

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título: Desarrollo del módulo Peso Específico de los Suelos para el Sistema de Gestión de Informes de Ensayos.

Autora: Arid Sofía Hernández Garcell.

Tutor(es): Ing. Eloy Jiménez Iglesias.

Ing. Robin Días Matos.

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez” y al Departamento de Informática para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ de 2013.

Arid Sofía Hernández Garcell _____
Firma Autor

Ing. Robin Díaz Matos _____
Firma Tutor

Ing. Eloy R. Jiménez Iglesias _____
Firma Tutor

Dedicatoria

Este trabajo va dedicado a mis padres y demás familiares por su apoyo incondicional.

A mis amigas y amigos que estuvieron siempre conmigo.

A todos aquellos que colaboraron en la realización de este trabajo.

Agradecimientos

Arid Sofia Hernández Garcell

Le agradezco a Dios por estar siempre conmigo.

A mis padres y demás familiares, por su apoyo incondicional.

A mis amigos, Compañeros, Profesores; en fin, todos los que se alguna manera contribuyeron en mi formación como profesional.

Gracias.

¡Que Dios Les Bendiga!

Resumen

En los laboratorios de Mecánica de Suelos y Rocas del Instituto Superior Minero Metalúrgico de Moa (ISMMM) se realizan informes de diferentes tipos de ensayos. Todo este proceso genera un gran cúmulo de datos que debe ser gestionado, esta información que se encuentra en diversos formatos, lo que provoca dificultades para su organización. Este trabajo trata acerca de la implementación del Módulo Peso Específico de los Suelos, para el Sistema de Gestión de los Informes de Ensayos.

El Sistema informático permitirá registrar todos los datos concerniente a los ensayos, o sea, desde el mismo momento en que se realiza el muestreo, toda su trayectoria en el laboratorio, hasta que se realicen los cálculos, posibilitando la actualización de los informes, siempre que sea necesario.

Abstract

In the laboratories of Mechanics of Floors and Rocks of the Institute Superior Mining Metallurgist of Moa (ISMMM) there are carried out reports of different types of rehearsals. This whole process generates a great heap of data that should be negotiated, this information that is in diverse formats, what causes difficulties for its organization. This work it's about the implementation of the Module Specific Weight of the Floors, for the System of Administration of the Reports of Rehearsals.

The System informatics will allow to register all the concerning data to the rehearsals, that is to say, from the same moment in that is carried out the sampling, all its trajectory in the laboratory, until carried out the calculations, facilitating the upgrade of the reports, whenever it is necessary.

Índice

Declaración de Autoría.....	II
Dedicatoria	IV
Agradecimientos	V
Introducción.....	1
Fundamentación Teórica	6
1.1 Introducción.....	6
1.2 Conceptos Fundamentales.....	6
1.3 Estado del arte	8
1.4 Tendencias.....	8
1.4.1 Software Libre.....	8
1.4.1.1 Libertades del Software Libre	8
1.4.2 Lenguajes de Programación	9
1.4.2.1 Java	9
1.4.3 Gestores de Bases de Datos.	12
1.4.3.1 PostgreSQL.....	13
1.5 Herramientas.....	15
1.5.1 Entorno de desarrollo NetBeans:	15
1.5.2 Visual Paradigm 5.3.....	16
1.5.3 E/R Studio.....	16
1.6 Metodología de Desarrollo de Sistemas Informáticos.	17
1.6.1 XP (Extreme Programming).	17
1.7 Arquitectura.....	23
1.8 Conclusiones de Capítulo.....	25
Planeación y Diseño.....	26
2.1 Introducción.....	26
2.2 Personas relacionadas con el sistema.....	26
2.3 Lista de Reserva	26
2.4 Historias de Usuarios.	27
2.5 Planificación de entregas.....	29
2.5.1 Estimación de esfuerzos por historias de usuario	30
2.5.2 Plan de iteraciones	31

2.5.3 Plan de duración de las iteraciones.....	31
2.5.3.1 Primera iteración:	32
2.5.3.2 Segunda iteración:.....	33
2.5.3.3 Tercera iteración:.....	33
2.6 Clases, responsabilidades y colaboradores.....	33
2.6.1 Tarjetas CRC	35
2.7 Conclusiones.....	36
Desarrollo y Pruebas.....	37
3.1 Introducción.....	37
3.2 Modelo de Datos	37
3.3 Desarrollo de las iteraciones	37
3.3.1 Tareas por historias de usuario	37
3.4 Pruebas.....	38
3.4.1 Alcance de las pruebas.....	39
3.4.2 Desarrollo dirigido por pruebas	39
3.4.3 Pruebas de aceptación	40
3.5 Conclusiones.....	42
Estudio de Factibilidad	43
4.1 Introducción.....	43
4.2 Efectos Económicos.....	43
4.2.1 Efectos directos	44
4.2.2 Efecto indirecto	44
4.2.3 Intangibles	44
4.2.3.1 Situación sin proyecto	44
4.2.3.2 Situación con proyecto	44
4.3 Beneficios Y Costos Intangibles en el proyecto	45
4.4 Ficha de costo	45
4.4.1 Costos en Moneda Libremente Convertible	45
4.4.2 Costos en Moneda Nacional	46
4.5 Conclusiones.....	50
Conclusiones	51
Recomendaciones	52

Referencias Bibliográficas.....	53
Bibliografía.....	54
Glosario de Términos.....	X
Anexo 1.....	XII
Anexo 2.....	XXI
Anexo 3.....	XXV
Anexo 4.....	XXXIII

Índice de Tablas

Tabla 1: Personas relacionadas con el sistema.....	26
Tabla 2: Representación de una historia de usuario	28
Tabla 3: H.U. 1 Crear informe de Peso Específico de los Suelos	28
Tabla 4: Estimación de esfuerzo por historias de usuarios.....	30
Tabla 5: Plan de iteraciones	31
Tabla 6: Duración de las iteraciones.....	32
Tabla 7: Release	33
Tabla 8: Tarea #1 Crear informe de Peso Específico de los Suelos	38
Tabla 9: Plantilla de Pruebas de Aceptación	40
Tabla 10: Prueba de Aceptación #1 Crear informe de Peso Específico de los Suelos.	41
Tabla 1.1: H.U. 2 Buscar informe de Peso Específico de los Suelos	XII
Tabla 1.2: H.U. 3 Actualizar informe de Peso Específico de los Suelos	XIII
Tabla 1.3: H.U. 4 Listar informe de Peso Específico de los Suelos	XIV
Tabla 1.4: H.U. 5 Eliminar informe de Peso Específico de los Suelos	XV
Tabla 1.5: H.U. 6 Crear informe de Límites de Plasticidad.....	XVI
Tabla 1.6: H.U. 7 Buscar informe de Límites de Plasticidad.....	XVII
Tabla 1.7: H.U. 8 Actualizar informe de Límites de Plasticidad	XVIII
Tabla 1.8: H.U. 9 Listar informe de Límites de Plasticidad	XIX
Tabla 1.9: H.U. 10 Eliminar informe de Límites de Plasticidad	XX
Tabla 3.1: Tarea #2 Calcular informe de Peso Específico de los Suelos	XXV
Tabla 3.2: Tarea #3 Imprimir informe de Peso Específico de los Suelos	XXV
Tabla 3.3: Tarea #4: Guardar Informe de Peso Específico de los Suelos	XXV
Tabla 3.4: Tarea #5 Buscar informe de Peso Específico de los Suelos	XXVI
Tabla 3.5: Tarea #6: Actualizar informe de Peso Específico de los Suelos	XXVII
Tabla 3.6: Tarea #7 Actualizar informe de Peso Específico de los Suelos	XXVII
Tabla 3.7: Tarea #8: Eliminar informe de Peso Específico de los Suelos	XXVIII
Tabla 3.8: Tarea #9 Crear Informe de Límites de Plasticidad.....	XXVIII
Tabla 3.9: Tarea #10 Calcular Informe de Límites de Plasticidad.	XXVIII
Tabla 3.10: Tarea #11 Graficar Informe de Límites de Plasticidad.	XXIX

Tabla 3.11: Tarea #12 Imprimir Informe de Límites de Plasticidad	XXX
Tabla 3.12: Tarea #13 Guardar Informe de Límites de Plasticidad.	XXX
Tabla 3.13: Tarea #14 Buscar Informe de Límites de Plasticidad.....	XXXI
Tabla 3.14: Tarea #15 Actualizar Informe de Límites de Plasticidad.	XXXI
Tabla 3.15: Tarea #16 Listar Informe de Límites de Plasticidad.	XXXII
Tabla 3.16 Tarea #17 Eliminar Informe de Límites de Plasticidad.	XXXII
Tabla 4.1 Prueba de Aceptación #2 Buscar informe de Peso Específico de los Suelos	XXXIII
Tabla 4.2: Prueba de Aceptación #3 Actualizar informe de Peso Específico de los Suelos	XXXIV
Tabla 4.3: Prueba de Aceptación #4 Listar informe de Peso Específico de los Suelos	XXXV
Tabla 4.4: Prueba de Aceptación #5 Eliminar informe de Peso Específico de los Suelos	XXXVI
Tabla 4.5: Prueba de Aceptación #6 Crear informe de Límites de Plasticidad	XXXVII
Tabla 4.6: Prueba de Aceptación #7 Buscar informe de Límites de Plasticidad.....	XXXVIII
Tabla 4.7: Prueba de Aceptación #8 Actualizar informe de Límites de Plasticidad.	XXXIX
Tabla 4.8: Prueba de Aceptación #9 Listar informe de Límites de Plasticidad.....	XL
Tabla 4.9: Prueba de Aceptación #10 Eliminar informe de Límites de Plasticidad	XLI

Índice de Figuras

Figura 1: Plantilla de Tarjetas CRC.....	34
Figura 2: Tarjeta CRC #1 Datos Generales	35
Figura 3: Gráfica de la solución sin el producto y solución con el producto.....	49
Figura 2.1 Tarjeta CRC #2 realizado	XXI
Figura 2.2 Tarjeta CRC #3 resultado índice plasticidad.....	XXI
Figura: 2.3 Tarjeta CRC #4 húmedo	XXII
Figura 2.5 Tarjeta CRC #6 límite plástico	XXIII
Figura 2.6 Tarjeta CRC #7 seco	XXIII
Figura 2.7 Tarjeta CRC #8Picnómetro	XXIV

Introducción

El desarrollo científico-técnico en la actualidad es uno de los pilares fundamentales en el progreso de la humanidad, implicando la renovación y el perfeccionamiento continuo de herramientas, funcionalidades y productos que satisfagan las necesidades existentes. La informática, como ciencia encargada del tratamiento adecuado de la información se extiende a todos los sectores de la sociedad, y su rápida evolución presiona a mantener el ritmo en dicho desarrollo. Su relación con otras ciencias se hace necesaria, especialmente en la geología para informatizar y facilitar el trabajo a los profesionales en cuanto a los procesos geológicos.

La Ingeniería Geológica es la ciencia aplicada al estudio y solución de los problemas de la ingeniería y del medio ambiente, producidos como consecuencia de la interacción entre las actividades humanas y el medio geológico, y su fin es asegurar que los factores geológicos condicionantes de las obras de ingeniería, sean tenidos en cuenta e interpretados adecuadamente, así como evitar o mitigar las consecuencias de los riesgos geológicos. Otras ciencias afines, como la mecánica del suelo y la mecánica de rocas, configuraron los principios de la moderna geotecnia, dentro de la cual la ingeniería geológica representa la visión adecuada a la solución de los problemas constructivos. (Vallejo, 2002).

Cuba es uno de los países que ha identificado desde muy temprano, la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones (TIC), para lograr una cultura digital como una de las características imprescindibles del hombre nuevo. En las instituciones universitarias cubanas, se trabaja para lograr una serie de acciones encaminadas al desarrollo de información, conocimiento e inteligencia, que fundamentalmente impacten en el medio en que se mueven.

El Instituto Superior Minero Metalúrgico de Moa (ISMMM) sobre todo en el área de Mecánica del suelo y rocas, se realizan diferentes tipos de experimentos. La realización de Informes de Ensayos, es una técnica empleada por los Geólogos para almacenar toda la información referente a la construcción de cualquier tipo de edificación. Los Departamentos de Geología e Informática han creado un estrecho vínculo y especialistas de ambas ciencias implementaron una herramienta (Sistema de Gestión de Informes de

Ensayos) que gestiona parte del proceso de los informes de ensayos; pero el proceso de cálculo del Peso Específico de los Suelos se realiza de forma manual, lo que trae consigo la pérdida de tiempo en las actividades docentes e investigativas. La interpretación de estos resultados es otra actividad que presenta inconvenientes ya que no existe una forma única de interpretarlos, provocando serias dificultades durante la confección de los informes. Por las condiciones mencionadas en este proceso, se utilizan de forma innecesaria, cantidades excesivas de papel y tinta donde en muchos casos las investigaciones se pierden o están duplicadas por carencia de centralización de la información.

Por los problemas evidenciados se plantea como **problema a resolver** en este trabajo ¿cómo facilitar el cálculo e interpretación del Peso Específico de los Suelos en los laboratorios de Mecánica del Suelo y Rocas del ISMMM.

Se define como **objeto de estudio** los Sistemas de gestión de información vinculados a la geología.

Teniendo como **campo de acción** la Informatización del cálculo del Peso Específico de los Suelos e interpretación de los resultados, en la carrera ingeniería geológica en el ISMMM.

Para resolver el problema identificado se propone el siguiente **objetivo general**: Implementar el Módulo Peso Específico de los Suelos.

Para guiar el trabajo se plantea la siguiente **idea a defender**: Si se implementa el Módulo Peso Específico de los Suelos para el cálculo e interpretación de los resultados, facilitará el trabajo de los especialistas, permitiéndole mayor rapidez y eficiencia en el proceso docente-investigativo, además se enriquece el Sistema de Gestión de Informes de Ensayos.

Debido a esto planteamos los siguientes **objetivos específicos**:

- Elaborar un marco teórico conceptual para la confección del Módulo Peso Específico de los Suelos.
- Analizar, Diseñar e Implementar el Módulo Peso Específico de los Suelos.

- Realizar pruebas de aceptación y un estudio de factibilidad al sistema.
- Realizar la documentación del mismo.

Para el logro de los objetivos fue necesario plantearse las siguientes **tareas específicas**:

- Detallar el proceso de gestión del cálculo del Peso Específico de los Suelos.
- Establecer una búsqueda de los antecedentes del sistema informático propuesto
- Estudiar las tecnologías, metodología y herramientas para la construcción del software.
- Seguir cada una de las etapas de la metodología de construcción de software.
- Realizar el análisis y diseño del módulo propuesto.
- Realizar la implementación del módulo.
- Realizar el estudio de factibilidad.
- Efectuar pruebas a los módulos desarrollados.
- Desarrollar el Manual de Usuarios.

Para cumplir estas tareas se han empleado **métodos teóricos** y **empíricos** de la investigación científica. Entre los métodos empíricos usados podemos citar la observación, entrevista, la modelación y el análisis de documentos para la recopilación de la información. La observación fue útil para entender el comportamiento del sistema y sus especificaciones. La entrevista nos ayudó a determinar los principales requerimientos del sistema y las nuevas funcionalidades que necesita plasmados en las historias de usuarios. Mediante el análisis de la documentación disponible, conocimos el funcionamiento actual del proceso de evaluación de muestras.

Los métodos teóricos proporcionan calidad a la investigación. En el desarrollo del proceso de investigación se usaron el análisis y síntesis para la recopilación y el procesamiento de la información obtenida en los métodos empíricos y arribar a las conclusiones de la investigación. Con la representación de la realidad se logró detectar problemas en la

forma actual de manejar la información y encontrar las funcionalidades que debe de tener el sistema que se propone, que lo harán más completo y le brindarán satisfacción al usuario con un producto de mayor calidad.

Aportes de la investigación:

Este trabajo tiene como **aporte práctico** la implementación del Módulo Peso Específico de los Suelos para el cálculo e interpretación de los ensayos de laboratorios de Mecánica del Suelo y Rocas.

El siguiente trabajo está compuesto por **Introducción, 5 Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Glosario de Términos y Anexos.**

En el **Capítulo 1 Fundamentación Teórica**, se define el flujo actual de los procesos, se hace un análisis crítico de la ejecución de los procesos, y una breve referencia sobre el estado del arte del sistema, además especificamos las herramientas, metodología y tecnología para el desarrollo de la aplicación, permitiéndonos obtener resultados funcionales observables a corto plazo.

En el **Capítulo 2 Planeación y Diseño**, se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema para su terminación. Además se presentan las tarjetas Clases, Responsabilidades y Colaboradores (CRC), que permitirán trabajar con una metodología basada en objetos.

En el **Capítulo 3 Desarrollo y Prueba**, se presenta el modelo de datos empleado para la aplicación concluyente, y se realiza el desarrollo de las iteraciones a partir del desglose de las historias de usuario en tareas. Asimismo aparecen las interfaces graficas de usuario diseñadas para la aplicación final. Se describen además las pruebas realizadas y se indican las respuestas de la aplicación en el empleo de las diferentes funcionalidades, así como los posibles mensajes de error, información o aceptación que emite la misma cuando se utiliza una de estas funcionalidades.

En el **Capítulo 4 Estudio de Factibilidad**, se tienen en cuenta los costos a incurrir, deduciéndose si el proyecto realizado será factible o no llevarlo a cabo. Hay muchas formas de calcular el costo, pero para nuestro caso se utilizará la Metodología Costo Efectividad, la cual sugiere que la conveniencia de la ejecución de un proyecto se determina por la observación de ciertos factores en conjunto.

Fundamentación Teórica

1

1.1 Introducción

En este capítulo se define el flujo actual de los procesos, se hace un análisis crítico de la ejecución de los procesos y una breve referencia sobre el estado del arte del sistema. También se hace una descripción sobre las tendencias y tecnologías actuales sobre las que se apoya el sistema, por otra parte se especifican las herramientas utilizadas para el desarrollo del sistema y el porqué de su uso.

1.2 Conceptos Fundamentales

Definiciones de Suelo:

Del macizo geológico es el suelo la parte que está en contacto directo con el hombre, en él se ejecutan las actividades principales del ser humano. La agricultura, construcciones y otras manifestaciones lo utilizan como medio, en el primer caso para la producción de alimentos y en el segundo para la edificación de las ciudades y países. Para nosotros es el segundo aspecto el que nos ocupa debido a que el suelo se integra como elemento constructivo, es decir, como un material más para la proyección de las obras ingenieras relacionadas con la construcción.

Tradicionalmente es la Mecánica de los Suelos la que se identifica con su estudio para su uso en la construcción. Los suelos están formados por minerales que poseen su forma característica y que al unirse dan lugar a la estructura de aquellos, las cuales son diferentes para los tipos de suelos. Los suelos gruesos (arenas y gravas) están formados por minerales en los cuales predomina la forma equidimensional (lados aproximadamente iguales) adquiriendo la forma redondeada y angulosa en diferentes magnitudes. Los suelos finos (limos y arcillas) están formado por minerales en los cuales predomina la forma aplastada (laminar).

Los suelos como parte integrantes de las rocas están formados por las fases sólida, líquida y gaseosa. La influencia de las diferentes fases en los suelos es mucho más significativa que en las rocas debido fundamentalmente a la poca consistencia de ellos.

- **Peso Específico de los Suelos:** Entre las principales propiedades físicas de los suelos areno-arcillosos se encuentra el Peso Específico, el cual expresa el estado físico así como el estado en las condiciones de yacencia natural o en obras como son las obras de tierra (presas, terraplenes, etcétera), permitiéndonos tener una idea indirecta sobre la solidez, estabilidad y deformabilidad de los suelos areno-arcillosos, así como también acerca de su utilización bajo la influencia de los procesos geológicos o factores artificiales. (Badillo, 1969).

$$\gamma_{s2} = \frac{W_s}{W_s + (W_p - W_{ps})} \cdot K$$

Ws: Masa seca del suelo, en gramos.

Wp: Masa del picnómetro lleno de agua hasta igualarse a la temperatura de en gramos.

Wps: Masa del picnómetro más agua, más suelo a la temperatura de ensayo, en gramos.

K: Constante para llevarlo a la temperatura de 20 °C.

- **Límites de Plasticidad:** En el caso de algunos terrenos húmedos, la plasticidad es la propiedad que les permite ser moldeados aplicándoles fuerzas externas, y mantener las formas adquiridas, aun cuando la humedad y las fuerzas externas desaparezcan. Según Atterberg se pueden definir dos límites de plasticidad, el máximo y el mínimo. Con porcentaje de humedad por encima del límite máximo de plasticidad, la masa terrosa adquiere fluidez y pierde su capacidad de mantener la forma, y si el terreno tiene un porcentaje de humedad por debajo del límite mínimo de plasticidad, la masa terrosa se vuelve quebradiza, y no se puede moldear. Es evidente que no todos los suelos tienen la misma plasticidad; las arenas y los limos tienen una plasticidad baja o muy baja, mientras que suelos con alto contenido de arcillas tienen una plasticidad mayor. En línea general puede afirmarse que terrenos con un contenido de arcilla inferior al 15% no son plásticos.

1.3 Estado del arte

En el año 2012 dos estudiantes de la carrera Informática como trabajo de tesis implementaron una herramienta que gestiona parte del proceso de Gestión de Informes de Ensayos (SIGIE), esta contiene los módulos Hinchamiento Controlado, Ángulo de Reposo, Límites de Contracción y Límites de Plasticidad. La misma está siendo utilizada en los laboratorios de Mecánica del Suelo y Rocas del ISMMM, solo que presenta dificultad a la hora de graficar el Límite de Plasticidad y no cuenta con el Módulo Peso Específico de los Suelos el cual se realiza de forma manual, por lo que surge la necesidad de implementarlo.

Realizando una búsqueda exhaustiva, partiendo de la investigación que se hizo anteriormente, se arrojó como resultado que solo existe un software a nivel nacional que realiza parte de estos informes; el SIGIE, utilizado en el laboratorio de Mecánica del Suelo y Rocas del ISMMM, pero no se encontró referencia de otro sistema que realice las mismas funcionalidades a nivel internacional. Por tanto no hay un software que satisfaga la necesidad actual de realizar el Módulo Peso Específico de los Suelos.

1.4 Tendencias

1.4.1 Software Libre

Según la Free Software Foundation (Fundación de Software Libre), el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el Software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software, la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo cual se puede ayudar a otros y de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie, para la segunda y última libertad mencionadas, el acceso al Código fuente es un requisito previo.

1.4.1.1 Libertades del Software Libre

- La libertad de ejecutar el programa para cualquier propósito.

- La libertad de estudiar cómo trabaja el programa y adaptarlo a sus necesidades (El acceso al código fuente es una condición necesaria).
- La libertad de redistribuir copias para que pueda ayudar al vecino.
- La libertad de mejorar el programa y publicar sus mejoras y versiones modificadas en general para que se beneficie toda la comunidad (El acceso al código fuente es una condición necesaria).

Las ventajas especialmente económicas que brindan las soluciones libres y las aportaciones de la comunidad de desarrollo han permitido un constante crecimiento del software libre hasta superar en ocasiones al mercado propietario. Estas ventajas hacen que nuestro país siga una política de migración hacia el software libre y como parte de este proceso se decide para el desarrollo de la aplicación la utilización de herramientas y tecnologías pertenecientes al software libre (Subirós, Junio 2009).

1.4.2 Lenguajes de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. (Subirós, Junio 2009)

1.4.2.1 Java

Es un lenguaje de programación desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre. (Eckel, 2000).

Las características principales que nos ofrece Java son:

- **Simple:** Ofrece toda la funcionalidad de un lenguaje potente. Debido a que C y C++ son los lenguajes más difundidos, Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. Elimina muchas de las características de otros lenguajes para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.
- **Orientado a objetos:** Con el objetivo de mantener la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, clases y sus copias, instancias. Estas instancias, necesitan ser construidas y destruidas en espacios de memoria. Incorpora funcionalidades como por ejemplo, la resolución dinámica de métodos mediante una interfaz específica llamada RTTI (Run Time Type Identification) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.
- **Distribuido:** Se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. Proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.
- **Robusto:** Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja

la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

- **Arquitectura neutral:** Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows NT, Linux, Irix, Aix, Mac, Apple y probablemente haya grupos de desarrollo trabajando en el porting a otras plataformas.
- **Seguro:** La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el casting se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador. El código Java pasa muchos tests antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto. El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior. Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego

en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

- **Multihebra:** Soporta sincronización de múltiples hilos de ejecución (multi threading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

1.4.3 Gestores de Bases de Datos.

Una Base de Datos (BD) es un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora, o sea, que puede considerarse una colección de datos variables en el tiempo. El software que permite la utilización y la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina Sistema de Gestión de Bases de Datos (SGBD), cuyo objetivo fundamental consiste en suministrar al usuario las herramientas que le permitan manipular los datos en términos abstractos, de forma que no necesite conocer el modo de almacenamiento de los mismos en la computadora, ni el método de acceso empleado (Guerra, 2009).

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Características

- **Abstracción de la información:** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

- **Independencia:** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima:** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- **Consistencia:** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad:** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra protegida frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipularla o destruirla; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad:** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Entre los SGBD comúnmente utilizados en el mundo tenemos MySQL y PostgreSQL, Oracle. Estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

1.4.3.1 PostgreSQL

Es un sistema gestor de base de datos objeto-relacional, bajo licencia BSD. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al

dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. Es el sistema de gestión de bases de datos de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre. Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos y Windows. Su documentación está muy bien organizada, pública y libre, y posee comentarios de los propios usuarios. Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, Soporte de todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.). Altamente adaptable a las necesidades del cliente (CAVSI, 2004).

Principales Características:

- **Atomicidad (Indivisible):** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia:** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- **Aislamiento:** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

Ventajas

- **Inhalación ilimitada:** Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.
- **Soporte:** Además de nuestras ofertas de soporte, tenemos una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.

- **Ahorros considerables en costos de operación:** PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
- **Estabilidad y Confiabilidad Legendarias:** PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.
- **Extensible:** El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.
- **Multiplataforma:** PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.

1.5 Herramientas

1.5.1 Entorno de desarrollo NetBeans:

Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Micro Systems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos. La Plataforma NetBeans es una base modular extensible usada como una estructura de integración para crear aplicaciones. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

El IDE NetBeans 7.0 es una herramienta para programadores, pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE,

JavaEE, Web, EJB y aplicaciones móviles). Entre sus características se encuentran un sistema de proyectos basado en Ant, control de versiones y refactoring. El IDE NetBeans IDE 7.0 corre en diferentes sistemas operativos entre los que podemos mencionar Linux, Mac OS X, Solaris y Windows. Como se puede observar se ha lanzado con soporte para la mayoría de las plataformas. Como requerimiento se necesita tener previamente instalado el JDK 6.0 o 7.0.

1.5.2 Visual Paradigm 5.3

Visual Paradigm para UML 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 más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se prefiere su utilización en el desarrollo del módulo por ser una herramienta no propietaria, multiplataforma, amigable en su entorno, lo que facilita su interoperabilidad con otras aplicaciones además de que tiene un uso de un lenguaje estándar común a todo el equipo de desarrollo lo que facilita la comunicación. (Visual Paradigm, 2009).

Principales Características:

- Producto de calidad.
- Soporta diseño de aplicaciones Web.
- Soporta varios idiomas.
- Fácil de instalar y actualizar.
- Licencia gratuita.

1.5.3 E/R Studio

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias

de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos. ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes. (ER/Studio, 2007)

1.6 Metodología de Desarrollo de Sistemas Informáticos.

Desarrollar un buen software depende de un sinnúmero de actividades y etapas, donde el impacto de elegir la mejor metodología para un equipo en un determinado proyecto, es trascendental, para el éxito del producto. El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte se encuentran aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán (Guerra, 2009).

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Pueden ser comparadas con un plan de contingencias en el que se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además quienes deben participar en el desarrollo de las actividades y qué papel deben de tener. Detallan además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (Solís Álvarez, 2005).

1.6.1 XP (Extreme Programming).

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. XP es un enfoque de la ingeniería de software formulado por Kent Beck y De Jean, Extreme Programming Explained:

Embrace Change (1999). Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. (Solís Álvarez, 2005).

Fases de la metodología XP

Las prácticas que componen la programación extrema se pueden agrupar en cuatro grandes bloques: plan, diseño, codificación y pruebas. Sin embargo, estos bloques no deben realizarse en orden, si no que cada uno consta de una serie de actividades, y todas ellas se irán realizando de manera evolutiva.

- **Planificación:** Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para el user storie y permita hacer una estimación de tiempo de desarrollo del mismo. Se crea un plan de lanzamiento (reléase planning), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usará como base los user stories, participando el cliente en la elección de los que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto. El desarrollo se divide en iteraciones, cada una de las cuales comienza con un plan de iteración para el que se eligen las user

stories a desarrollar y las tareas de desarrollo. Se cambia el proceso lo que sea necesario para adaptarlo a tu proyecto.

- **Diseño:** Se eligen los diseños más simples que funcionen. Se elige una metáfora del sistema para que el nombrado de clases, etcétera, siga una misma línea, facilitando la reutilización y la comprensión del código. Se escriben tarjetas CRC de clase-responsabilidades-colaboración para cada objeto, que permiten abstraerse el pensamiento estructurado y que el equipo de desarrollo al completo participe en el diseño.
- **Codificación:** El cliente está siempre disponible, a ser posible cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP (escribe las historias de usuarios con la ayuda de los desarrolladores, participa en la elección de los que entrarán en el plan de lanzamientos, prueba pequeños lanzamientos, participa en las pruebas de funcionalidad...). La idea es usar el tiempo del cliente para estas tareas en vez de para que cree una detalladísima especificación de requisitos, y evitar la entrega de un producto peor que le hará perder tiempo. El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código. Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro qué objetivos tiene que cumplir lo que se va a codificar. La programación del código se realizará en parejas, para aumentarla calidad del mismo. En cada momento, sólo habrá una pareja de programadores integrando código. Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes periodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades habrán sido detectadas enseguida. Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentarla contribución de ideas por parte de todo el

equipo de desarrollo, se deja la optimización para el final. No se hacen horas extra de trabajo.

- **Pruebas:** Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado. Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo. Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las user stories elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que el correspondiente user storie se ha completado.

Prácticas de XP

“Las principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar al desarrollo de software y a la aplicación disciplinada de las siguientes prácticas”.

- **El juego de la planificación**

Hay una comunicación frecuente entre el cliente y los programadores. El equipo de técnico realiza una estimación del esfuerzo requerido para la implantación de las historia de usuarios y los clientes deciden sobre el ámbito y el tiempo de las entregas de cada iteración.

- **Entregas pequeñas**

Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión del ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de tres meses.

- **Metáfora**

El sistema es definido mediante un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo funciona el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).

- **Diseño simple**

Se debe diseñar la más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

- **Pruebas**

La producción de código está por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.

- **Refactorización (Refactoring)**

Es el proceso de modificar el código de un sistema software modo que no se altere su comportamiento externo pero se mejore su estructura interna. Es una técnica disciplinada de reestructuración de código. Parte del proceso que puede automatizarse y de hecho existen herramientas que facilitan la tarea.

- **Programación en parejas**

Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva a ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).

- **Propiedad colectiva del código**

Cualquier programador cambia cualquier parte del código en cualquier momento.

- **Integración continua**

Cada pieza de código es integrada en el sistema una vez que esté lista. Así el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

- **Cliente in-situ**

El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportara mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

- **Estándares de programación**

“XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible”.

Las seis prácticas esenciales por donde se debe empezar:

- Pruebas.
- Refactorización.
- Programación en pares.
- Juego de planificación.
- Liberaciones pequeñas.
- Integraciones sistemáticas.

Valores que promueve XP:

- **Comunicación:** XP se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación como la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio. Cuando dos o más personas se comunican directamente pueden no sólo

consumir las palabras formuladas por la otra persona, sino que también aprecian los gestos, miradas etcétera, que hace su compañero. Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva.

- **Coraje:** El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe tener el coraje de exponer sus dudas, miedos, experiencias sin “embellecer” éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es grave.
- **Simplicidad:** Dado que no se puede predecir cómo va a ser en el futuro, el software que se desarrolla; un equipo de programación extrema intenta mantener el software lo más sencillo posible. Esto quiere decir que no se va a invertir ningún esfuerzo en hacer un desarrollo que en un futuro pueda llegar a tener valor. En XP, frases como, “...en un futuro vamos a necesitar...” o, “...haz un sistema genérico de...”, no tienen ningún sentido ya que no aportan ningún valor en el momento.
- **Retroalimentación:** La agilidad se define, entre otras cosas, por la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Por este motivo uno de los valores que nos hace más ágiles es el continuo seguimiento o retroalimentación que recibimos a la hora de desarrollar en un entorno ágil de desarrollo. La retroalimentación se toma del cliente, de los miembros del equipo y de todo el entorno en el que se mueve un equipo de desarrollo ágil.

1.7 Arquitectura

La Arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". En la Web es muy común la utilización de la arquitectura “n-capas”, “MVC”, entre otras. El lenguaje de programación JAVA, seleccionado anteriormente, implementa a su vez el patrón arquitectónico MVC, es por ello que

adoptamos esta arquitectura para el desarrollo de la propuesta de solución. Modelo Vista Controlador (MVC) es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Presenta el modelo en un formato adecuado, como en una página Web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) logrando un mantenimiento más rápido y sencillo de las aplicaciones.

Ejemplo, para el caso de la web, si se fuera a mostrar una misma aplicación en un navegador estándar, como en un navegador de un dispositivo móvil, sólo es necesario crear una vista nueva por cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones

(Aplicación de escritorio, HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

Ventajas del Modelo Vista Controlador

- La separación del Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación en caso de ser requerido.

1.8 Conclusiones de Capítulo

En este capítulo hemos abordados los conceptos fundamentales asociados al dominio del problema. Para la implementación del módulo Peso Específico de los Suelos no se realizó un amplio estudio de herramientas ya que la construcción de estos módulos se seguirá bajo la misma línea del Sistema de Gestión de Ensayos ya implementado donde encontramos a Java como lenguaje de programación y NetBeans7.0 como IDE desarrollo, PostgreSQL como Gestor de Base de Datos además, se continuará la aplicación sobre la base de la metodología ágil XP, pues nos permitirá obtener resultados funcionales observables a corto plazo.

Planeación y Diseño

2

2.1 Introducción

En este capítulo, se introduce la fase de planeación y diseño, donde se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema para su terminación. Además se presentan las tarjetas Clases, Responsabilidades y Colaboradores (CRC), que permitirán trabajar con una metodología basada en objetos.

2.2 Personas relacionadas con el sistema

Tabla 1: Personas relacionadas con el sistema.

Persona(s) relacionadas con el sistema	Justificación
Administrador	Es la persona responsable de administrar y controlar la aplicación, otorgándoles a los usuarios sus respectivos permisos.
Usuario	Esta persona es la que tiene cierto conocimiento de cómo se realizan los diferentes tipos de informes de ensayos así como interpretar sus resultados.

2.3 Lista de Reserva

Después de conocer el personal relacionado e identificar los requisitos generales, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para darle respuesta a los mismos. Para ello se enumerarán mediante una lista de reserva, las funcionalidades que el sistema debe capaz de cumplir.

De acuerdo a lo antes expuesto, el sistema debe ser capaz de:

1. Crear informe de Peso Específico de los Suelos.
2. Buscar informe de Peso Específico de los Suelos.
3. Actualizar informe de Peso Específico de los Suelos.
4. Calcular informe de Peso Específico de los Suelos.
5. Listar informe de Peso Específico de los Suelos.
6. Eliminar informe de Peso Específico de los Suelos.
7. Guardar informe de Peso Específico de los Suelos.
8. Imprimir informe de Peso Específico de los Suelos.
9. Crear informe de Límites de Plasticidad.
10. Buscar informe de Límites de Plasticidad.
11. Actualizar informe de Límites de Plasticidad.
12. Calcular informe de Límites de Plasticidad.
13. Listar informe de Límites de Plasticidad.
14. Eliminar informe de Límites de Plasticidad.
15. Guardar informe de Límites de Plasticidad.
16. Imprimir informe de Límites de Plasticidad.
17. Graficar informe de Límites de Plasticidad.

2.4 Historias de Usuarios.

Las historias de usuario (HU) son la técnica utilizada en XP para especificar los requisitos del software. Estos constituyen el resultado directo de la interacción entre los clientes y desarrolladores a través de reuniones donde el flujo de ideas determina no solo los requerimientos del proyecto sino también las posibles soluciones. De forma general se describen brevemente las características que el sistema debe tener desde el punto de vista del cliente. Para definir las historias de usuario se emplea la siguiente plantilla.

Tabla 2: Representación de una historia de usuario

Historia de Usuario	
Número: (Número de la Historia de Usuario)	Usuario: (Usuario entrevistado para obtener la función requerida a automatizar)
Nombre historia: (Nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente)	
Prioridad en negocio: (Importancia de la historia para el cliente) (Alta / Media / Baja)	Riesgo en desarrollo: (Dificultad para el programador) (Alto / Medio / Bajo)
Puntos estimados: 1-3	Iteración asignada: (Iteración a la que corresponde)
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: (Se especifican las operaciones por parte del usuario y las respuestas que dará el sistema)	
Observaciones: (Algunas observaciones de interés, como glosario, información sobre usuarios, etc.)	

A continuación se muestran las historias de usuarios confeccionadas por el cliente, teniendo estas las funcionalidades principales a integrar en la aplicación.

Tabla 3: H.U. 1 Crear informe de Peso Específico de los Suelos

Historia de Usuario	
Número: 1	Usuario: Beatriz Riverón
Nombre historia: Crear informe de Peso Específico de los Suelos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta

Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Arid Sofía Hernández Garcell	
Descripción: El usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, y se llenan los campos mostrados en el formulario con las opciones de calcular, guardar y cancelar.	
Observaciones: Para poder realizar la funcionalidad calcular es necesario que los campos especificados estén llenos y con los tipos de datos requeridos de lo contrario se mostrarán mensajes indicando el tipo de error.	

Ver Historias de Usuarios [ANEXO 1](#)

2.5 Planificación de entregas

En esta parte se establece la prioridad de cada historia de usuario así como una estimación del esfuerzo necesario de cada una de ellas con el fin de determinar un cronograma de entregas. Las estimaciones de esfuerzo asociado a la implementación de las historias se establecen utilizando como medida el punto. Un punto, equivale a una semana ideal de programación (6 días). Las historias generalmente valen de 1 a 3 puntos. Por otra parte, se mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

2.5.1 Estimación de esfuerzos por historias de usuario

El plan de entregas se realiza teniendo en cuenta las unidades funcionales que se quieren entregar y cada uno de estos módulos abarca un número de historias de usuarios a implementar para dar cumplimiento al funcionamiento del mismo.

Tabla 4: Estimación de esfuerzo por historias de usuarios

Historia de Usuario	Número	Puntos Estimados
1.Crear informe de Peso Específico de los Suelos	1	3
2.Buscar informe de Peso Específico de los Suelos	2	2
3.Actualizar informe de Peso Específico de los Suelos	3	2
4.Listar informe de Peso Específico de los Suelos	4	1
5. Eliminar informe de Peso Específico de los Suelos.	5	1
6.Crear informe de Límites de Plasticidad	6	1
7.Buscar informe de Límites de Plasticidad	7	1
8.Actualizar de informes de Límites de Plasticidad	8	1
9.Listar informe de Límites de Plasticidad	9	1
10.Eliminar informe de Límite de Plasticidad	10	1

2.5.2 Plan de iteraciones

Tabla 5: Plan de iteraciones

Módulo	Historia(s) de Usuario que abarca
Peso Específico de los Suelos	1,2,3,4,5
Límites de Plasticidad	6,7,8,9,10

2.5.3 Plan de duración de las iteraciones

Como parte del proceso de desarrollo de un proyecto guiado por la metodología XP, se crea el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas.

Combinando el plan de entrega y el plan de iteraciones se harán releases o liberaciones al sistema en las fechas mostradas a continuación

Tabla 6: Duración de las iteraciones

Iteración	Orden de implementación por Historias de Usuario	Duración de la iteración
1	Crear Peso Específico de los Suelos (1) Buscar Peso Específico de los Suelos (2) Actualizar Peso Específico de los Suelos (3)	7 Semanas
2	Listar Peso Específico de los Suelos (4) Eliminar Peso Específico de los Suelos (5)	2 Semanas
3	Crear Límites de Plasticidad (6) Buscar Límites de Plasticidad (7) Actualizar Límites de Plasticidad (8) Listar Límites de Plasticidad (9) Eliminar Límites de Plasticidad (10)	5 Semanas

2.5.3.1 Primera iteración:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1, 2 y 3 las cuales hacen alusión a la creación, búsqueda y actualización de los Informes de Ensayos, la agregación de suplementos a estos, así como la asignación de responsables a los mismos. Además se tendrá la primera versión de prueba, que contará con dos modelos de desarrollo que incorporan todas las funcionalidades antes vistas, éstos modelos se le presentarán al

cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

2.5.3.2 Segunda iteración:

Esta iteración tiene como finalidad desarrollar las HU 4 y 5. Las mismas son las que brindan las funcionalidades de listar y eliminar la información de los informes de ensayos ya existentes. La versión que se obtenga de esta iteración en unión con la entregada en la iteración anterior se le facilitará al cliente para comprobar si cumple con las necesidades antes acordadas con él.

2.5.3.3 Tercera iteración:

Esta última iteración tiene como propósito llevar a cabo el desarrollo de la HU 6, 7, 8, 9 y 10. Las cuales pertenecen al Módulo Límite de Plasticidad con sus respectivas mejoras. Estas historias de usuario serán integradas con el resultado de las iteraciones anteriores, y como fruto de esta integración se obtendrá la versión 1.1 del producto final con los nuevos módulos implementados. A partir de este momento el software será puesto a un proceso de prueba para evaluar el desempeño del mismo.

Tabla 7: Release

Iteración/Modulo	Peso Específico de los Suelos	Peso Específico de los Suelos
Final 1ra Iteración	30/03/2013	
Final 2da Iteración		15/04/2013
Final 3ra Iteración		20/05/2013

2.6 Clases, responsabilidades y colaboradores.

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como *tarjetas CRC*, las cuáles se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas

permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño. Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.

Clase	
Description:	
Attributes:	
Name	Description
Responsibilities:	
Name	Collaborator

Figura 1: Plantilla de Tarjetas CRC

El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar.

2.6.1 Tarjetas CRC

A continuación se muestra una de las tarjetas CRC teniendo en cuenta estas las funcionalidades principales a integrar en la aplicación.

Datos_Generales	
Description: Esta es la clase que va almacenar todos generales del ensayo independientemente del tipo de exp	
Atributes:	
Name	Description
código	LLave primaria y código de la obra
muestra	Este es el número de la muestra a ensayar
cala	Número de la cala
norma	De acuerdo al tipo de ensayo es la norma a utilizar
registro	Registro de la obra
referencia	Referencia
obra	Nombre de la obra que se desea utilizar
nombre	Nombre del tipo de ensayo
fecha	Fecha en que se realiza el ensayo
descripción	Una breve descripción de las propiedades de la muestra
profundidadI	Profundidad que se tomó a la primera muestra
profundidadF	Profundidad que se tomó a la última muestra
Responsibilities:	
Name	Collaborator
Insertar_Datos_Generales()	datosgenerales
Modificar_Datos_Generales()	
Eliminar_Datos_Generales()	
Listar_Datos_Generales()	

Figura 2: Tarjeta CRC #1 Datos Generales

Ver Tarjetas CRC [ANEXOS 2](#)

2.7 Conclusiones

En este capítulo se abordó la fase de planeación y diseño donde se delinearon las HU con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Presentando además, las principales clases mediante el empleo de las tarjetas CRC, culminando así esta fase y se determina que el equipo de trabajo está lista para pasar a la siguiente etapa de desarrollo.

Desarrollo y Pruebas

3

3.1 Introducción

En este capítulo se inicia la fase de desarrollo y pruebas conforme a la metodología XP. Se presenta el modelo de datos empleado para la aplicación concluyente, y se realiza el desarrollo de las iteraciones a partir del desglose de las historias de usuario en tareas. Asimismo aparecen las interfaces gráficas de usuario diseñadas para la aplicación final. Se describen además las pruebas realizadas y se indican las respuestas de la aplicación en el empleo de las diferentes funcionalidades, así como los posibles mensajes de error, información o aceptación que emite la misma cuando se utiliza una de estas funcionalidades.

3.2 Modelo de Datos

En esta parte se muestra el modelo de datos empleado para la aplicación. Para la realización del informes de ensayos, se crean varias tablas en las que se van a almacenar los datos con los que va a trabajar la aplicación.

3.3 Desarrollo de las iteraciones

Durante la fase planificación y diseño fueron detalladas las historias de usuario correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

Para darle cumplimiento a cada HU, primeramente se debe realizar una revisión del plan de iteraciones, y si es necesario, se le hacen modificaciones a este.

3.3.1 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que

pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las tarjetas de tareas.

A continuación se presentan las Tareas de Ingeniería agrupadas por las respectivas historias de usuario a las que pertenecen.

Tabla 8: Tarea #1 Crear informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 1	Número de Historia: 1
Nombre de la Tarea: Crear Informe de Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 3
Fecha de Inicio: 20/enero/2013.	Fecha Fin: 05/febrero/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se muestra un formulario con los campos a llenar y se crea el informe.	

Ver las tareas [ANEXO 3](#)

3.4 Pruebas

En la Programación Extrema es esencial el desarrollo de las pruebas, permitiendo probar continuamente el código. Cada vez que se desea implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba, para ser pasada por el código posteriormente. El proceso constante de las pruebas permite la obtención un producto con mayor calidad ofreciendo a los programadores una mayor certeza en el trabajo que desempeñan. En la metodología XP hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas (TDD test driven development), desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración verificando así que se cumplió la funcionalidad requerida por el cliente. Con estas normas se obtiene un código simple y

funcional de manera bastante rápida y eficiente. Por esto es importante pasar las pruebas al 100%.

3.4.1 Alcance de las pruebas

La prueba de software tiene limitantes, tanto teóricos como prácticos. Desde el punto de vista teórico, la prueba es un problema que llamamos no-decidible; esto implica, de algún modo, que no podemos escribir un programa que pruebe los programas sin intervención humana. Sin embargo, como mencionábamos anteriormente, la prueba sí es automatizable en muchos aspectos. Desde el punto de vista práctico, la cantidad de posibilidades para probar exhaustivamente un sistema es sencillamente inmanejable; es necesario entonces utilizar técnicas adecuadas para maximizar la cantidad de fallas importantes encontradas con los recursos asignados. Cada método que se utilice para detectar defectos deja un residuo de defectos más sutiles contra los cuales ese método es ineficaz (la llamada “Paradoja del Pesticida”). La prueba de software implica pues, la aplicación de técnicas y herramientas apropiadas en el marco de un proceso bien definido, determinado por el tipo de proyectos de desarrollo de software que se abordan.

3.4.2 Desarrollo dirigido por pruebas

El desarrollo dirigido por pruebas, se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso para lograr un resultado, aunque no con lógica para el negocio, pero si funcional. Algunas personas confunden este término con las llamadas “pruebas de caja blanca” las cuáles se les practican a los métodos u operaciones para medir la funcionalidad del mismo, desde el punto de vista de validez del cliente. Sin embargo, el TDD se aplica antes de comenzar a implementar cada paso de la tarea en desarrollo, asumiendo que la prueba es insatisfactoria desde un inicio. Sólo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida. Luego se realiza un proceso conocido como “refactorización” de código perteneciente a una de las doce prácticas planteadas por la metodología XP, el cual consiste en mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. Es esencia el TDD, se enfoca en la lógica del negocio y las pruebas de caja blanca en la lógica del negocio.

3.4.3 Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los paradigmas anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación.

Para realizar las pruebas de aceptación el cliente utiliza la siguiente plantilla.

Tabla 9: Plantilla de Pruebas de Aceptación

Prueba de aceptación
HU: Nombre de la historia de usuario que va a comprobar su funcionamiento.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.
Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar.
Entrada/Pasos de ejecución: Pasos para probar la funcionalidad.
Resultado esperado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o Denegada.

Tabla 10: Prueba de Aceptación #1 Crear informe de Peso Específico de los Suelos.

Prueba de aceptación Crear informe de Peso Específico de los Suelos
HU: Crear informe de Peso Específico de los Suelos.
Nombre: Prueba para crear un informe de Peso Específico de los Suelos.
Descripción: El propósito de esta prueba es para determinar si se calcula el peso correctamente, así como lograr imprimir y se guardar en la base de datos.
Condiciones de ejecución: El usuario debe llenar los campos especificados.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, y se llenan los campos mostrados en el formulario con las opciones de calcular, guardar y cancelar.
Resultado esperado: Se muestran los resultados esperados y se logra imprimir el informe además de ser guardado en la base de datos.
Evaluación de la prueba: Aceptada.

Ver Tablas de Pruebas [ANEXO 4](#)

3.5 Conclusiones

En este capítulo se llevó a cabo la fase de desarrollo y prueba donde se presenta el modelo de datos de la aplicación a construir, logrando una visión detallada de sus atributos y las relaciones entre sus clases. Se realizó el desarrollo de las iteraciones a partir de la distribución de tareas por HU, y se les practicó las pruebas de aceptación a las funcionalidades de mayor importancia.

Estudio de Factibilidad

4

4.1 Introducción

En la actualidad con el desarrollo de los proyectos viene incluido el estudio de factibilidad del sistema, el cual es vital pues se tienen en cuenta los costos a incurrir, deduciéndose si el proyecto realizado será factible o no llevarlo a cabo.

Hay muchas formas de calcular el costo, pero para nuestro caso se utilizará la Metodología Costo Efectividad, la cual sugiere que la conveniencia de la ejecución de un proyecto se determina por la observación de ciertos factores en conjunto, estos son:

- El costo que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware / software y los costos de operación asociados.
- La efectividad que se entiende como capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo por el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad del cumplimiento del objetivo).

Esta parte es fundamental en la elaboración de cualquier proyecto pues haciendo un estudio correcto de factibilidad se puede ahorrar semanas, meses e incluso años de trabajo, hasta evitar poner en duda la reputación profesional si se realiza un sistema mal planificado desde una etapa temprana. (Ficha de costo ,2000)

4.2 Efectos Económicos

- Efectos directos
- Efectos indirectos
- Efectos externos
- Intangibles

4.2.1 Efectos directos

POSITIVOS:

- Se optimizará la realización de informes de ensayos.
- Se facilitará la interpretación de los resultados mediante la graficación.
- Se mejorará la eficiencia y explotación del Sistema de Gestión de Informes de Ensayos (SIGIE 1.0).

NEGATIVOS:

Para usar la aplicación es imprescindible el uso de un ordenador, aparejado a los gastos que este trae de consumo de corriente eléctrica y mantenimiento.

4.2.2 Efecto indirecto

Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

4.2.3 Intangibles

En la valoración económica siempre hay elementos perceptibles por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible. A fin de medir con precisión los efectos, deberán considerarse dos situaciones:

4.2.3.1 Situación sin proyecto

El Módulo Peso Específico de los Suelos se realiza en hoja de cálculo de Excel o de forma manual, realizando los cálculos en un período de tiempo excesivamente largo y muchas veces perdiendo la información por no contar con una base de datos potente.

4.2.3.2 Situación con proyecto

Con la realización del Módulo Peso Específico de los Suelos para el Sistema de Gestión de Informes de Ensayos SIGIE 1.1 se ha logrado disminuir el tiempo de realización de un

informe de ensayo además mejora el trabajo de los especialistas en ensayos debido a que esta herramienta facilita los cálculos del experimento.

4.3 Beneficios Y Costos Intangibles en el proyecto

Costos:

- Resistencia al cambio.

Beneficios:

- Mayor comodidad para los usuarios.
- Mejora en la calidad de la información.
- Menor tiempo empleado en la introducción de los datos.
- Facilidad a la hora de interpretar los resultados.

4.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar Una Ficha De Costo de un Producto Informático [Dra. Ana Ma. Gracia Pérez, UCLV]. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

4.4.1 Costos en Moneda Libremente Convertible

Costos Directos.

1. Compra de equipos de cómputo: No procede.
2. Alquiler de equipos de cómputo: No procede.
3. Compra de licencia de Software: No procede.
4. Depreciación de equipos: \$ 50.00.
5. Materiales directos: No procede.

Total: \$ 50.00.

Costos Indirectos.

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Total: \$0.00.

Gastos de distribución y venta.

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00.

4.4.2 Costos en Moneda Nacional

Costos Directos.

1. Salario del personal que laborará en el proyecto: \$100.00.
2. El 12% del total de gastos por salarios se dedica a la seguridad social: No procede.
3. El 0.09% de salario total, por concepto de vacaciones a acumular: No procede.
4. Gasto por consumo de energía eléctrica: \$ 5.50.
5. Gastos en llamadas telefónicas: No procede.
6. Gastos administrativos: No procede.

Costos Indirectos.

1. Know How: No procede.

Total: \$ 105.50.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado desde que se introducen los datos hasta el tiempo en que se muestran los resultados y el gráfico de comportamiento.

Valores de la variable (Solución manual):

- Gestionar los datos de Peso Específico de los Suelos por la vicedecana de la facultad de Geología en los laboratorios de Mecánica de Suelo y Rocas del ISMMM (60 min), este excesivo tiempo es producido porque el personal calificado en llenar todos los campos donde se almacenan los datos que son recogidos por los especialistas en el estudio de los suelos, lo realizan de forma manual.
- Calcular los ensayos e interpretar los resultados de este proceso (30 min) provocado por la cantidad de cálculos realizados de forma manual o con el uso de algún accesorio como son las calculadoras.
- Gestionar los datos de Límites de Plasticidad por la vicedecana de la facultad de Geología en los laboratorios de Mecánica de Suelos y Rocas del ISMMM (50 min), por la gran cantidad de datos que son copiados en las prácticas de campo por los especialistas y hace engorroso el trabajo en el laboratorio.
- Calcular los ensayos de este proceso (20 min), provocado por la cantidad de cálculos realizados de forma manual o con el uso de algún accesorio como son las calculadoras.

Valores de la variable (Solución con el software):

- Gestionar los datos de Peso Específico de los Suelos por la vicedecana de la facultad de Geología en los laboratorios de Mecánica de Suelo y Rocas del ISMMM (12 min), el tiempo porque el personal calificado en llenar todos los campos donde se almacenan los datos que son recogidos por los especialistas en el estudio de los suelos, solo lo tienes que escribir en la aplicación usando las facilidades de la computadora.
- Calcular los ensayos e interpretar los resultados de este proceso (4 min), una vez llenados todos los campos el especialista solo debe presionar un botón en la aplicación y esta de forma rápida y eficiente realiza todos los cálculos arrojando un resultado final.
- Gestionar los datos de Límites de Plasticidad por la vicedecana de la facultad de Geología en los laboratorios de Mecánica de Suelos y Rocas del ISMMM (10 min), el tiempo porque el personal calificado en llenar todos los campos donde se almacenan los datos que son recogidos por los especialistas en el estudio de los suelos, solo lo tienes que escribir en la aplicación usando las facilidades de la computadora.
- Calcular los ensayos de este proceso (3 min), una vez llenados todos los campos el especialista solo debe presionar un botón en la aplicación y esta de forma rápida y eficiente realiza todos los cálculos arrojando un resultado final.

A continuación se muestra un gráfico que muestra la solución de las actividades sin proyecto comparado con la solución con proyecto. Tiempo de la realización de las actividades.

Tiempo de realización de las actividades

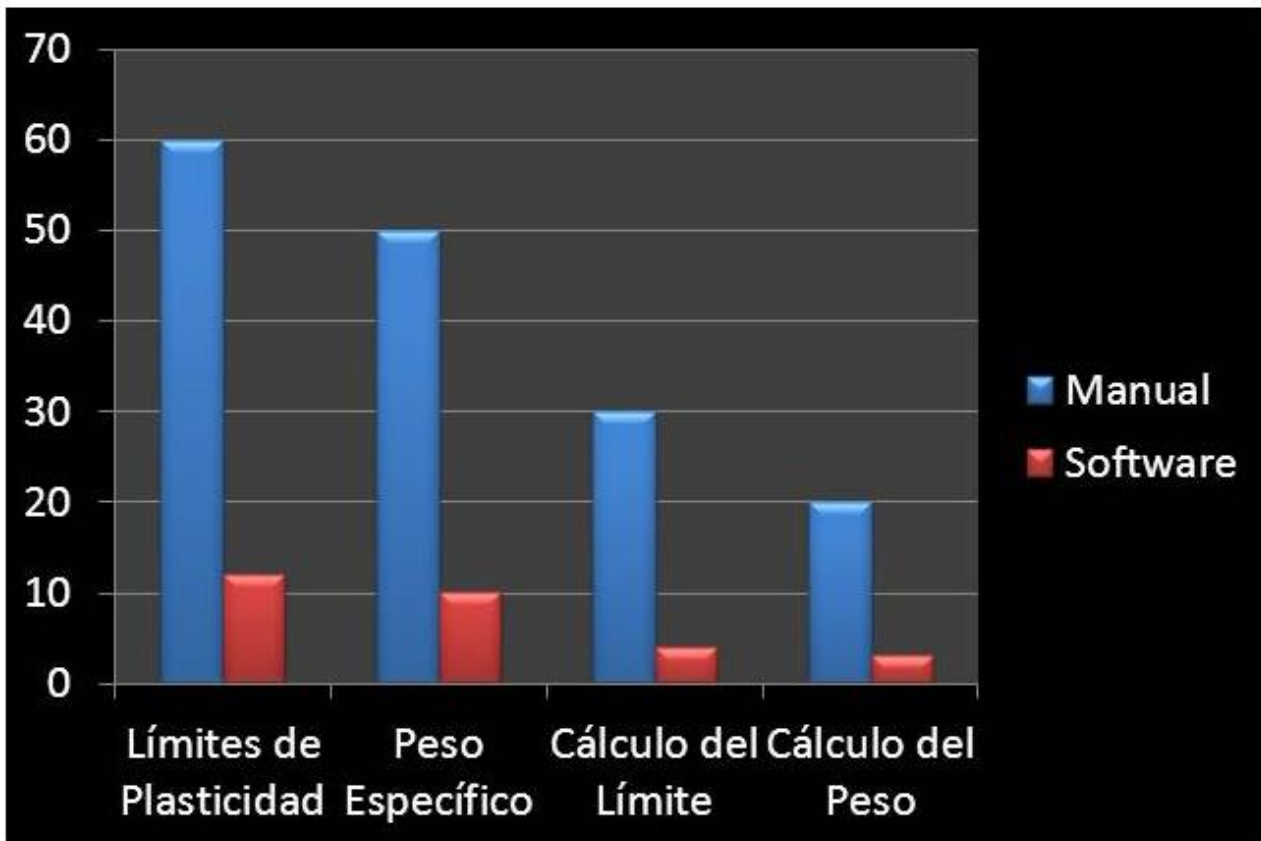


Figura 3: Gráfica de la solución sin el producto y solución con el producto.

4.5 Conclusiones

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo Efectividad (Beneficio), se analizó los efectos económicos, los beneficios y costos intangibles, así como se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$50.00 CUC y \$105.50 CUP, además la solución con el software disminuye de forma considerable el tiempo de realización de los informes y la búsqueda de resultados anteriores. También se reduce el uso innecesario de papel y otros insumos de oficina. Demostrándose la factibilidad económica del proyecto.

Conclusiones

Durante la realización de este trabajo se dio cumplimiento a los objetivos específicos concebidos para el desarrollo de la implementación del módulo, dando así cumplimiento al objetivo general del trabajo.

Con el desarrollo de este trabajo se determinó:

- El análisis y comprensión del funcionamiento e interpretación de los datos generados por los informes de ensayos Peso Específico de los Suelos y Límites de Plasticidad.
- El Diseño e Implementación del módulo Peso Específico de los Suelos.
- Se comprobó la factibilidad del software
- Las pruebas de aceptación arrojaron resultados satisfactorios por parte de los clientes.

De manera general:

Se especificó la metodología XP para la elaboración del sistema, en las que se les dio cumplimiento a las Historias de Usuarios del sistema, se realizaron las Tarjetas CRC, las Tareas de Ingenierías para cada Historia de Usuario y las Pruebas de Aceptación realizadas por el cliente. Luego del Análisis de Factibilidad podemos concluir que el trabajo responde como propuesta de solución al problema planteado. A demás el sistema se encuentra disponible para ser usado.

Recomendaciones

De manera general los objetivos trazados al inicio de esta trabajo han sido logrados, al mismo tiempo, en el transcurso del proceso de desarrollo, ha quedado evidenciado, que la propuesta es sólo la primera fase de un proyecto que puede ser mucho más ambicioso.

Por tanto se hacen las siguientes recomendaciones:

1. Perfeccionar e incorporar nuevas funcionalidades a las búsquedas.
2. Se recomienda la explotación de las funcionalidades que brinda el software en los Laboratorios de Mecánica del Suelo y Rocas del Instituto Superior Minero Metalúrgico de Moa.
3. Que se implementen más funcionalidades al sistema además de crear una nueva versión la cual abarque los diferentes tipos de informes que se realizan en el Instituto Superior Minero Metalúrgico de Moa (ISMMM).
4. Que se pueda acceder a las nuevas funcionalidades implementadas sin tener que cargar la aplicación completa.
5. Desarrollar un sistema modular que admita un crecimiento continuo.

Referencias Bibliográficas

1. **González de Vallejo, Luis I., y otros.** *Ingeniería Geológica*. Madrid : Pearson Educación, 2002. ISBN 84-205-3104-9.
2. **Badillo, J.E., A.R. Rodríguez.** 1969. *Mecánica de suelos (I)*. Revolucionaria. La Habana : s.n., 1969.
3. **CAVSI.** 2004. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD?
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.<http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos>. [En línea] 2004.
4. **Freedownloadmanager.org.** 2004. Visual Paradigm for UML.
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3\)](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3)). [En línea] 2004.
5. **Guerra, Maité Sosa Veranes y Vladimir Gonzales.** 2009. Tesis "Sistema de Gestión de la Facultad 2". 2009. págs. 38 y 50 - 54.
6. **Gutiérrez, Jorge A. Saavedra.** 2007. El Mundo Informático. Software Libre.
<http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>. [En línea] 2007.
7. **Roche, Katusca Jiménez.** 2010. Tesis "Obtención del Modelo geométrico para el diseño y explotación de canteras de materiales de construcción.". 2010. págs. 17 - 18.
8. **Sandó, Lesther Delgado Pérez y José Rolando Pérez.** 2009. Tesis "Herramientas para la revisión y seguimiento de errores de documentación en los Proyectos de la Facultad 7 ". 2009. págs. 25 - 30.
9. **Solís Álvarez, Camilo Javier and Figueroa Díaz, Roberth Gustavo.** 2005. *Metodologías Tradicionales vs. Metodologías Ágiles*. 2005.
10. **Subirós, Dariel Raúl.** Junio 2009. Tesis "Desarrollo de una interfaz gráfica de usuario para el Procesador Meteorológico". Junio 2009. pág. 9 y 15.
11. **Caquot A. Kreisel:** Tratado de mecánica de suelo. Inteciencia, 1966.
12. **Sowers, B. G. F. Sowers:** Introducción a la mecánica de suelos y cimentaciones. Edición Revolucionaria, La Habana, 1975. () (MarcadorDePosición2).
13. **Cortina, Robin Perdomo.** Junio 2012: Sistema de Gestión de Informes de Ensayos, Moa, 2012.
14. **Batista, Acel Hernández:** Sistema de Gestión de Informes de Ensayos, Moa, 2012.

Bibliografía

1. **González de Vallejo, Luis I., y otros.** *Ingeniería Geológica*. Madrid : Pearson Educación, 2002. ISBN 84-205-3104-9.
2. **Badillo, J.E., A.R. Rodríguez.** 1969. *Mecánica de suelos (I)*. Revolucionaria. La Habana : s.n., 1969.
3. **CAVSI.** 2004. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD?
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.<http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos>. [En línea] 2004.
4. **Freedownloadmanager.org.** 2004. Visual Paradigm for UML.
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3\)](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3)). [En línea] 2004.
5. **Guerra, Maité Sosa Veranes y Vladimir Gonzales.** 2009. Tesis "Sistema de Gestión de la Facultad 2". 2009. págs. 38 y 50 - 54.
6. **Gutiérrez, Jorge A. Saavedra.** 2007. El Mundo Informático. Software Libre.
<http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>. [En línea] 2007.
7. **Roche, Katusca Jiménez.** 2010. Tesis "Obtención del Modelo geométrico para el diseño y explotación de canteras de materiales de construcción.". 2010. págs. 17 - 18.
8. **Sandó, Lesther Delgado Pérez y José Rolando Pérez.** 2009. Tesis "Herramientas para la revisión y seguimiento de errores de documentación en los Proyectos de la Facultad 7 ". 2009. págs. 25 - 30.
9. **Solís Álvarez, Camilo Javier and Figueroa Díaz, Roberth Gustavo.** 2005. *Metodologías Tradicionales vs. Metodologías Ágiles*. 2005.
10. **Subirós, Dariel Raúl.** Junio 2009. Tesis "Desarrollo de una interfaz gráfica de usuario para el Procesador Meteorológico". Junio 2009. pág. 9 y 15.
11. **Caquot A. Kreisel:** Tratado de mecánica de suelo. Interciencia, 1966.
12. **Sowers, B. G. F. Sowers:** Introducción a la mecánica de suelos y cimentaciones. Edición Revolucionaria, La Habana, 1975. () (MarcadorDePosición2).
13. **Cortina, Robin Perdomo.** Junio 2012: Sistema de Gestión de Informes de Ensayos, Moa, 2012.
14. **Batista, Acel Hernández:** Sistema de Gestión de Informes de Ensayos, Moa, 2012.

Glosario de Términos

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

IDE: Integrated Development Environment / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Iteraciones: En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio. Múltiples iteraciones contribuyen a crear un producto completamente integrado.

Metodología Ágil: Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

Metodologías de Desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

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.

Programación Extrema(XP): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Release: Versión candidata definitiva de un producto de software y se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

Software: Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente, aunque conserve su carácter de libre, puede ser vendido comercialmente.

Testing: Proceso de pruebas usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Picnómetro: Es un frasco volumétrico, con una capacidad de 50 cc. La tapa deberá ser del mismo material del picnómetro y deberá entrar con facilidad hasta la profundidad indicada. Además deberá tener un hueco en el centro para permitir la salida del aire y del agua en exceso.

Anexo 1

Tabla 1.1: H.U. 2 Buscar informe de Peso Específico de los Suelos

Historia de Usuario	
Número: 2	Usuario: Beatriz Riverón
Nombre historia: Buscar Informe de Peso Específico de los Suelos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra.	

Tabla 1.2: H.U. 3 Actualizar informe de Peso Específico de los Suelos

Historia de Usuario	
Número: 3	Usuario: Beatriz Riverón
Nombre historia: Actualizar Informe de Peso Específico de los Suelos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra, y se llenan los campos mostrados en el formulario con las opciones de calcular, actualizar y cancelar.	
Observaciones: Para poder actualizar un informe primero se debe verificar que el informe este en la base de datos de no estar se muestra un mensaje especificando que no existe.	

Tabla 1.3: H.U. 4 Listar informe de Peso Específico de los Suelos

Historia de Usuario	
Número: 4	Usuario: Beatriz Riverón
Nombre historia: Listar Informes de Peso Específico de los Suelos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar. Luego se muestra un formulario con la lista de informes que se encuentran guardados en la base de datos.	

Tabla 1.4: H.U. 5 Eliminar informe de Peso Específico de los Suelos

Historia de Usuario	
Número: 5	Usuario: Beatriz Riverón
Nombre historia: Eliminar Informe de Peso Específico de los Suelos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar, apareciendo una lista de los informes que se encuentran guardados en la base de datos, el usuario selecciona el informe a eliminar y presiona la opción de eliminar.	
Observaciones: Para eliminar un informe se selecciona y se acepta luego se muestra un mensaje de confirmación para la eliminación segura.	

Tabla 1.5: H.U. 6 Crear informe de Límites de Plasticidad

Historia de Usuario	
Número: 6	Usuario: Beatriz Riverón
Nombre historia: Crear informe de Límites de Plasticidad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Arid Sofía Hernández Garcell	
Descripción: El usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, y se llenan los campos mostrados en el formulario con las opciones de calcular, guardar, graficar y cancelar.	
Observaciones: Para poder realizar la funcionalidad calcular es necesario que los campos especificados estén llenos y con los tipos de datos requeridos de lo contrario se mostrarán mensajes indicando el tipo de error.	

Tabla 1.6: H.U. 7 Buscar informe de Límites de Plasticidad

Historia de Usuario	
Número: 7	Usuario: Beatriz Riverón
Nombre historia: Buscar Informe de Límites de Plasticidad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra.	

Tabla 1.7: H.U. 8 Actualizar informe de Límites de Plasticidad

Historia de Usuario	
Número: 8	Usuario: Beatriz Riverón
Nombre historia: Actualizar Informe de Límites de Plasticidad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra, y se llenan los campos mostrados en el formulario con las opciones de calcular, actualizar y cancelar.	
Observaciones: Para poder actualizar un informe primero se debe verificar que el informe este en la base de datos de no estar se muestra un mensaje especificando que no existe.	

Tabla 1.8: H.U. 9 Listar informe de Límites de Plasticidad

Historia de Usuario	
Número: 9	Usuario: Beatriz Riverón
Nombre historia: Listar Informes de Límites de Plasticidad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar. Luego se muestra un formulario con la lista de informes que se encuentran guardados en la base de datos.	

Tabla 1.9: H.U. 10 Eliminar informe de Límites de Plasticidad

Historia de Usuario	
Número: 10	Usuario: Beatriz Riverón
Nombre historia: Eliminar Informe de Límites de Plasticidad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Arid Sofía Hernández Garcell.	
Descripción: El usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar, apareciendo una lista de los informes que se encuentran guardados en la base de datos, el usuario selecciona el informe a eliminar y presiona la opción de eliminar.	
Observaciones: Para eliminar un informe se selecciona y se acepta luego se muestra un mensaje de confirmación para la eliminación segura.	

Anexo 2

Realizado	
Description: Guarda los datos de la personas que realizaron el informe	
Attributes:	
Name	Description
codigo	Lave de la tabla datosgenerales
nom_realiza	Nombre del que realiza el ensayo
nom_reviza	Nombre del que reviza y aprueba el ensayo
nom_calcula	Nombre del que clcula el ensayo
cargo_realiza	Cargo del que realiza el ensayo
cargo_calcula	Cargo del que realiza los calculos
cargo_reviza	Cargo del que reviza el ensayo
Responsibilities:	
Name	Collaborator
Insertar_Realizado()	datosgenerales
Modificar_Realizado()	
Eliminar_Realoizado()	

Figura 2.1 Tarjeta CRC #2 realizado

resultadoindiceplasticidad	
Description: Guardar Informacion resultadoindiceplasticidad	
Attributes:	
Name	Description
codigo	
limiteliquido	
limiteplastico	
Responsibilities:	
Name	Collaborator
Crear resultadoindiceplasticidad	datosgenerales
Modificar resultadoindiceplasticidad	
Eliminar resultadoindiceplasticidad	
Buscar resultadoindiceplasticidad	
Listar resultadoindiceplasticidad	

Figura 2.2 Tarjeta CRC #3 resultado índice plasticidad

humedo	
Description: Guardar información humedo	
Attributes:	
Name	Description
codigo	
sumergido1	
sumergido2	
sumergido3	
lecturavertical21	
lecturavertical22	
lecturavertical23	
lecturahorizontal21	
lecturahorizontal22	
lecturahorizontal23	
tan21	
tan22	
tan23	
asumergido1	
asumergido2	
asumergido3	
Responsibilities:	
Name	Collaborator
Crear humedo	datosgenerales
Modificar humedo	
Eliminar humedo	
Buscar humedo	
Listar humedo	

Figura: 2.3 Tarjeta CRC #4 húmedo

límitelíquido	
no recipiente3	
no recipiente4	
no recipiente5	
masahumedarecipiente1	
masahumedarecipiente2	
masahumedarecipiente3	
masahumedarecipiente4	
masahumedarecipiente5	
masasecarecipiente1	
masasecarecipiente2	
masasecarecipiente3	
masasecarecipiente4	
masasecarecipiente5	
masasrecipiente1	
masasrecipiente2	
masasrecipiente3	
masasrecipiente4	
masasrecipiente5	
himedades1	
himedades2	
himedades3	
himedades4	
himedades5	
Responsibilities:	
Name	Collaborator
Crear límitelíquido	datosgenerales
Modificar límitelíquido	
Eliminar límitelíquido	
Buscar límitelíquido	
Listar límitelíquido	

Figura: 2.4 Tarjeta CRC #5 límite líquido

limiteplastico	
Description: Guardar información limiteplastico	
Attributes:	
Name	Description
codigo	
norecipienteinicial	
norecipientefinal	
masahumedarecipienteinicial	
masahumedarecipientefinal	
masasecarecipienteinicial	
masasecarecipientefinal	
masarecipienteinicial	
masarecipientefinal	
humedadinicial	
humedadfinal	
humedadpromedio	
Responsibilities:	
Name	Collaborator
Crear limiteplastico	datosgenerales
Modificar limiteplastico	
Eliminar limiteplastico	
Buscar limiteplastico	
Listar limiteplastico	

Figura 2.5 Tarjeta CRC #6 límite plástico

seco	
Description: Guardar información humedades	
Attributes:	
Name	Description
codigo	
seco1	
seco2	
seco3	
lecturavertical1	
lecturavertical2	
lecturavertical3	
lecturahorizontal1	
lecturahorizontal2	
lecturahorizontal3	
tan1	
tan2	
tan3	
aseco1	
aseco2	
aseco3	
Responsibilities:	
Name	Collaborator
Crear seco	datosgenerales
Modificar seco	
Eliminar seco	
Buscar seco	
Listar seco	

Figura 2.6 Tarjeta CRC #7 seco

Picnómetro	
Descripción: Guarda información del Picnómetro del Peso Específico de los Suelos	
Attributes:	
Nombre	Descripción
picnómetro_1	
picnómetro_2	
temperatura_1	
temperatura_2	
densidad_agua_1	
densidad_agua_2	
constante_1	
constante_2	
pesafiltro_1	
pesafiltro_2	
masa_seca_mas_masa_pesafiltro_1	
masa_seca_mas_masa_pesafiltro_1	
masa_pesafiltro_1	
masa_pesafiltro_2	
masa_seca_1	
masa_seca_2	
masa_picnómetro_agua_1	
masa_picnómetro_agua_2	
masa_picnómetro_agua_mas_suelo_1	
masa_picnómetro_agua_mas_suelo_2	
peso_especifico_suelo_1	
peso_especifico_suelo_2	
peso_especifico_promedio	
Responsabilidades:	
Nombre	Collaborator
Crear_picnómetro()	datosgenerales
Buscar_picnómetro()	

Figura 2.7 Tarjeta CRC #8Picnómetro

Anexo 3

Tabla 3.1: Tarea #2 Calcular informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 2	Número de Historia: 1
Nombre de la Tarea: Calcular Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 06/febrero/2013.	Fecha Fin: 10/febrero/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se crea un Informe de Peso Específico de los Suelos en el cual se introducen los datos y luego se calcula el peso de la muestra.	

Tabla 3.2: Tarea #3 Imprimir informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 3	Número de Historia: 1
Nombre de la Tarea: Imprimir informe de Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.5
Fecha de Inicio: 11/febrero/2013.	Fecha Fin: 15/febrero/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Después de haber calculado el peso de la se procede a realizar el impresión del informe.	

Tabla 3.3: Tarea #4: Guardar Informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 4	Número de Historia: 1
Nombre de la Tarea: Guardar informe de Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.5
Fecha de Inicio: 16/febrero/2013.	Fecha Fin: 20/febrero/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Una vez calculado se procede a guardar el informe terminado en la base de datos.	

Tabla 3.4: Tarea #5 Buscar informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 5	Número de Historia: 2
Nombre de la Tarea: Buscar informe de Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2
Fecha de Inicio: 21/febrero/2013.	Fecha Fin: 01/marzo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se busca un informe por el nombre de la obra y se muestra un formulario con los datos de esa obra.	

Tabla 3.5: Tarea #6: Actualizar informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 6	Número de Historia: 3
Nombre de la Tarea: Actualizar informe de Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2
Fecha de Inicio: 02/marzo/2013.	Fecha Fin: 10/marzo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se realiza una búsqueda del peso deseado por el nombre de la obra mostrando una Interfaz con los datos creados, de esta forma modificamos y continuamos con el cálculo e impresión de los resultados modificados.	

Tabla 3.6: Tarea #7 Actualizar informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 7	Número de Historia: 4
Nombre de la Tarea: Listar informe de Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 11/marzo/2013.	Fecha Fin: 17/marzo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se muestra una Interfaz con la Lista existente de Peso Específico de los Suelos.	

Tabla 3.7: Tarea #8: Eliminar informe de Peso Específico de los Suelos

Tareas	
Número de la Tarea: 8	Número de Historia: 5
Nombre de la Tarea: Eliminar Informe Peso Específico de los Suelos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 18/marzo/2013.	Fecha Fin: 25/marzo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se muestra un formulario con una lista existente de Peso Específico de los Suelos el cual al seleccionar uno se activa un botón "Eliminar". Cuando seleccionamos en él se elimina correctamente este peso.	

Tabla 3.8: Tarea #9 Crear Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 9	Número de Historia: 6
Nombre de la Tarea: Crear Informe de Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 26/marzo/2013.	Fecha Fin: 03/abril/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se muestra un formulario con los campos a llenar y se crea el informe.	

Tabla 3.9: Tarea #10 Calcular Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 10	Número de Historia: 6
Nombre de la Tarea: Calcular Informe de Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.5
Fecha de Inicio: 04/abril/2013.	Fecha Fin: 10/abril/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se crea un Informe de Límites de Plasticidad en el cual se introducen los datos y luego se calcula el Límite de la muestra.	

Tabla 3.10: Tarea #11 Graficar Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 11	Número de Historia: 6
Nombre de la Tarea: Graficar Informe de Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.5
Fecha de Inicio: 15/abril/2013.	Fecha Fin: 20/abril/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se crea un Informe de Límites de Plasticidad en el cual se introducen los datos y después de calculado el límite de la muestra se presiona la opción de graficar y el sistema muestra la gráfica de número de golpes contra humedades.	

Tabla 3.11: Tarea #12 Imprimir Informe de Límites de Plasticidad

Tareas	
Número de la Tarea: 12	Número de Historia: 6
Nombre de la Tarea: Imprimir Informe de Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 21/abril/2013.	Fecha Fin: 26/abril/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Después de haber calculado el límite de la muestra y realizado el gráfico del comportamiento de esta se procede a realizar el impresión del informe.	

Tabla 3.12: Tarea #13 Guardar Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 13	Número de Historia: 6
Nombre de la Tarea: Guardar Informe de Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 27/abril/2013.	Fecha Fin: 02/mayo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Una vez calculado, imprimido y realizado el gráfico se procede a guardar el informe terminado en la base de datos.	

Tabla 3.13: Tarea #14 Buscar Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 14	Número de Historia: 7
Nombre de la Tarea: Buscar Informe Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 03/mayo/2013.	Fecha Fin: 06/mayo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se busca un informe por el nombre de la Obra y se muestra un formulario con los datos de esa Obra.	

Tabla 3.14: Tarea #15 Actualizar Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 15	Número de Historia: 8
Nombre de la Tarea: Actualizar Informe Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 07/mayo/2013.	Fecha Fin: 11/mayo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se realiza una búsqueda del Límite deseado por el nombre de Obra mostrando una Interfaz con los datos creados, de esta forma modificamos y continuamos con el cálculo, la graficación e impresión de los resultados modificados.	

Tabla 3.15: Tarea #16 Listar Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 16	Número de Historia: 9
Nombre de la Tarea: Listar Informe Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 12/mayo/2013.	Fecha Fin: 16/mayo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se muestra una Interfaz con la Lista existente de Límites de Plasticidad.	

Tabla 3.16 Tarea #17 Eliminar Informe de Límites de Plasticidad.

Tareas	
Número de la Tarea: 17	Número de Historia: 10
Nombre de la Tarea: Eliminar Informe Límites de Plasticidad.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1
Fecha de Inicio: 17/marzo/2013.	Fecha Fin: 22/marzo/2013.
Programador Responsable: Arid Sofía Hernández Garcell.	
Descripción: Se muestra un formulario con una lista existente de Límites de Plasticidad el cual al seleccionar uno se activa un botón "Eliminar". Cuando seleccionamos en él se elimina correctamente este Límite.	

Anexo 4

Tabla 4.1 Prueba de Aceptación #2 Buscar informe de Peso Específico de los Suelos

Prueba de aceptación Buscar informe de Peso Específico de los Suelos
HU: Buscar informe de Peso Específico de los Suelos.
Nombre: Prueba para buscar un informe de Peso Específico de los Suelos.
Descripción: El propósito de esta prueba es para determinar si se busca un informe que se encuentra en la base de datos.
Condiciones de ejecución: -El usuario debe escribir en el formulario el nombre de la obra. -El usuario presiona la opción Buscar.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra.
Resultado esperado: Se muestran los datos del informe que fue seleccionado.
Evaluación de la prueba: Aceptada.

Tabla 4.2: Prueba de Aceptación #3 Actualizar informe de Peso Específico de los Suelos

Prueba de aceptación Actualizar informe de Peso Específico de los Suelos
HU: Actualizar informe de Peso Específico de los Suelos.
Nombre: Prueba para modificar un informe de Peso Específico de los Suelos.
Descripción: El propósito de esta prueba es para determinar si se modifica un informe que se encuentre en la base de datos.
Condiciones de ejecución: <ul style="list-style-type: none">- El usuario debe especificar el informe a modificar- El informe debe existir en la base de datos
Entrada/Pasos de ejecución: Primeramente El usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra, y se llenan los campos mostrados en el formulario con las opciones de calcular, actualizar y cancelar.
Resultado esperado: Se modifican los datos del informe en la base de datos
Evaluación de la prueba: Aceptada.

Tabla 4.3: Prueba de Aceptación #4 Listar informe de Peso Específico de los Suelos

Prueba de aceptación Listar informes de Peso Específico de los Suelos
HU: Listar informe de Peso Específico de los Suelos.
Nombre: Prueba para listar los informes de Peso Específico de los Suelos.
Descripción: El propósito de esta prueba es para determinar si se listan todos los informes que se encuentran en la base de datos.
Condiciones de ejecución: El usuario presiona la opción Eliminar Informe de Peso Específico de los Suelos y marca el botón mostrar.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar. Luego se muestra un formulario con la lista de informes que se encuentran guardados en la base de datos.
Resultado esperado: Se muestran todos los informes que se encuentran en la base de datos.
Evaluación de la prueba: Aceptada.

Tabla 4.4: Prueba de Aceptación #5 Eliminar informe de Peso Específico de los Suelos

Prueba de aceptación Eliminar informe de Peso Específico de los Suelos
HU: Eliminar informe de Peso Específico de los Suelos.
Nombre: Prueba para eliminar un informe de Peso Específico de los Suelos que se encuentre en la base de datos.
Descripción: El propósito de esta prueba es para determinar si se elimina correctamente un informe que se encuentra en la base de datos.
Condiciones de ejecución: <ul style="list-style-type: none">- El usuario selecciona de la lista el informe que desea eliminar.- Presiona la opción Eliminar Informe de Peso Específico de los Suelos.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Peso Específico de los Suelos. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar, apareciendo una lista de los informes que se encuentran guardados en la base de datos, el usuario selecciona el informe a eliminar y presiona la opción de eliminar.
Resultado esperado: Se eliminan todos los datos del informe que fue seleccionado.
Evaluación de la prueba: Aceptada.

Tabla 4.5: Prueba de Aceptación #6 Crear informe de Límites de Plasticidad

Prueba de aceptación Crear informe de Límites de Plasticidad.
HU: Crear informe de Límites de Plasticidad.
Nombre: Prueba para crear un informe de Límites de Plasticidad.
Descripción: El propósito de esta prueba es para determinar si se calcula el peso correctamente, así como lograr imprimir y se guardar en la base de datos.
Condiciones de ejecución: El usuario debe llenar los campos especificados.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, y se llenan los campos mostrados en el formulario con las opciones de calcular, guardar y cancelar.
Resultado esperado: Se muestran los resultados esperados y se logra imprimir el informe además de ser guardado en la base de datos.
Evaluación de la prueba: Aceptada.

Tabla 4.6: Prueba de Aceptación #7 Buscar informe de Límites de Plasticidad.

Prueba de aceptación Buscar informe de Límites de Plasticidad
HU: Buscar informe de Límites de Plasticidad.
Nombre: Prueba para buscar un informe de Límites de Plasticidad.
Descripción: El propósito de esta prueba es para determinar si se busca un informe que se encuentra en la base de datos.
Condiciones de ejecución: -El usuario debe escribir en el formulario el nombre de la obra. -El usuario presiona la opción Buscar.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra.
Resultado esperado: Se muestran los datos del informe que fue selecciona do en la lista de informes.
Evaluación de la prueba: Aceptada.

Tabla 4.7: Prueba de Aceptación #8 Actualizar informe de Límites de Plasticidad.

Prueba de aceptación Actualizar informe de Límites de Plasticidad
HU: Actualizar informe de Límites de Plasticidad.
Nombre: Prueba para modificar un informe de Límites de Plasticidad
Descripción: El propósito de esta prueba es para determinar si se modifica un informe que se encuentre en la base de datos.
Condiciones de ejecución: <ul style="list-style-type: none">- El usuario debe especificar el informe a modificar- El informe debe existir en la base de datos
Entrada/Pasos de ejecución: Primeramente El usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar, se marca la opción actualizar. Luego se busca el informe por el nombre de la obra, y se llenan los campos mostrados en el formulario con las opciones de calcular, actualizar y cancelar.
Resultado esperado: Se modifican los datos del informe en la base de datos
Evaluación de la prueba: Aceptada.

Tabla 4.8: Prueba de Aceptación #9 Listar informe de Límites de Plasticidad.

Prueba de aceptación Listar informes de Límites de Plasticidad
HU: Listar informe de Límites de Plasticidad.
Nombre: Prueba para listar los informes de Límites de Plasticidad
Descripción: El propósito de esta prueba es para determinar si se listan todos los informes que se encuentran en la base de datos.
Condiciones de ejecución: El usuario presiona la opción Eliminar Informe de Límite de plasticidad y marca el botón mostrar.
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar. Luego se muestra un formulario con la lista de informes que se encuentran guardados en la base de datos.
Resultado esperado: Se muestran los datos de todos los informes que se encuentran en la base de datos
Evaluación de la prueba: Aceptada.

Tabla 4.9: Prueba de Aceptación #10 Eliminar informe de Límites de Plasticidad

Prueba de aceptación Eliminar informe de Límites de Plasticidad
HU: Eliminar informe de Límites de Plasticidad.
Nombre: Prueba para eliminar un informe de Límites de Plasticidad que se encuentre en la base de datos.
Descripción: El propósito de esta prueba es para determinar si se elimina correctamente un informe que se encuentra en la base de datos.
Condiciones de ejecución: <ul style="list-style-type: none">- El usuario selecciona de la lista el informe que desea eliminar- Presiona la opción Eliminar Informe de Límites de Plasticidad
Entrada/Pasos de ejecución: Primeramente el usuario selecciona en la interfaz principal la opción Límites de Plasticidad. Luego se muestra un formulario con las opciones Crear-Actualizar y Eliminar. Se elige la opción Eliminar, dentro de esta se marca el botón Mostrar, apareciendo una lista de los informes que se encuentran guardados en la base de datos, el usuario selecciona el informe a eliminar y presiona la opción de eliminar.
Resultado esperado: Se eliminan todos los datos del informe que fue seleccionado en la lista de informes.
Evaluación de la prueba: Aceptada.