selecciona la opción cancelar.	
2.		Cancela la ejecución de la operación y vuelve al entorno de trabajo.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	RF7, RF8, RF9, RF10.	

Tabla 2. Descripción textual del caso de uso Administrar Recurso.

2.7 Diseño del Sistema

El diseño es la etapa que le sigue al análisis del sistema. Desarrolla las reglas propuestas en el análisis con la meta de satisfacer las necesidades del usuario reflejadas tanto desde el punto de vista funcional como no funcional. El diseño del sistema es el arte de definir la arquitectura de hardware, software, componentes, módulos y datos de un sistema de cómputo para satisfacer ciertos requerimientos.

Para hacer un diseño eficiente se tomarán en cuenta un conjunto de patrones, dado que los mismos son una guía importante para solucionar problemas comunes en programación que generalmente encuentras o están presentes en el diseño de un programa. Cada patrón explica explícitamente cómo resolver un determinado problema, bajo determinadas circunstancias.

2.8 Modelo del Diseño

Una entrada principal en el diseño es el resultado del análisis, el modelo de análisis que proporciona un entendimiento detallado de los requisitos. En el diseño se modela el sistema y se encuentra su

Capítulo 2: Análisis y Diseño

forma, incluyendo la arquitectura, para que soporte todos los requisitos funcionales y las restricciones que se le suponen. Es específico para una implementación y depende del lenguaje de programación.

2.9 Diagramas de clases del diseño

Un diagrama de Clases del Diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Una clase es una descripción de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica. (RUMBAUGH, 2000)

A continuación se mostrará el diagrama de clases del diseño utilizado en la realización de este trabajo para el caso de uso más significativo.

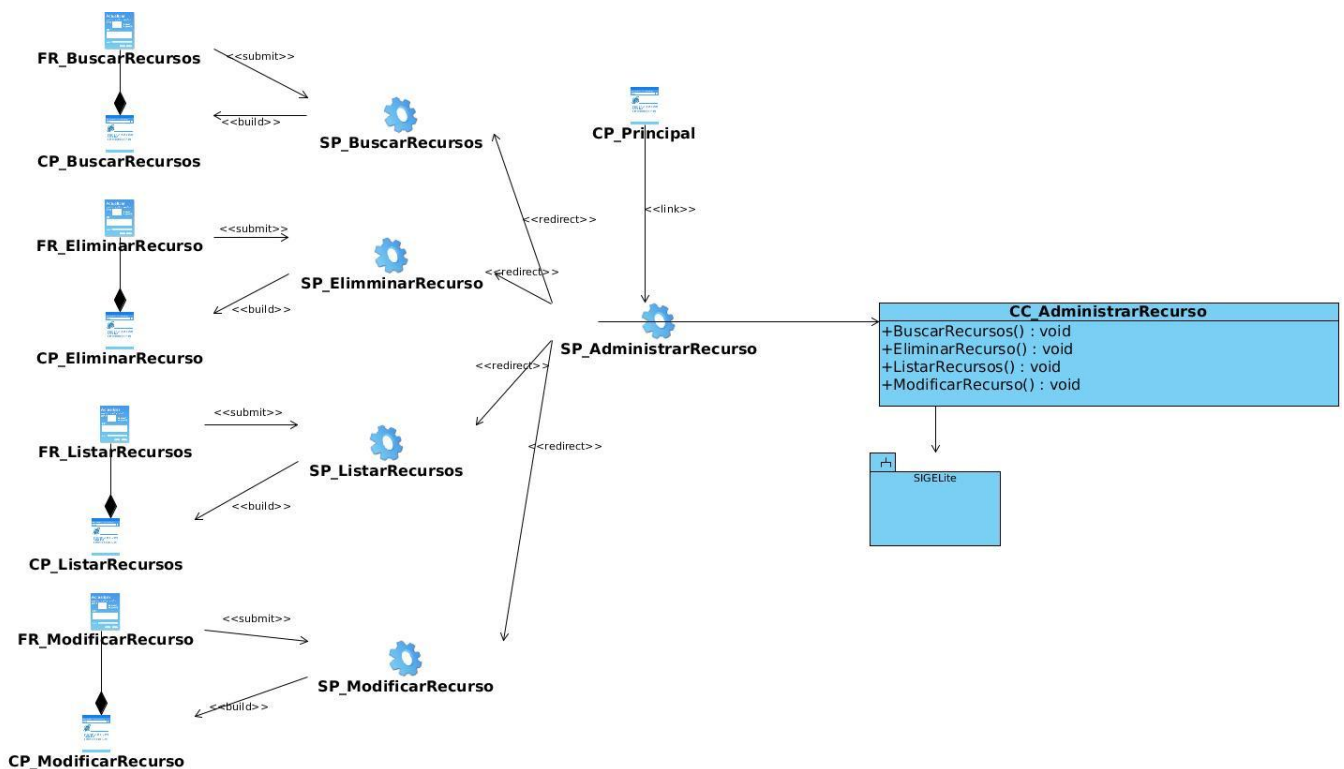


Figura 4. Diagrama de clases del diseño del CU Administrar Recurso.

2.10 Patrones de Diseño

Un patrón describe un problema que ocurre varias veces y describe también el núcleo de la solución al problema, de forma que puede utilizarse en ilimitadas ocasiones sin tener que hacer dos veces lo mismo. (LARMAN, 2003)

Capítulo 2: Análisis y Diseño

Los patrones de diseño son utilizados para los aspectos más específicos de diseño de los subsistemas. En el diseño y desarrollo del componente a desarrollar, se utilizan algunos de los patrones GRASP (del inglés: General Responsibility Assignment Software Patterns) para la asignación de responsabilidades en el diseño de objetos. Por las características de estos patrones, se consideró necesaria su utilización en este trabajo para obtener un producto de mayor calidad.

Bajo Acoplamiento

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. En este sentido, un alto acoplamiento tiene como consecuencia que las clases sean más difíciles de entender cuando estén aisladas y de reutilizar debido a que se requiere la presencia de otras clases de las que dependen. Por lo que un bajo acoplamiento plantea que deben existir pocas dependencias entre las clases. El Bajo Acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también más reutilizables.

En el diagrama de clase del diseño presentado en la figura 4 este patrón se evidencia en el nivel de dependencia establecida entre la clase controladora con el sistema SIGELite, ya que las mismas se establecen a través de una sola relación de dependencia.

Alta Cohesión

Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas para que no realicen un gran trabajo. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos. En el sistema se tiene bien definido las responsabilidades de cada clase, por ejemplo, se definen clases del negocio por cada caso de uso o por cada requisito funcional especificado, para evitar la sobrecarga de funcionalidades en una clase. Esto queda evidenciado en el diagrama de clases del diseño anteriormente presentado donde la clase CC_AdministrarRecursos tiene definida como única responsabilidad la asociada a la administración de recursos evitando sobrecargar las clases asociadas a ella con otras responsabilidades dentro del sistema. (Ver figura 4)

Experto

Este patrón de diseño define asignar la responsabilidad a la clase que cuente con la información necesaria para manejar esta responsabilidad. Es necesario que la asignación de responsabilidades a las clases se lleve a cabo de una forma adecuada, lo que permitirá que el sistema sea más fácil de entender así como que se puedan reutilizar componentes. Con su aplicación se conserva el

Capítulo 2: Análisis y Diseño

encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. El comportamiento se distribuye entre las clases que cuentan con la información requerida permitiendo la creación de clases sencillas y más cohesivas. Este patrón queda evidenciado en el diagrama de clases del diseño del caso de uso Administrar Recursos (ver figura 4), donde la clase controladora CC_AdministrarRecursos es la experta en información ya que la misma encapsula la información relacionada con la administración de los recursos del sistema de manera sencilla.

Controlador

Este patrón GRASP establece la asignación de la responsabilidad de manejar los eventos a una clase que será la encargada de controlarlos. Permite la reutilización de componentes al delegar las responsabilidades en una clase, lo que facilita que aunque existan cambios desde el punto de vista de la interfaz del sistema las funcionalidades se mantengan intactas y puedan ser reutilizadas. Aplicado en el sistema, este patrón se ve reflejado en las clases controladoras, las cuales son las encargadas de manejar las peticiones del cliente. (Ver figura 4)

2.11 Diagramas de interacción del diseño

Los diagramas de interacción modelan los aspectos dinámicos del sistema, mostrando las interacciones entre objetos mediante transferencia de mensajes entre ellos. Dentro de estos diagramas están los de secuencia y los de colaboración. En el presente trabajo se emplearon los diagramas de secuencia teniendo en cuenta que muestran las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto.

A continuación se presenta el diagrama de interacción del CU Administrar Recurso específicamente para la sección Modificar Recurso, el resto de los diagramas de dicho caso de uso pueden verse en el anexo 2. En dicha figura se presenta el flujo de eventos que debe realizar el actor Usuario para llevar a cabo la modificación de un recurso. El actor debe seleccionar el recurso a modificar del formulario principal (FR_Principal) para posteriormente hacer la petición de modificar en el mismo, dicha petición es recibida por la página servidora (SP_AdministrarRecurso) la cual se encarga de redireccionarla a la página servidora que cuenta con la información necesaria para realizar esta labor (SP_ModificarRecurso). La clase controladora CC_AdministrarRecurso es la que contiene todas las funcionalidades para hacer la petición al servidor SIGELite el cual es el elemento crucial en toda la interacción pues se encarga tanto de enviar a la SP_ModificarRecurso el encabezado y cuerpo del recurso seleccionado, así como de llevar a cabo la modificación del recurso cuando le sean enviados

Capítulo 2: Análisis y Diseño

la información. La otra clase presentada es la página cliente modificar recurso (CP_ModificarRecurso) con su respectivo formulario (FR_ModificarRecurso) que son aquellos donde se mostrarán todos los datos posibles a modificar.

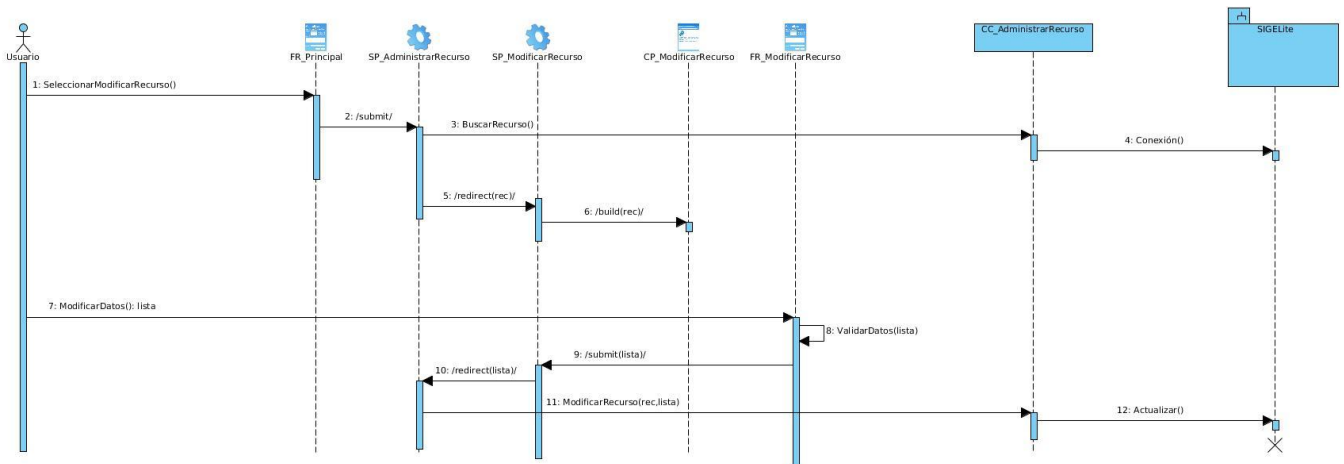


Figura 5. Diagrama de secuencia del CU Administrar Recurso sección Modificar Recurso.

2.12 Diagrama de Despliegue

Un diagrama de despliegue modela la topología del hardware sobre el que se ejecuta un sistema. Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Un nodo es un elemento físico que existe en tiempo de ejecución. Representa un recurso computacional que, por lo general, tiene memoria y capacidad de almacenamiento. Además, representa el despliegue físico de un componente. (RUMBAUGH, 2000)

A continuación se muestra el diagrama de despliegue del sistema:

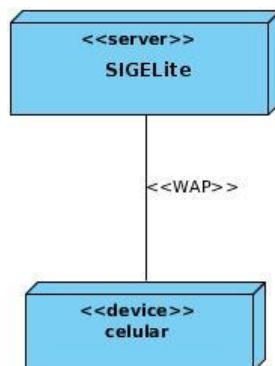


Figura 6. Diagrama de Despliegue.

Capítulo 2: Análisis y Diseño

Conclusiones del Capítulo 2

En el presente capítulo se realizó el análisis y el diseño de la aplicación el cual estuvo conformado por el modelo de dominio, los requisitos funcionales y no funcionales, quedando confeccionado el diagrama de casos de uso del sistema. Además, se ilustraron los diagramas de clases del sistema así como los patrones de diseño a emplear para el futuro desarrollo de dicho sistema. Teniendo en cuenta los casos de uso, se realizaron los diagramas de secuencia, reflejando un proceso lógico de acciones para resolver las peticiones del cliente. A través del diagrama de despliegue se describió como se realizará el despliegue de los componentes a lo largo de la infraestructura del sistema, para así dar comienzo a la implementación de la aplicación.

Capítulo 3: Implementación y Prueba

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Introducción al Capítulo 3

Durante este capítulo se realiza el flujo de trabajo de implementación, se analizan los artefactos principales del sistema como el modelo de implementación que incluye componentes, subsistemas de implementación y diagramas de componentes. Se realiza el inicio de la implementación del sistema mediante el término de componentes, se muestran fragmentos de las clases e implementaciones significativas de la Interfaz Web de Captura de Datos en Dispositivos Móviles en el Sistema Integrado de Gestión Estadística y las pruebas realizadas para su validación.

4.1 Modelo de implementación

El modelo de implementación representa la composición física de la implementación en término de subsistemas de implementación y elementos de implementación. Describe cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación y cómo dependen los componentes unos de otros. Dicho modelo se considera el artefacto más significativo del flujo de trabajo de implementación debido a la importancia que tiene para los desarrolladores. Este modelo está conformado por el diagrama de componente.(RUMBAUGH, 2000)

4.1.1 Diagrama de Componentes

Dentro del Modelo de implementación se encuentran los diagramas de componentes. Un diagrama de componentes representa cómo un sistema es dividido en componentes y muestra las dependencia entre estos además describe los elementos físicos y sus realizaciones en el entorno de implementación y en el lenguaje de programación utilizado.(PRESSMAN, 2005)

Capítulo 3: Implementación y Prueba

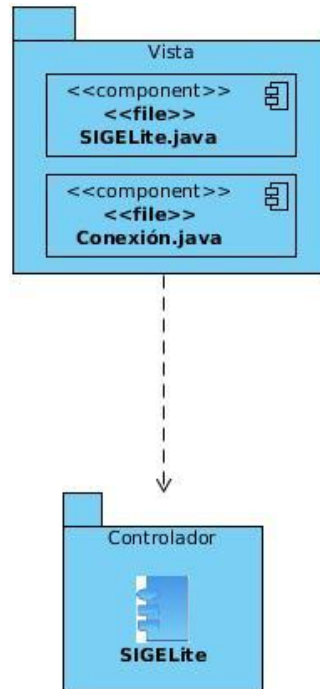


Figura 7. Diagrama de Componentes.

4.2 Código Fuente

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido; como resultado se obtienen archivos que contienen el código fuente de la aplicación. Este no es más que un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. En el código fuente de un programa está descrito por completo su funcionamiento.(LASSO, 2008)

4.2.1 Estándares de codificación

La combinación de técnicas de codificación sólidas y las buenas prácticas de programación con el objetivo de lograr un código robusto, es de vital importancia para la calidad del software. La aplicación continua de un estándar de codificación puede contribuir a obtener un sistema de software fácil de comprender y de mantener.(RUMBAUGH, 2000)

Nomenclatura

- Los nombres de cada uno de los elementos del programa deben ser significativos; cada nombre debe explicar lo que realiza el elemento en la aplicación.

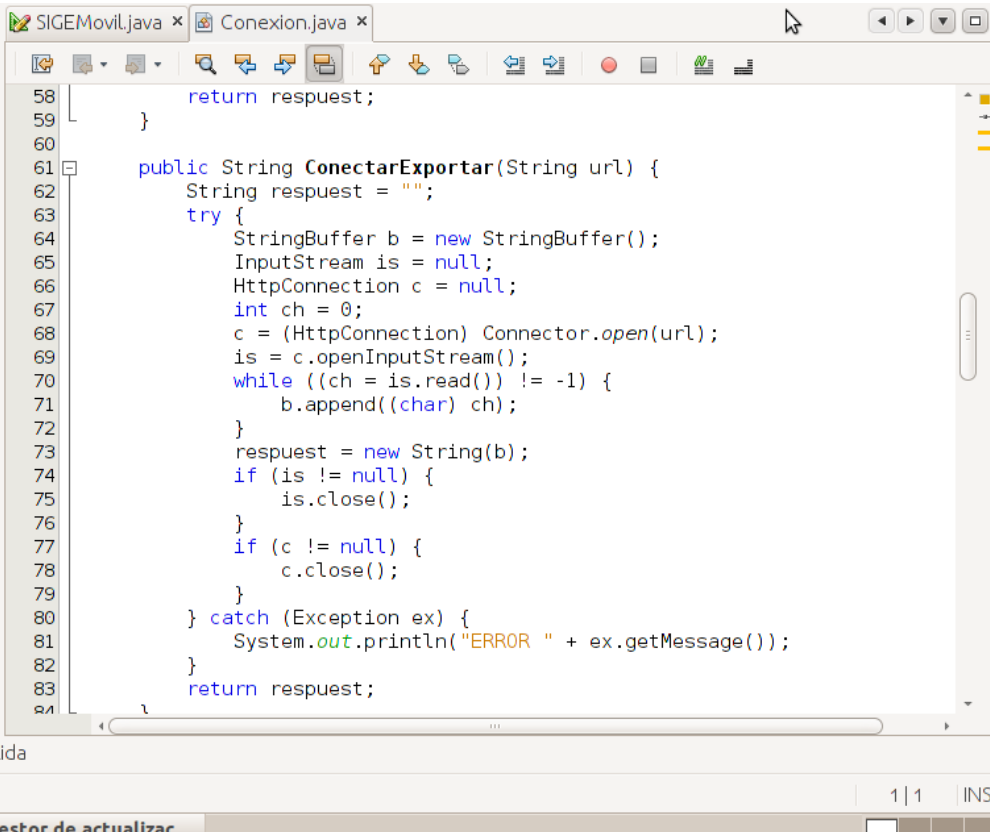
Capítulo 3: Implementación y Prueba

- La forma de construir los nombres será colocando primero el verbo o el sustantivo, y luego su complemento con la primera letra en mayúscula.
- La mayoría de los elementos se deben nombrar usando sustantivos.

Estilo de codificación utilizado

- Los bloques de códigos siempre deben estar encerrados por llaves (excepto si constan de una línea).
- Declaraciones dentro de las clases.
- Enunciado dentro de métodos y funciones.
- Enunciados dentro de bloques de comandos.

Ejemplos de código fuente



```
58     return respuest;
59 }
60
61 public String ConectarExportar(String url) {
62     String respuest = "";
63     try {
64         StringBuffer b = new StringBuffer();
65         InputStream is = null;
66         HttpConnection c = null;
67         int ch = 0;
68         c = (HttpConnection) Connector.open(url);
69         is = c.openInputStream();
70         while ((ch = is.read()) != -1) {
71             b.append((char) ch);
72         }
73         respuest = new String(b);
74         if (is != null) {
75             is.close();
76         }
77         if (c != null) {
78             c.close();
79         }
80     } catch (Exception ex) {
81         System.out.println("ERROR " + ex.getMessage());
82     }
83     return respuest;
84 }
```

Figura 8. Ejemplo del código fuente de la aplicación.

Capítulo 3: Implementación y Prueba

4.3 Pruebas

La calidad de un sistema está determinada, entre otras cosas, por la coincidencia entre lo que se programó y los requisitos establecidos en la primera fase. Para comprobar el grado de cumplimiento de estos requisitos se usan las pruebas del sistema. Estas definen un conjunto amplio de acciones de comprobación que abarcan todas las características que determinan la calidad de un software.

Se comprueban las funcionalidades diseñando casos de prueba que definen cómo proceder. Estos casos de prueba incluyen los juegos de datos a usar que son los válidos o esperados y los no válidos o no esperados por el programa. Además establecen los resultados a alcanzar en correspondencia de la lógica del programa y los datos ingresados. Describen las condiciones generales en las que se debe aplicar las pruebas para obtener los objetivos propuestos. El objetivo de los casos de prueba es forzar al máximo el sistema en los puntos críticos para encontrar fallos y detectar defectos.

Las pruebas se deben aplicar durante todo el ciclo de vida del software e invariablemente se le debe dedicar una gran parte del esfuerzo total del desarrollo. Se deben planificar correctamente desde el inicio y establecer qué hacer, cómo hacer, quién va a hacer y en qué condiciones hacer las comprobaciones. Es beneficioso que los desarrolladores prueben su producto pero que no falte la mano de terceras personas que no intervinieron en el proyecto directamente ya que así se detecta mayor cantidad de fallas.(PRESSMAN, 2005)

Se debe escoger los tipos de prueba que se adapten mejor al sistema que se va a probar. Para esto se debe tener en cuenta el lenguaje de programación, el proceso de desarrollo, las características de los desarrolladores, el tipo de funcionalidad que se implementa, la plataforma en que se ejecutan los procesos, los errores más importantes, si la aplicación es de escritorio o web, si realiza conexiones a bases de datos entre otras observaciones. Los dos grandes grupos de pruebas unitarias existentes son las pruebas de Caja Negra y las pruebas de Caja Blanca.

4.3.1 Pruebas unitarias

Las pruebas unitarias permiten probar, como su nombre lo indica, cada unidad independiente del software. Actúan esencialmente sobre el código fuente y sobre los elementos básicos de la interfaz del módulo.

Según Pressman (PRESSMAN, 2005) los casos de prueba que se generan durante las pruebas de unidad deben estar encaminados a verificar los siguientes elementos:

Capítulo 3: Implementación y Prueba

- Interfaz: “Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada”.
- Estructuras de datos locales: “Se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo”.
- Condiciones límites: “Se prueban las condiciones límites para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento”.
- Caminos independientes: “Se ejercitan todos los caminos independientes (caminos básicos) de la estructura de control con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez”.
- Caminos de manejo de errores: “... se prueban todos los caminos de manejo de errores”.

Pruebas de Caja Blanca

Las pruebas de Caja Blanca se nombran de esta forma porque a diferencia de las pruebas de Caja Negra que actúan sobre la interfaz, estas revisan la parte interna del software, específicamente sobre el código fuente. Se basan en el examen minucioso de los detalles procedimentales. Se comprueban los caminos lógicos del sistema generando casos de prueba que ejerciten las estructuras condicionales y los bucles. Es por esto que las Pruebas Unitarias se basan en las Técnicas de Pruebas de Caja Blanca.

Existen varios métodos que analizan diferentes partes del programa y se complementan entre sí para garantizar la calidad del sistema. La técnica del camino básico se utiliza para comprobar la complejidad lógica de un diseño procedimental. Permite diseñar casos de prueba para cubrir todas las sentencias de un programa a partir de la obtención de un conjunto de caminos independientes. La complejidad ciclomática, como resultado fundamental de estas pruebas, acota la cantidad mínima de casos de prueba que se deben ejecutar.

La Prueba de las Condiciones Es un método que se encamina hacia la ejercitación de las condiciones. Se basa en el principio de que si un conjunto de casos de prueba es capaz de ejercitar todas las condiciones contenidas en un bloque de código, este mismo conjunto serviría para encontrar más errores en el programa que no tengan que ver directamente con las condiciones. La Prueba del Flujo de Datos verifica la validez en el uso de las variables para manipular los datos de la aplicación.

Capítulo 3: Implementación y Prueba

Selecciona los casos de prueba atendiendo a las definiciones y los usos de las variables. El procedimiento indica que se debe encontrar las sentencias donde se define cada variable y las sentencias donde se hace uso de las mismas. Luego se encuentran las —cadenas de definición - uso— que representan el ciclo de vida de las variables y se diseñan casos de prueba que las ejecuten en su totalidad. La Prueba de los Bucles se centra en la validez de las estructuras cíclicas o bucles. El objetivo es probar el comportamiento de estas estructuras en sus valores límites de iteración. Los bucles se clasifican en cuatro tipos: Bucles simples, Bucles anidados, Bucles concatenados y Bucles no estructurados. Aunque la esencia es la misma, cada tipo se prueba de forma diferente. De lo anterior pudiera pensarse que las pruebas de Caja Blanca logran enmendar todos los errores del programa. La desventaja de estas es entonces de tipo logística ya que resulta imposible abarcar todo el código fuente de un sistema medianamente grande. El tiempo necesario para realizarlas sería considerable y se torna compleja su aplicación sobre algoritmos críticos.

Pruebas de Caja Negra

Las Pruebas de Caja Negra deben su nombre a los elementos que estas revisan y las condiciones en que se hace la revisión. Estas se basan en los requerimientos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Técnica de la Partición Equivalente: Es un método de prueba de caja negra que consiste en analizar los valores de entrada de un programa para diseñar casos de prueba con datos que representen clases válidas e inválidas. Los datos tienen tres tipos de condiciones de entrada:

- Condición de entrada lógica (en lo adelante CEL).
- Condición de entrada de rango (en lo adelante CER).
- Condición de entrada de valor (en lo adelante CEV).

Por ejemplo, un dato de entrada en un formulario correspondiente al nombre de un medicamento que sea dañino para una persona, tiene como condiciones de entrada:

CEL: El nombre del medicamento puede o no aparecer.

CEV: El nombre del medicamento si aparece es un alfanumérico sin signos.

Entonces, para probar la validez de asimilación de la entrada de dicho campo en el programa se pueden diseñar casos de prueba donde las entradas para dicho campo serían:

Dejar el campo vacío.

Capítulo 3: Implementación y Prueba

Ingresar —Dipirona|| .

Ingresar —Dipirona@|| .

Análisis de Valores Límites: Este método complementa al de Partición Equivalente. La diferencia está en que este selecciona los casos de prueba con valores que se encuentran en los límites de los rangos de validez. Por ejemplo, si un campo de entrada debe poseer valores comprendidos entre A y B. Se deben diseñar casos de prueba como sigue:

Que el valor sea mayor que A y lo más próximo posible a este.

Que el valor sea menor que B y lo más próximo posible a este.

Que el valor sea menor que A y lo más próximo posible a este.

Que el valor sea mayor que B y lo más próximo posible a este.

Que el valor sea igual que A.

Que el valor sea igual que B.

Las pruebas de caja negra son importantes a la hora de medir el grado de cumplimiento de los requerimientos solicitados por el cliente y se aplican sobre la interfaz de la aplicación observando las respuestas del sistema antes determinadas acciones y los datos de salida para determinados datos de entrada. (PRESSMAN, 2005)

Al analizar los métodos de prueba de Caja Blanca y Caja Negra se llega a la conclusión de que es más factible aplicar las pruebas de Caja Negra para comprobar la validez en las respuestas del programa ante las acciones del usuario y la calidad de las salidas en dependencia de las entradas. Para comprobar la eficiencia de la codificación es más adecuado emplear las pruebas de Caja Blanca, más específicamente, la Técnica del Camino Básico. Para probar la validez de la solución propuesta se decidió probar los elementos fundamentales de la interfaz de usuario mediante casos de prueba de Caja Negra así como la validez de las salidas del sistema en dependencia de las entradas del usuario. Los algoritmos más importantes o más complejos se comprobarán mediante la técnica del Camino Básico.

4.3.2 Prueba de aceptación

Las pruebas de aceptación son realizadas principalmente por los usuarios con el apoyo del equipo de trabajo. Las pruebas de aceptación se realizan con el objetivo de validar la solución a través de un

Capítulo 3: Implementación y Prueba

encuentro con el cliente, el cual comprueba que el sistema cumple con el funcionamiento esperado y da muestra de su conformidad en el documento oficial Carta de aceptación del cliente donde quedan plasmados los resultados. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales. (Ver anexo 3)

4.3.3 Prueba de sistema

Las pruebas del sistema examinan qué tan bien el sistema cumple con los requerimientos de la organización y su utilidad, seguridad y desempeño. Son las pruebas que se realizan para determinar el correcto funcionamiento de un sistema y su cumplimiento contra las especificaciones del producto. También se realizan estas pruebas a la documentación del sistema. (Ver anexo 4)

4.4 Diseño y ejecución de los Casos de Prueba de Caja Negra

4.4.1 Caso de Prueba Administrar Plantilla de Formulario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Número	campo de texto	Si	Valor alfanumérico de 1 a 8 caracteres solo acepta cadenas de números del 0-9.
2	Subnúmero	campo de texto	Si	Valor alfanumérico de 1 o 2 caracteres. Solo acepta cadenas de números del 0-9.

Tabla 3. Descripción de las variables del caso de prueba Administrar Plantilla de Formulario.

Escenario	Descripción	Número	Subnúmero	Respuesta del sistema	Flujo central
EC 1.1	Buscar la	V	V	Se establece	1-Selecciona la opción de

Capítulo 3: Implementación y Prueba

Buscar plantilla de formulario.	planilla de formulario satisfactoriam ente en la lista de plantillas existentes.	003	03	la búsqueda de forma correcta.	buscar plantilla de formulario. 2-Introduce el criterio de búsqueda. 3-Se Selecciona el botón aceptar.
		V	V		
		005	51		
EC 1.2 Buscar plantilla de formulario con campos en blanco.	Realizar la búsqueda dejando campos en blanco.	V	V	Se establece la búsqueda de forma correcta.	1-Selecciona la opción "Buscar plantilla de formulario". 2-Se Introduce los datos y se dejan campos en blanco. 3-Se selecciona el botón aceptar.
		NA	51		
		V	V		
		005	NA		
EC 1.4 Cancelar la ejecución de la opción "Buscar plantilla en formulario".	Se selecciona la opción de cancelar para evitar buscar una planilla de formulario.	V	V	Se realiza la operación de cancelación satisfactoria mente.	1-Selecciona la opción "Buscar plantilla de formulario". 2-Selecciona el botón cancelar.
		NA	NA		

Tabla 4. Sección a probar Buscar Plantilla de Formulario.

Capítulo 3: Implementación y Prueba

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar plantilla de formulario.	Se elimina la planilla de formulario satisfactoriamente de la lista de plantillas de formularios existentes.	Se elimina la plantilla satisfactoriamente.	1-Selecciona la plantilla de formulario de la lista de plantillas existentes. 2-Selecciona la opción "Eliminar plantilla de formulario". 3-Selecciona el botón aceptar.
EC 1.2 Cancelar la ejecución de la opción "Eliminar plantilla en formulario".	Se selecciona la opción de cancelar para evitar eliminar planilla de formulario seleccionada.	Se realiza la operación de cancelación satisfactoriamente.	1-Selecciona la plantilla de formulario de la lista de plantillas existentes. 2-Selecciona la opción "Eliminar plantilla de formulario". 3-Selecciona el botón cancelar.

Tabla 5. Sección a probar Eliminar Plantilla de Formulario.

Capítulo 3: Implementación y Prueba

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Listar plantilla de formulario.	Se listan las planillas de formulario satisfactoriamente.	Se establece la conexión de forma correcta mostrando el listado de plantillas almacenados en SIGELite.	1-Se carga la aplicación. 2- Se listan las plantillas de formulario existentes.

Tabla 6. Sección a probar Listar Plantilla de Formulario.

4.4.2 Caso de Prueba Transferir Recurso

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Importar recurso.	Se importar recurso satisfactoriamente.	Se establece la importación del recurso seleccionado de forma correcta.	1-Selecciona la opción "Importar recurso". 2-Selecciona el recurso. 3-Selecciona el botón aceptar.
EC 1.2 Cancelar la ejecución de la opción "Importar recurso".	Se selecciona la opción de cancelar para evitar importar recurso.	Se realiza la operación de cancelación satisfactoriamente.	1-Selecciona la opción "Importar recurso". 2-Selecciona el recurso. 3-Seleccionar el botón cancelar.

Tabla 7. Sección a probar Importar recurso.

Capítulo 3: Implementación y Prueba

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Exportar recurso.	Se exporta el recurso seleccionado satisfactoriamente.	Se exporta el recurso satisfactoriamente.	1-Selecciona el recurso de la lista de recursos existentes. 2-Selecciona la opción "Exportar recurso". 3- Selecciona el botón aceptar.
EC 1.2 Cancelar la ejecución de la opción "Exportar recurso".	Se selecciona la opción de cancelar para evitar exportar recurso.	Se realiza la operación de cancelación satisfactoriamente.	1-Selecciona el recurso de la lista de recursos existentes. 2-Selecciona la opción "Exportar recurso". 3-Selecciona el botón de cancelar y así evita que se exporte el recurso seleccionado.

Tabla 8. Sección a probar Exportar recurso.

4.5 Diseño y ejecución del método Caja Blanca

En esta sección se escogerán algunos algoritmos o funciones importantes en la implementación del sistema para diseñar los casos de prueba mediante la Técnica del Camino Básico. El proceso se realizará de forma manual y automatizada. Primeramente se aplicará la técnica de forma manual obteniendo el grafo de flujo correspondiente al código. Luego se calculará la complejidad ciclomática

Capítulo 3: Implementación y Prueba

mediante las tres variantes explicadas. Por último se obtiene un conjunto de caminos independientes para el grafo obtenido.

4.5.1 Técnica del camino básico del método de Caja Blanca para el método Conectar

El objetivo del método conectar enviar las disímiles peticiones especificadas por el usuario al servidor, por ello al método se le pasa por parámetro una dirección http y devuelve como resultado la respuesta recibida por el servidor en un objeto de tipo JSON. A continuación, en la siguiente figura se aplica la Técnica del Camino Básico de forma manual para dicha función.

Capítulo 3: Implementación y Prueba

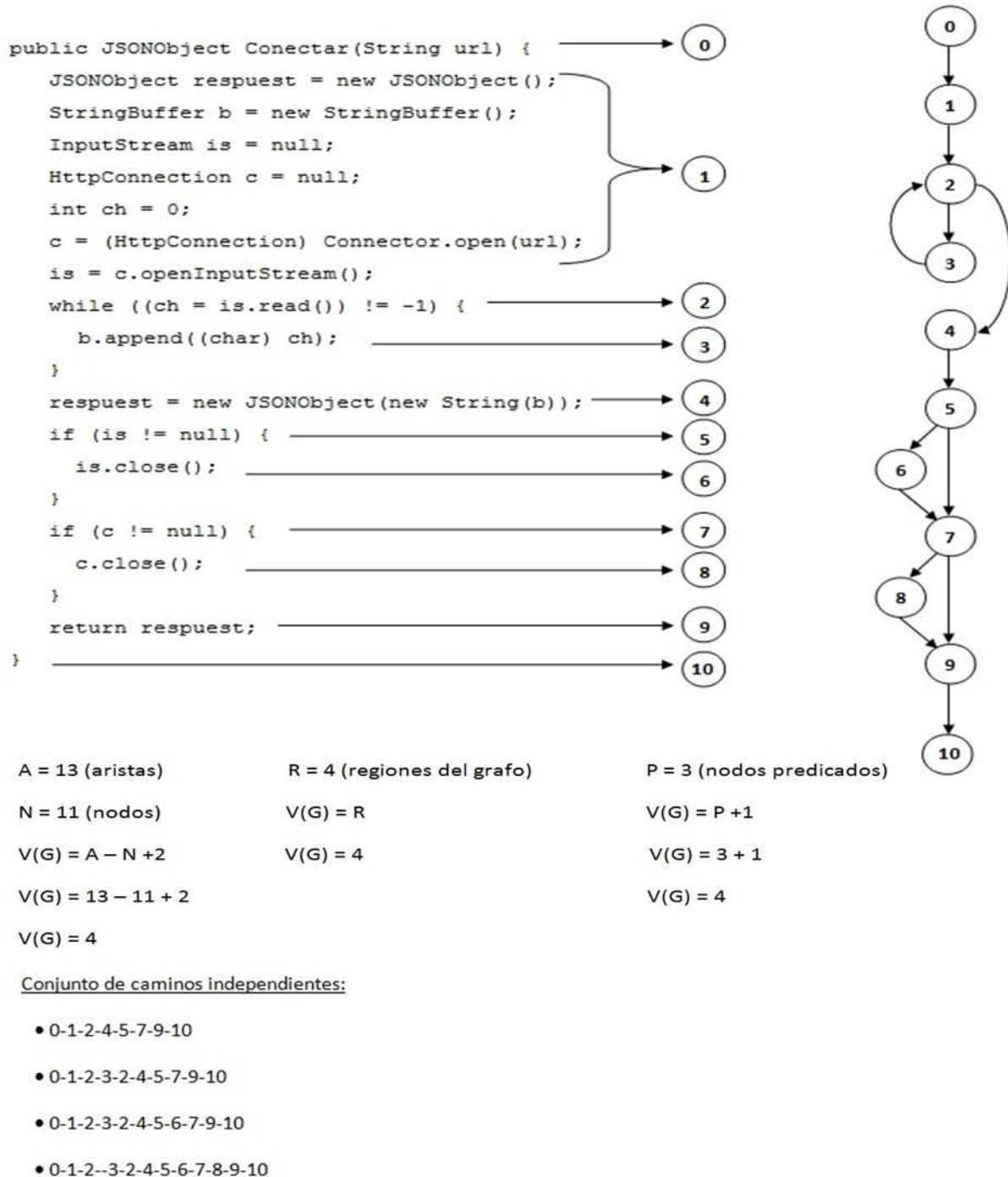


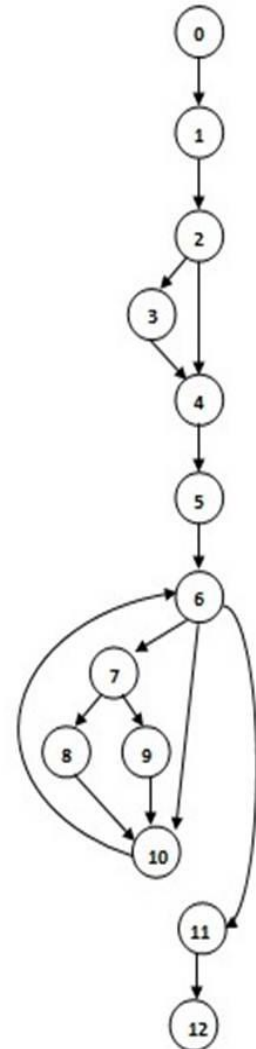
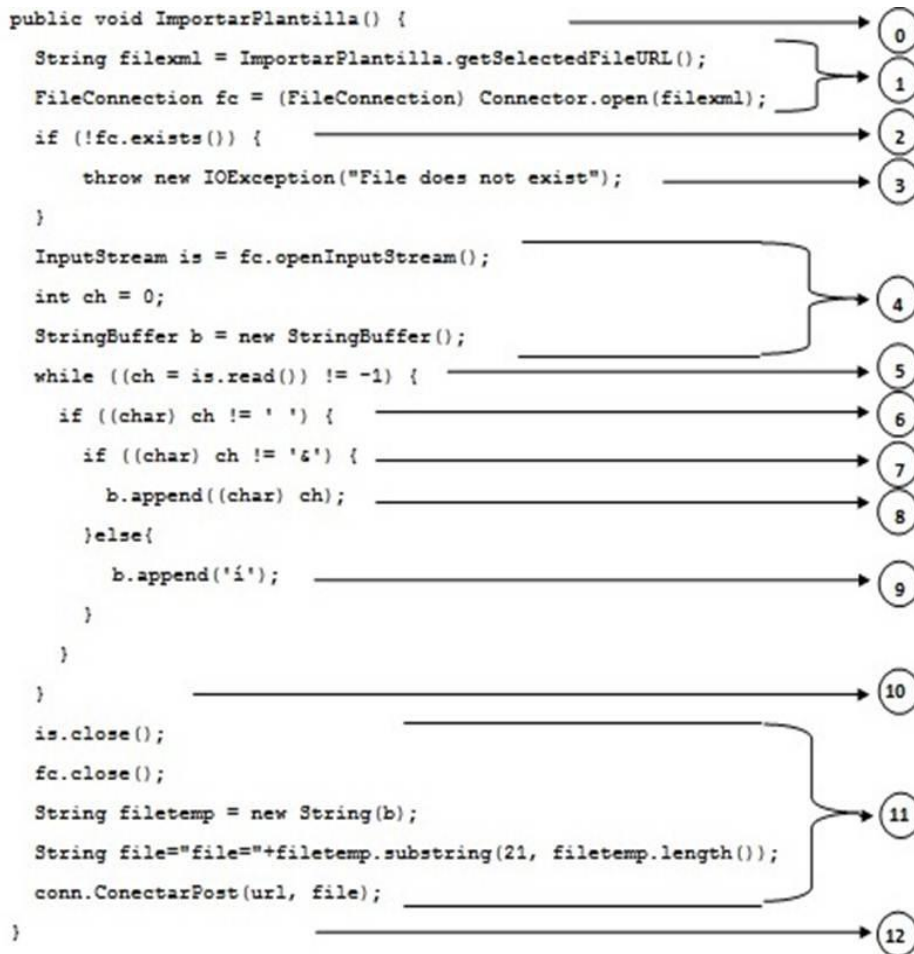
Figura 9. Aplicación de la Técnica del Camino Básico al método Conectar.

Capítulo 3: Implementación y Prueba

4.5.2 Técnica del camino básico del método de Caja Blanca para el método Importar Plantilla

El objetivo del método importar plantilla es realizar la importación de plantillas no existentes en el servidor. A continuación, en la siguiente figura se aplica la Técnica del Camino Básico de forma manual para dicha función.

Capítulo 3: Implementación y Prueba



A = 16 (aristas) R = 5 (regiones del grafo) P = 4 (nodos predicados)

N = 13 (nodos) V(G) = R V(G) = P + 1

V(G) = A - N + 2 V(G) = 5 V(G) = 4 + 1

V(G) = 16 - 13 + 2 V(G) = 5 V(G) = 5

V(G) = 5

Conjunto de caminos independientes:

- 0-1-2-4-5-6-11-12
- 0-1-2-3-4-5-6-11-12
- 0-1-2-3-4-5-6-10-6-11-12
- 0-1-2-3-4-5-6-7-8-10-6-11-12
- 0-1-2-3-4-5-6-7-9-10-6-11-12

Figura 10. Aplicación de la Técnica del Camino Básico al método Importar Plantilla.

Capítulo 3: Implementación y Prueba

Luego de aplicar las pruebas funcionales se arrojan diferentes resultados, pero en esencia, se busca mayor eficacia y eficiencia a la hora de encontrar defectos, verificando la calidad, de forma tal que se minimicen tiempos, costos, y se obtenga un resultado satisfactorio. El sistema desarrollado después de aplicarle las pruebas arrojó 3 no conformidades significativas, las cuales fueron resueltas, culminando el presente trabajo con resultados satisfactorios.

4.6 Interfaces Principales de la Aplicación

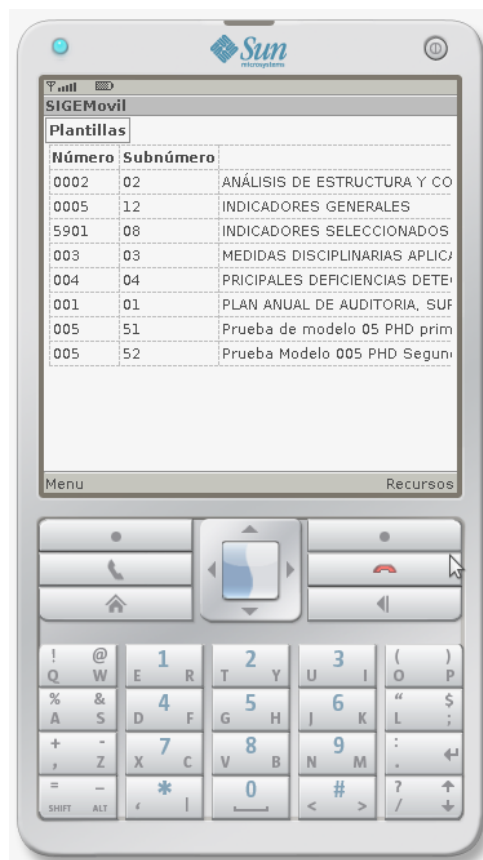


Figura 11. Interfaz Listar Plantilla.

Capítulo 3: Implementación y Prueba

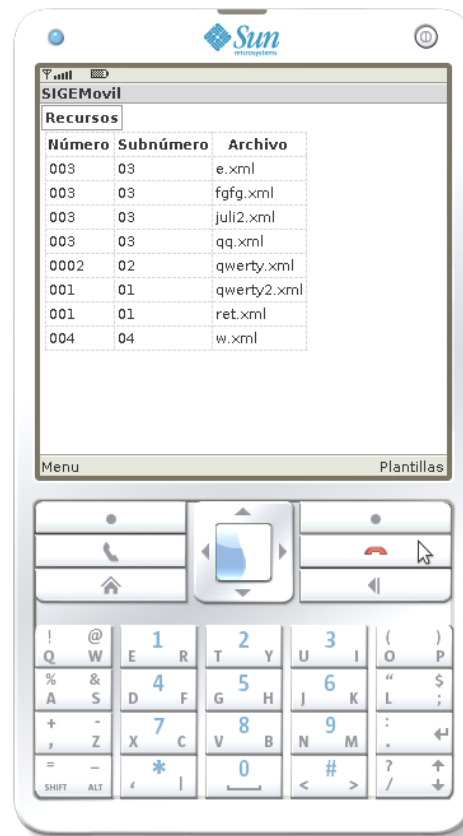


Figura 12. Interfaz Listar Recurso.

Conclusiones del Capítulo 3

Como resultado de este capítulo se obtuvieron los diagramas de componentes y de esta forma se realizó la implementación del sistema en términos de componentes, ofreciendo solución a los requisitos especificados en el capítulo anterior. Se brindó una descripción del código fuente implementado así como de los estilos de codificación y nomenclatura. Finalmente se realizaron los casos de pruebas, luego de la implementación del sistema, para validar la completitud de los requerimientos, obteniéndose 2 no conformidades significativas, las cuales fueron resueltas, culminando el presente trabajo con resultados satisfactorios.

Conclusiones

CONCLUSIONES

Una vez culminado el trabajo se arriba a las siguientes conclusiones:

- Se lograron identificar las funcionalidades principales de los sistemas para capturar datos en dispositivos móviles.
- Se diseñó la Interfaz Web para la Captura de Datos en dispositivos móviles.
- Se desarrolló una interfaz web para la captura de datos en el Sistema Integrado de Gestión Estadística (SIGE) usando herramientas, lenguajes y tecnologías libres a tono con las políticas de estándares abiertos que propone el país para garantizar la soberanía tecnológica.
- Se validó la solución propuesta a través de las pruebas de caja negra.

Recomendaciones

RECOMENDACIONES

Después de haber concluido con éxito los objetivos trazados al comienzo del desarrollo de este trabajo y que la Interfaz Web de Captura de Datos en Dispositivos Móviles en el Sistema Integrado de Gestión Estadística, la cual se encuentra en su primera versión se plantean las siguientes recomendaciones:

- Analizar después de un tiempo el impacto que ha tenido el sistema en los clientes.
- Desarrollar una aplicación en java para la captura de datos que permita el trabajo local en el dispositivo móvil.

Bibliografía

BIBLIOGRAFÍA

1. Departamento de Sistemas Informáticos y Computación. Introducción a Herramientas CASE y System Architect. [En línea] [Citado el: 8 de Noviembre de 2011.] http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.
2. Lenguajes de Programación. [En línea] 2009. [Citado el: 7 de Noviembre de 2011.] <http://www.lenguajes-de-programacion.com/programacion-web.shtml>.
3. PHP Manual. [En línea] 11 de Noviembre de 2011. [Citado el: 11 de Noviembre de 2011.] <http://php.net/manual/es/intro-what-is.php>.
4. Ángel Álvarez, Miguel. Desarrolloweb. [En línea] 9 de Mayo de 2001. [Citado el: 6 de Noviembre de 2011.] <http://www.desarrolloweb.com/articulos/392.php>.
5. González Fernández, Frank, y otros, y otros. INFORMATIZACIÓN DE LOS PROCESOS DE GESTIÓN ESTADÍSTICA EN CUBA. Cuba: UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2011.
6. Introducción a J2ME, 2010, http://java.ciberaula.com/articulo/introduccion_j2me.
7. Juan, F.J.M., Guía de construcción de software en JAVA con Patrones de Diseño, in Departamento de Ingeniería Técnica en Informática. 2000, Universidad de Oviedo: Oviedo.
8. Pedro Álvarez, J.Á.B., Conceptos y estándares de arquitecturas orientadas a servicios Web. 2006/2007, Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza.
9. Molina, J.G., El Lenguaje Unificado de Modelado, UML. 2004-2005, Departamento de Informática y Sistemas, Universidad de Murcia.
10. Martínez, E., La evolución de la telefonía móvil (La guerra de los celulares). in Revista RED. 2001.
11. Juzgado, P.D.B., Java 2 Micro Edition Soporte Bluetooth.
12. Lea, Doug. 2001. Programación Concurrente en Java. Principios y Patrones de Diseño.
13. Larman, Craig. 1999. UML y Patrones: Introducción al análisis y programación orientada a objetos.

Bibliografía

14. José H. Canós, Patricio Letelier, Ma. Carmen Penadés. Metodologías ágiles en el desarrollo del software. [En línea] [Citado el: 20 de febrero de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
15. WebExperto. WebExperto. [En línea] 8 de julio de 2006. [Citado el: 4 de marzo de 2010.] <http://www.webexperto.com/downloads/262/openwave-phone-simulator-7/>.
16. Letelier, P. Metodologías ágiles para el desarrollo de software. 2004.
17. Cameron Moll. Mobile Web Design. 2007.
18. Kim Topley. J2ME in a Nutshell. 2002.
19. Sergio Gálvez Rojas, Lucas Ortega Díaz. JAVA A TOPE: J2ME (JAVA 2MICRO EDITION). 2003.
20. Pedro Daniel Borches Juzgado. Java 2 Micro Edition: Soporte Bluetooth. 2004.
21. Guillermo Diez-Andino Sancho, Rosa M^a García Rioja. Java 2 Micro Edition. Un primer Vistazo. 2002.
22. Dr. Jesús Soto Carrión. DESARROLLO DE APLICACIONES MÓVILES. 2008.
23. Félix Gómez Mármol. Desarrollo de Aplicaciones Java para Dispositivos Móviles J2ME. 2011.
24. Arphean. Tutorial para aplicaciones móviles J2ME con NetBeans y Mobility Pack. 2006.
25. Alberto García Serrano. Programación de juegos para móviles con J2ME. 2005.
26. Antonio García Cabot, Eva García López. Primeros pasos en la programación para Móviles. 2007.
27. Herbert Schildt. JAVA 2 Manual de referencia. 2003.
28. Steven Holzner. La biblia de Java 2. 2003.

Referencias Bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

ABOUT JAVA ME. 2005, Disponible en: <http://www.oracle.com/technetwork/java/javame/about-java-me-395899.html>.

RAÚL MONFERRER AGUT. Especificación de Requisitos Software según el estándar IEEE 830. 2008, Disponible en: http://www.google.com.cu/url?sa=t&rct=j&q=Especificaci%C3%B3n+de+Requisitos+Software+seg%C3%BAAn+el+est%C3%A1ndar+IEEE+830&source=web&cd=2&ved=0CFUQFjAB&url=http%3A%2F%2Fsi.ml.googlecode.com%2Ffiles%2FERS.pdf&ei=40vET_qWN8zogAfGybS_CQ&usg=AFQjCNFj2Xj8OMaKkue4Qcg5SHI5w4TYUg&cad=rja.

RAFAEL BARZANALLANA. Apuntes. Ingeniería del software. Sistemas Informaticos. 2006, Disponible en: <http://www.um.es/docencia/barzana/LAGP/lagp02.pdf>.

TIM BOUDREAU. NetBeans: the definitive guide. 2002, Disponible en: books.google.com.

CECILIA MILIÁN DELGADO. Componente para la captación de encuestas económicas. 2011,

ESTEBAN TRIGOS GARCÍA. PHP 2000, Disponible en: <http://dialnet.unirioja.es/servlet/libro?codigo=242312>.

DENYS BUEDO HIDALGO. *Sirio Mobile 2.0, plataforma para la gestión de juegos multiusuario desde dispositivos móviles vía GPRS*. 2008, Disponible en: <http://rcci.uci.cu/index.php/rcci/article/view/35/34>.

KAREN. *openup-como-alternativa-metodolgica*. 2008, Disponible en: kasyles.blogspot.com.

BRIAN W KERNIGHAN, RITCHIE, DENNIS M. Y GÓMEZ MUÑOZ, NÉSTOR. Lenguajes de programación. 1991, Disponible en: books.google.com.

CRAIG LARMAN. UML y Patrones. 2003, Disponible en: <http://archivos5.movistar.cl/comunidad/Book-UMLyPatrones-CraigLarman/UMLyPatrones-CraigLarman.rar>.

IVÁN LASSO. Qué es el código fuente. 2008, Disponible en: <http://www.proyectoautodidacta.com/comics/que-es-el-codigo-fuente/>.

LUCIANO. Entornos de Desarrollo Integrados para Java. 2008,

Referencias Bibliográficas

CRISTINA MOLINA. Resumen de tecnología. 2005 Disponible en: http://www.bnamericas.com/news/tecnologia/Resumen_de_tecnologia117.

ROBERTO CANALES MORA. Modelado UML con Visual Paradigm. 2007, Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm>.

2009, Disponible en: <http://www.google.es/cse?cx=partner-pub-0219323732506342%3Ab1dcbirdhfm&ie=ISO-8859-1&q=presentacion+del+modelo+de+objeto+de+dominio&sa=Buscar&siteurl=www.eumed.net%2Flibros%2F2009c%2F583%2F&ref=#gsc.tab=0&gsc.q=modelo%20de%20objeto%20de%20dominio>.

MARCELO CLAUDIO PERISSÉ. HERRAMIENTAS CASE. 2001, Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.

ROGER PRESSMAN. Ingeniería del Software: Un Enfoque Práctico, Roger Pressman (Sexta Edición). 2005, Disponible en: <http://www.intercambiosvirtuales.org/libros-manuales/ingenieria-del-software-un-enfoque-practico-roger-pressman-sexta-edicion>.

ROSARIO. MÓDULO DE CAPTURA MÓVIL DE DATOS. 2000, Disponible en: www.knowtech.ws/Brochures/QBMobile.pdf.

JAMES RUMBAUGH. El lenguaje Unificado de Modelado, Manual de Referencia. 2000 Disponible en: <http://www.intercambiosvirtuales.org/libros-manuales/el-lenguaje-unificado-de-modelado-uml-de-james-rumbaugh-manual-de-referencia>.

JAMES RUMBAUGH. El Proceso Unificado de Desarrollo de Software. 2000, Disponible en: <http://es.scribd.com/doc/51515328/El-Proceso-Unificado-de-Desarrollo-de-Software-Jacobson-Booch-%E2%80%93-Rumbaugh-capitulo-6>.

Tecnología y Synergix. 2009, Disponible en: <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional>.

Glosario de Términos

GLOSARIO DE TÉRMINOS

A

API (Application Programming Interface): Conjunto de rutinas, protocolos, y de herramientas para construir aplicaciones de software. Un buen API hace más fácil desarrollar un programa, proporcionando todos los bloques de la construcción. El programador luego pone esos bloques juntos.

C

CLDC (Connected, Limited Device Configuration): Es una configuración diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada como teléfonos móviles y asistentes personales.

H

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

HTTP (Hypertext Transfer Protocol): Protocolo de Transferencia de Hipertexto utilizado en la WWW para transmitir las páginas de información entre el programa navegador y el servidor. Se destaca que el HTTP seguro es un protocolo HTTP mejorado con funciones de seguridad con clave simétrica.

I

IDE (Integrated Development Environment): Un entorno de desarrollo integrado es un programa compuesto por un conjunto de herramientas para un programador.

Internet: Es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, posee alcance mundial, constituye una red mundial de comunicación.

J

J2EE (Java 2 Enterprise Edition): Grupo de especificaciones diseñadas por la Sun Microsystems de Java que permiten la creación de aplicaciones empresariales, brindando múltiples funcionalidades.

J2ME (Java 2 Micro Edition): Versión desarrollada por la Sun Microsystems de Java, destinada a dispositivos de recursos limitados como PDAs, teléfonos móviles.

Glosario de Términos

M

MD5: En criptografía, MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits.

Midlets: Son juegos y aplicaciones que corren en un teléfono móvil.

MIDP (Mobile Information Device Profile): Es el perfil para dispositivos de información móviles que combina con la configuración CLDC para proporcionar un entorno de ejecución para dispositivos móviles.

Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

P

PDA: Un PDA (Personal Digital Assistant o Ayudante personal digital) es una computadora de mano originalmente diseñada como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura.

Plataforma: Se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo.

U

UML: Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

URL: Uniform Resource Locator, es decir, localizador uniforme de recurso y se refiere a la dirección única que identifica a una página web en Internet.

W

Web: Abreviatura de World Wide Web (WWW); interfaz de comunicación en la Internet, que hace uso de enlaces de hipertexto en el interior de una misma página, o entre distintas páginas.

X

XML: Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Se propone como un estándar para

Glosario de Términos

el intercambio de información estructurada entre diferentes plataformas. XML separa la estructura del contenido y permite el desarrollo de vocabularios modulares. Se trata de un formato abierto.

Anexos

ANEXOS

Anexo 1: Emuladores del NetBeans 6.9



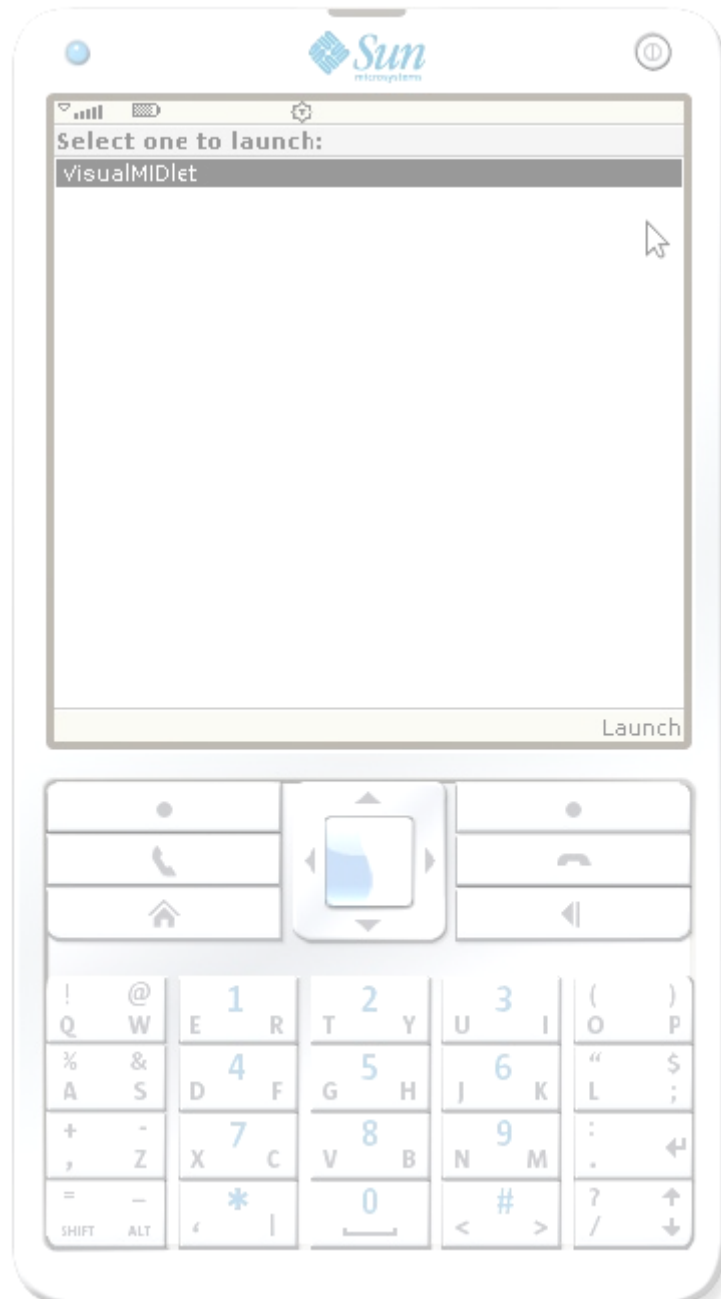
Emulador 1

Anexos



Emulador 2

Anexos



Emulador 3

Anexos



Emulador 4

Anexos

Anexo 2: Diagrama de secuencia del CU Administrar Recurso.

Diagrama de secuencia del CU Administrar Recurso sección Buscar Recurso.

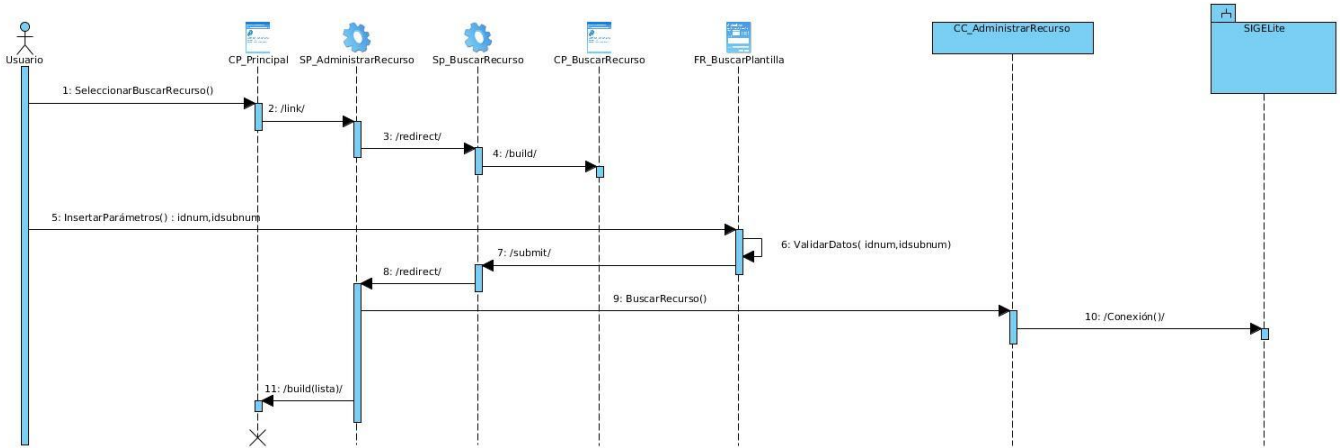
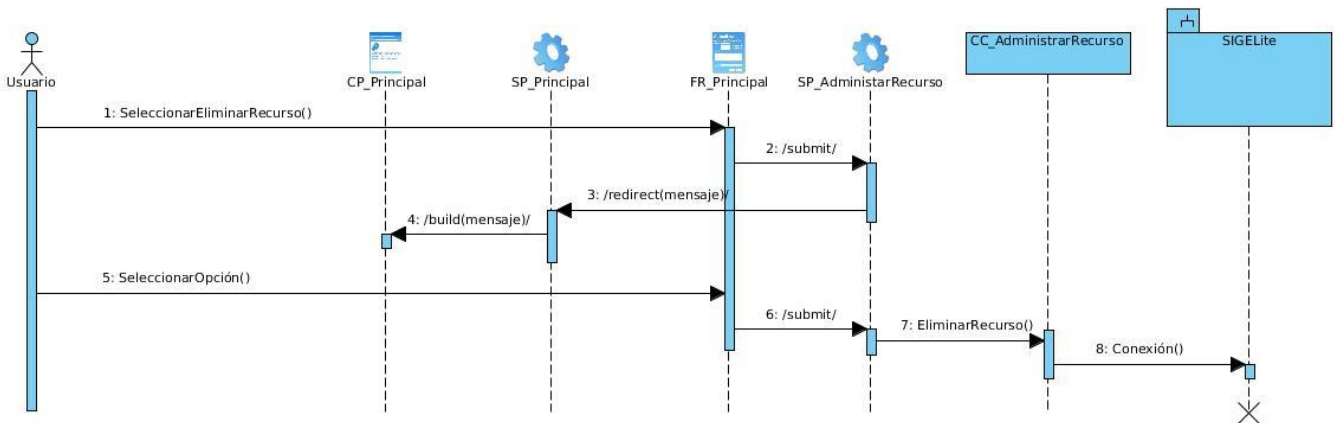



Diagrama de secuencia del CU Administrar Recurso sección Eliminar Recurso.



Anexos

Anexo 3: Carta de Aceptación del Cliente.

	<p>Vicerrectoría de Formación Dirección de Formación del Profesional Departamento de Práctica Profesional</p>
---	---

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Interfaz Web de captura de datos en dispositivos móviles en el Sistema Integrado de Gestión Estadística**, fue realizado en **Centro de Tecnologías de Gestión de Datos (DATEC)** para la **Oficina Nacional de Estadística e Informática (ONEI)**. Esta última considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente
 Parcialmente en un 85 %, debido a la limitante del trabajo con redes inalámbricas en nuestro país.

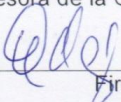
Los resultados de este Trabajo de Diploma le reportan a esta oficina los beneficios siguientes (cuantificar):

1. Aumento en la productividad y la eficiencia de los trabajadores de la Oficina Nacional de Estadística e Información (ONEI).
2. Ahorro de recursos humanos y materiales.
3. Aumento de la calidad y oportunidad de la información.

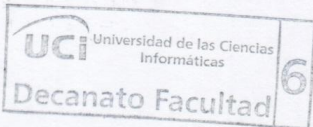
Concluyendo, pensamos que es una línea de trabajo muy útil por los beneficios que reporta, sobre la que hay que investigar además otras posibilidades de obtener el recurso y depositar la información captada para poder hacer un uso generalizado de la misma.

Y para que así conste, se firma la presente a los 21 días del mes de mayo del año 2012.

Elena Leonila Fernández García
Asesora de la ONEI


Firma

Cuño



Anexos

Anexo 4: Prueba del sistema

Acta de liberación interna del sistema interfaz web de captura de datos en dispositivos móviles en el Sistema Ligerero de Captura de Datos.

Anexo 5: Acta de Liberación de SIGE.



Firma válida
Digitally signed by Tayché Capote García
Date: 2012.02.16 15:08:00
Reason: I am the author of this document
Cuba



Acta de Liberación de Productos Software

Fecha de emisión del acta: 02/02/2012

Emitida a favor de: SIGE

Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
Aplicación	1.8	0 NC	3	Pruebas funcionales	02/02/2012
Manual de usuario MGC	1.8	0 NC	2	Evaluación estática	01/02/2012
Manual de usuario MDF	1.8	0 NC	2	Evaluación estática	01/02/2012
Manual de usuario MED	1.8	0 NC	2	Evaluación estática	01/02/2012

Yaneida Rondón Hernández
Responsable Calisoft

Frank González Fernández
Responsable Proyecto

Tayché Capote García
Jefe del Departamento de Pruebas

Anexos

Anexo 6: Acta de Liberación de SIGELite 1.0.



Acta de Liberación de Productos Software

Fecha de emisión del acta: 27/02/2012.

Emitida a favor de: SIGELITE

Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
SIGE_Ligero	1.0	0	3	Pruebas Funcionales	23/02/2012
Manual de Usuario SIGE_Ligero	1.0	0	2	Revisiones técnicas y Correspondencia con otro artefacto.	13/03/2012

Ing. Loraima Maning Aldana
Responsable Calisoft

Ing. Claudia Nuñez Sanz
Responsable Proyecto

MsC. Tayché Capote García
Jefe del Departamento de Pruebas de Software

Anexos

Anexo 7: Acta de Liberación de SIGELite 1.1.



Acta de Liberación de Productos Software

Fecha de emisión del acta: 30/05/2012.

Emitida a favor de: Sistema Integrado de Gestión Estadísticas (SIGE).

Datos del producto:

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
SIGE 1.8.1 y SIGE Ligero	1.8.1	0 NC	2	Funcionales	30/05/2012
Manual de usuario MDF	1.8.1	0 NC	1	Evaluación estática	30/05/2012
Manual de usuario MED	1.8.1	0 NC	1	Evaluación estática	30/05/2012
Manual de usuario MGC	1.8.1	0 NC	1	Evaluación estática	30/05/2012
Manual de usuario SIGE Ligero	1.1	0 NC	1	Evaluación estática	30/05/2012

Yailin Fundora Carrasco
Responsable Calisoft

Yanet Rosales Morales
Responsable Proyecto

Tayché Capote García
Jefa DPSW

Anexos

Anexo 8: Estudio de factibilidad.

Objetivo

El objetivo del presente estudio es realizar un perfil de factibilidad entorno al desarrollo de una Interfaz Web de captura de datos en dispositivos móviles en el Sistema Ligero de Captura de Datos (SIGELite).

1.2 Pertinencia

Decir antes de todo que SIGE, diseña plantilla que SIGELite utilizan para capturar la información.

Pasos para la captura de información en SIGE:

- Se aplica la plantilla de encuesta: El encuestador le aplica la plantilla de encuesta a una persona.
- Se trasladan las encuestas levantadas hacia el lugar donde se van digitalizar.
- El digitador captura en el sistema, los datos de las encuestas recibidas.

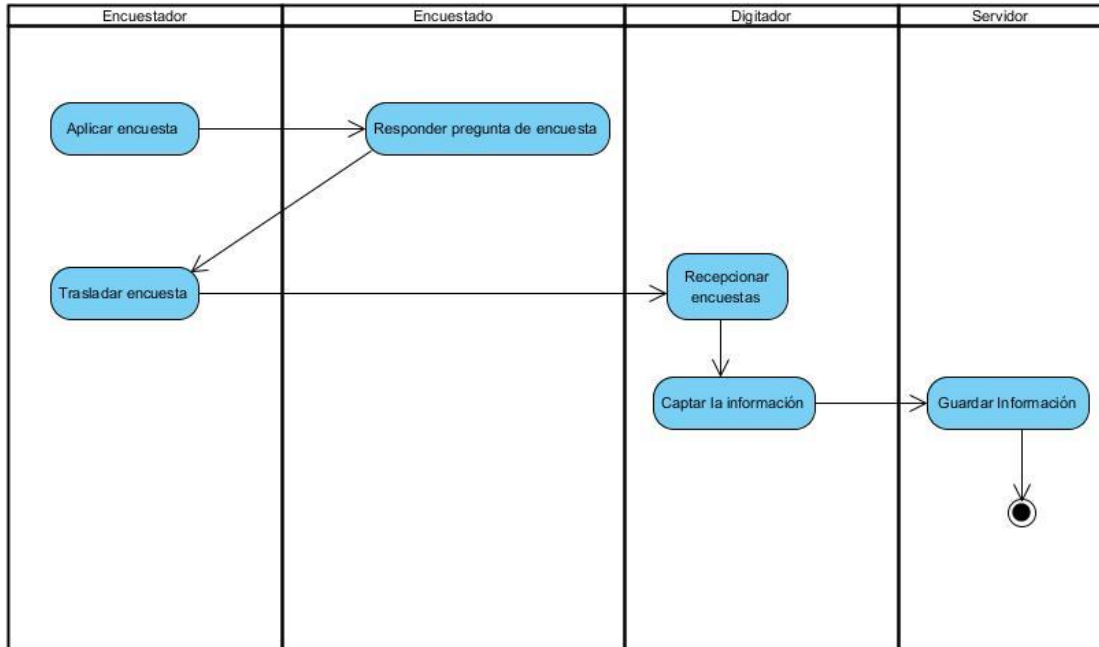
Actualmente el manejo de la información en SIGE presenta algunos riesgos que a grandes rasgos pueden ocasionar la pérdida de la información capturada, entre ellos se tienen los siguientes:

- El encuestador puede perder la información captada o se le puede deteriorar debido a la naturaleza de los materiales que utiliza para esta actividad.
- El digitador puede introducir en la aplicación datos incorrectos tomando como punto de partida que este no es el que realiza la encuesta ni el que la aplica en el terreno.

En Cuba hoy no existe una infraestructura adecuada para implantar SIGE a todos los niveles: municipal, provincial y nacional. Todo esto, sumado al volumen de información a captar, que a grandes rasgos constituye la información estadística de todas las empresas del país, hace que el sistema implantado presente deficiencias en aquellos entornos que tengan como característica la situación anterior. Especialmente esto ocurre en las comunicaciones de municipio a provincia.

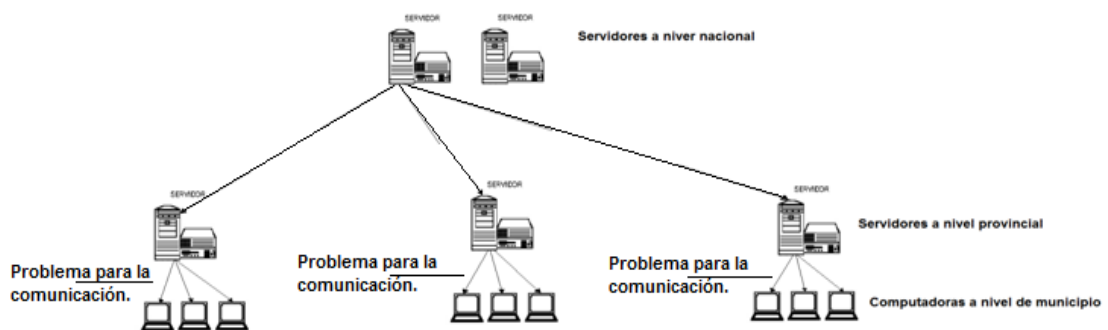
Anexos

1.3 Situación actual



Pasos para la captura de información actualmente en el SIGE.

Representación de la infraestructura informática que utiliza SIGE en el país y se muestra también donde ocurren principalmente los problemas para la transferencia de información.



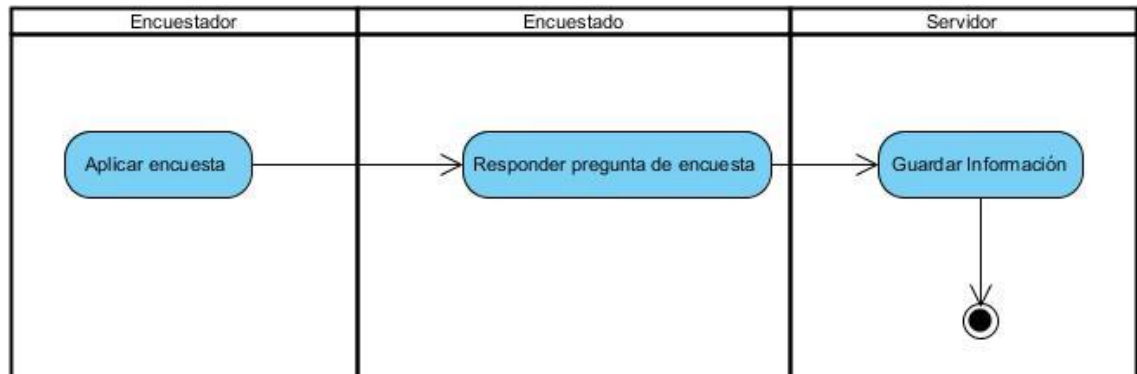
1.4 Estudio

Utilizando SIGEMovil se haría todo el proceso de captación de la información en dos pasos.

- Se inicia cuando el encuestador realiza las preguntas al encuestado o encuestados.

Anexos

- El encuestador introduce los datos en un celular.



Pasos para la captura de información con un celular.

La utilización SIGEMovil permitirá:

- Oportunidad de la información.
- Reducción de los recursos materiales.
- Calidad de la información.
- Reducción de los recursos humanos.

1.6 Resultados.

La Interfaz Web de captura de datos en dispositivos móviles en el Sistema Integrado de Gestión Estadística propicia:

- Oportunidad de la información pues se le llega a la persona que necesita la información más rápido.
- Reducción de los recursos materiales (hojas, lapiceros, computadoras, servidores) asociados a la captura de la información a partir de la sustitución de estos por móviles.
- Calidad de la información debido a que la misma persona que capta la información es la encargada de digitalizarla, reduciendo así los errores humanos.