



Universidad de las Ciencias
Informáticas

Universidad de Las Ciencias Informáticas

Facultad 2

Título:

***Herramienta Web para la Gestión de Auditorías a
Sistemas gestores de Base de Datos.***

*Trabajo de Diploma Para optar por el título de Ingeniero en
Ciencias Informáticas.*

Autores: Yassel Pérez Camacho

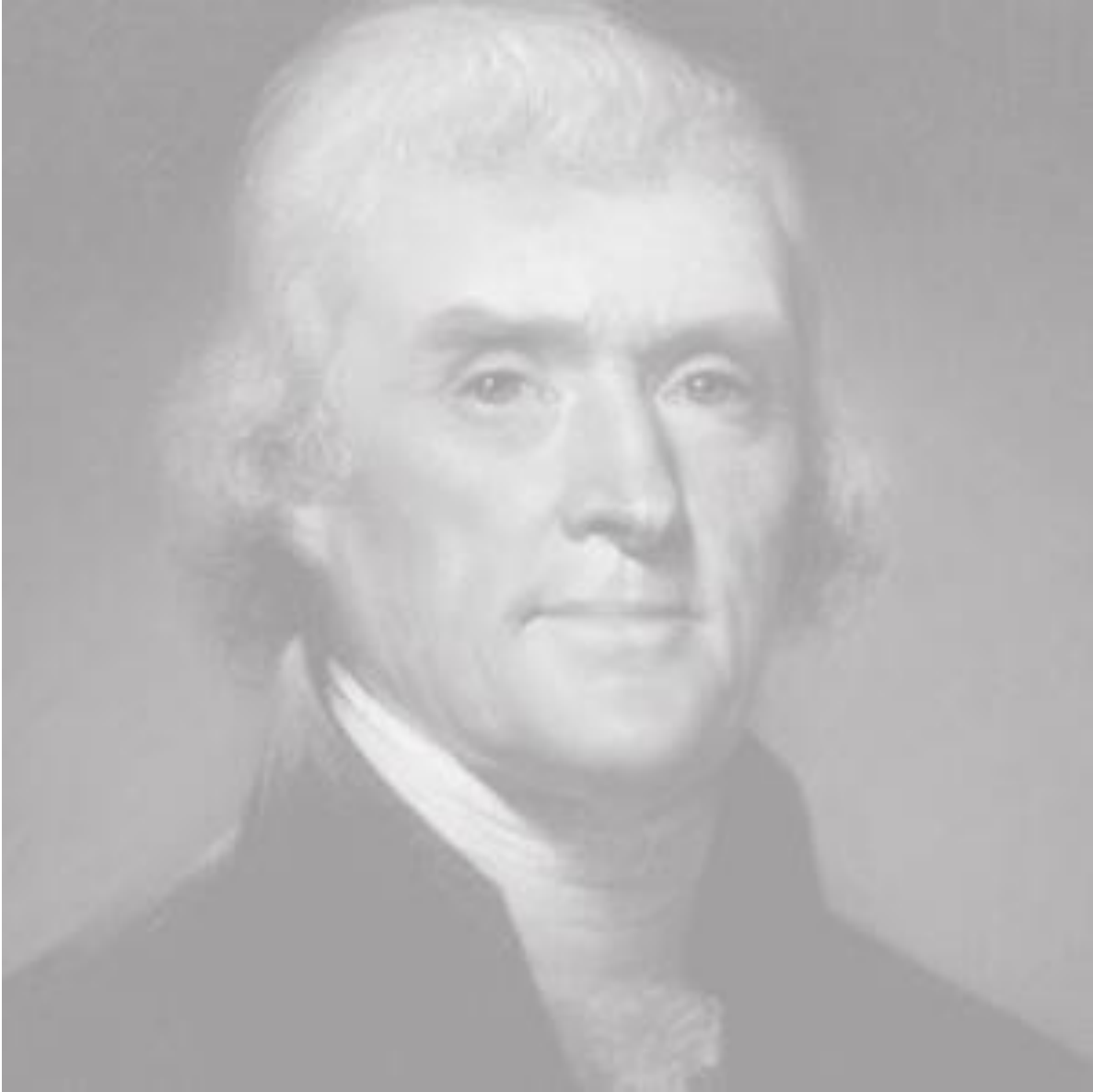
Félix Duque López.

Tutores: Ing. Fernando Ricardo Romero

Ing. Yeilenia Iris Pérez Vázquez

La Habana, junio de 2015

Pensamiento



***“Yo creo bastante en la suerte. Y he constatado que,
cuanto más duro trabajo, más suerte tengo”***

Thomas Jefferson

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yassel Pérez Camacho

Firma del Autor

Félix Duque López

Firma del Autor

Ing. Fernando Ricardo Romero

Firma del Tutor

Ing. Yeilenia Pérez Vázquez

Firma del Co-Tutor

Datos de Contacto

Yassel Pérez Camacho

Correo: ycamacho@estudiantes.uci.cu

Ciudad de la Habana, Cuba

Félix Duque López

Correo: fduque@estudiantes.uci.cu

Ciudad de la Habana, Cuba

Fernando Ricardo Romero

Correo: fricardo@uci.cu

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

Yeilenia Pérez Vázquez

Correo: yiperez@uci.cu

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

Agradecimientos

Yassel:

[Opcional]

Félix:

[Opcional]

Dedicatoria

Yassel:

Dedico la tesis a mis padres, a mi hermano, a mi abuela, a mis tíos y a mis primos que sin ellos no hubiese podido alcanzar todas mis metas.

Félix:

Dedico la tesis a mis padres, mis hermanos, a mis abuelos, a mi novia que sin ellos no hubiese podido alcanzar todas mis metas y objetivos.

Resumen

La Empresa de Telecomunicaciones de Cuba (ETECSA), dedicada a brindar este servicio en todo el país, maneja grandes volúmenes de información, cuya integridad y correcta gestión es indispensable para el funcionamiento de sus actividades comerciales. Con el objetivo de verificar cuán vulnerables son sus servicios, la empresa cuenta con un Departamento de Seguridad informática el cual realiza auditorías a sus Sistemas Informáticos. Con el fin de automatizar el proceso surge un convenio con la Universidad de las Ciencias Informáticas y se desarrolló el Sistema para Realizar Auditorías a Sistemas Gestores de Bases de Datos (SASGBD). Desarrollado en el Centro TLM usando el lenguaje de programación Java, el sistema cuenta con un módulo encargado de las auditorías a Bases de Datos. Se implementó una versión Web del Módulo de Base de Datos (BD) de SASGBD utilizando Python como lenguaje de desarrollo y Django como marco de trabajo. La herramienta permite al igual que el Módulo, la gestión y evaluación de las auditorías a Sistemas Gestores de Bases de Datos. Entre sus funcionalidades está crear un script de revisión usado por el administrador de la BD para realizar la auditoría. También permite importar un script resultado de la auditoría con el cual se conforma un informe con el resultado final de la auditoría. La versión surge debido a la búsqueda de la estandarización tecnológica en la que está inmerso el Centro TLM. Esta nueva versión permite al Centro TLM brindarle una mayor posibilidad de mantenimiento y soporte a la herramienta y así alargar la vida útil del software.

Palabras Clave: Auditoría, Base de Datos, Seguridad Informática.

Índice

Resumen	VI
Introducción.....	1
Capítulo 1 Fundamentación Teórica.....	5
1.1. Introducción.....	5
1.2. Conceptos asociados a la Investigación:.....	5
1.3. Estudio de SASGBD:.....	7
1.4. Metodología de desarrollo de software.....	8
1.5. Herramientas y Tecnologías informáticas:.....	10
1.5.1. Lenguaje de programación del lado del servidor.....	10
1.5.2. Framework de Desarrollo.....	11
1.5.3. Lenguaje de programación del lado del cliente.....	12
1.5.4. Estructura de intercambio de datos entre cliente y servidor.....	12
1.6. IDE (Entorno de Desarrollo Integrado).....	13
1.7. Lenguaje Unificado de Modelado (UML, Unified Modeling Language).....	13
1.8. Notación para el Modelado de Procesos de Negocio (BPMN, Business Process Modeling Notation).....	13
1.9. Herramienta CASE (Computer Aided Software Engineering).....	13
1.10. Gestor de Bases de Datos.....	14
1.11. Conclusiones parciales.....	14
Capítulo 2: Características y Diseño de la Herramienta.....	15
2.1. Introducción.....	15
2.2. Procesos de negocio Gestión de Auditorías a Sistemas Gestores de Bases de Datos.....	15
2.3. Propuesta de Solución.....	16
2.4. Funcionalidades del GASGBD.....	17
2.5. Lista de Reserva de Producto.....	18
2.5.1. Usabilidad.....	18
2.5.2. Hardware.....	18
2.5.3. Software.....	18
2.5.4. Seguridad.....	19
2.5.5. Interfaz de usuario.....	19
2.6. Personas relacionadas.....	19
2.7. Fase de Planificación.....	19
2.7.1. Historias de Usuario.....	19
2.7.2. Estimación de esfuerzo por Historia de Usuario.....	21
2.7.3. Plan de Iteraciones.....	22

2.7.3.1.	Iteración 1:	22
2.7.3.2.	Iteración 2:	22
2.7.3.3.	Iteración 3:	22
2.7.4.	Plan de duración de las Iteraciones	22
2.7.5.	Plan de entrega	23
2.8.	Conclusiones parciales.....	23
Capítulo 3: Diseño, Implementación y Pruebas de la Herramienta		25
3.1.	Introducción.....	25
3.2.	Patrón de arquitectura:	25
3.3.	Capas de la Arquitectura MTV	26
3.4.	Patrones de Diseño.....	28
3.4.1.	GRASP.....	29
3.5.	Tarjetas CRC (Clase-Responsabilidad-Colaborador).....	30
3.6.	Diseño de Base de Datos	31
3.7.	Tareas de Ingeniería.....	33
3.8.	Estándar de Codificación	34
3.8.1.	Python.....	34
3.9.	Pruebas.....	35
3.10.	Pruebas Unitarias.....	36
3.10.1.	Iteración 1	36
3.10.2.	Iteración 2.....	36
3.10.3.	Iteración 3.....	36
3.11.	Pruebas de aceptación.....	37
3.12.	Conclusiones Parciales.....	39
Conclusiones Generales		40
Recomendaciones		41
Referencias		42
Bibliografía		44
Anexos I Muestra de Historias de Usuario.....		45
Anexos II Muestra de Tareas de Ingeniería.....		49
Anexos III Muestra de Pruebas de Aceptación.....		63

Índice de Tablas

TABLA 1. FUNCIONALIDADES.....	18
-------------------------------	----

TABLA 2. HISTORIA DE USUARIO # 1.....	21
TABLA 3. ESTIMACIÓN DE ESFUERZO POR HU.....	22
TABLA 4. PLAN DE ITERACIÓN Y ENTREGA.	23
TABLA 5. PLAN DE ENTREGA DE LA HERRAMIENTA	23
TABLA 6. TARJETA CRC: CLASE GENSCRIPTREV.....	30
TABLA 7. TARJETA CRC: CLASE IMPFICHERESULT.	30
TABLA 8. TARJETA CRC: CLASE REGPARAMETRO.	30
TABLA 9. TARJETA CRC: CLASE REGAPLICACION.	31
TABLA 10. TAREAS DE INGENIERÍA HU# 1.	33
TABLA 11. TAREA 1 DE LA HU# 1.....	34
TABLA 12. PRUEBA DE ACEPTACIÓN #1: HU: REGISTRAR APLICACIÓN.....	39
TABLA 13. HISTORIA DE USUARIO # 2.....	45
TABLA 14. HISTORIA DE USUARIO # 3.....	46
TABLA 15. HISTORIA DE USUARIO # 4.....	47
TABLA 16. HISTORIA DE USUARIO # 5.....	48
TABLA 17. HISTORIA DE USUARIO # 6.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 18. HISTORIA DE USUARIO # 7.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 19. HISTORIA DE USUARIO # 8.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 20. HISTORIA DE USUARIO # 9.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 21. HISTORIA DE USUARIO # 10.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 22. TAREAS DE INGENIERÍA HU# 2.	49
TABLA 23. TAREAS DE INGENIERÍA HU# 3.	50
TABLA 24. TAREAS DE INGENIERÍA HU# 4.	50
TABLA 25. TAREAS DE INGENIERÍA HU# 5.	51
TABLA 26. TAREAS DE INGENIERÍA HU# 6.	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 27. TAREAS DE INGENIERÍA HU# 7.	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 28. TAREAS DE INGENIERÍA HU# 8.	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 29. TAREA 2 DE LA HU# 1.....	51
TABLA 30. TAREA 3 DE LA HU# 1.....	52
TABLA 31. TAREA 4 DE LA HU# 1.....	52
TABLA 32. TAREA 5 DE LA HU# 1.....	53
TABLA 33. TAREA 1 DE LA HU# 2.....	53
TABLA 34. TAREA 2 DE LA HU# 2.....	53
TABLA 35. TAREA 3 DE LA HU# 2.....	54

TABLA 63. PRUEBA DE ACEPTACIÓN #4: HU: REGISTRAR ENTIDAD. ¡ERROR! MARCADOR NO DEFINIDO.	
TABLA 64. PRUEBA DE ACEPTACIÓN #5: HU: IMPORTAR FICHERO DE RESULTADOS.....	63
TABLA 65. PRUEBA DE ACEPTACIÓN #6: HU: CONCLUIR MATRIZ.	65
TABLA 66. PRUEBA DE ACEPTACIÓN #7: HU: IMPORTAR FICHERO DE RESULTADOS.....	66
TABLA 67. PRUEBA DE ACEPTACIÓN #8: HU: GENERAR INFORME.....	66
TABLA 68. PRUEBA DE ACEPTACIÓN #9: HU: CONCLUIR INFORME. ¡ERROR! MARCADOR NO DEFINIDO.	

Índice de Ilustraciones

Ilustración 1 Arquitectura del Módulo de BD.	7
Ilustración 2. Fases de la Metodología XP (5).....	9
Ilustración 3. Ciclo de la Metodología XP. (5).....	9
Ilustración 4. Modelo de Procesos de Negocio: Auditorías de Bases de Datos.....	16
Ilustración 5. Propuesta de Solución.	17
Ilustración 6. Arquitectura MTV.	25
Ilustración 7. Capa Modelo.....	27
Ilustración 8. Capa Vista.	27
Ilustración 9. Capa Plantilla.....	28
Ilustración 10.	28
Ilustración 11. Modelo de BD.	32
Ilustración 12. Ejemplo de Sangría (Python)	35
Ilustración 13. Ejemplo de Importaciones (Python)	35
Ilustración 14. Pruebas Unitarias Iteración 1	36
Ilustración 15. Pruebas Unitarias Iteración 2	36
Ilustración 16. Pruebas Unitarias Iteración 3	36
Ilustración 17: Pruebas de Aceptación	38

Introducción

El avance indetenible de la informática en el nuevo siglo ha permitido que la información se convierta en un activo de altísimo valor para muchos países, que debe protegerse para garantizar su integridad, confidencialidad y disponibilidad. Del mismo modo en que la informática expande sus límites, el auge del internet acorta la distancia entre las personas, provee a los usuarios cada vez de más información, los negocios se benefician del comercio electrónico y de transacciones financieras. Pero aparejado a este desarrollo virtual crece una contra parte dedicada a falsear, robar, alterar y controlar la información almacenada por empresas, organizaciones o gobiernos. En la actualidad, dados los altos niveles de complejidad que han alcanzado los procesos tecnológicos dentro de las empresas y la importancia de los sistemas de información, ha surgido una creciente necesidad de supervisar los sistemas informáticos. (1)

ETECSA, destinada a brindar servicios de telecomunicación a todo el territorio de Cuba, no está exenta de este proceso de supervisión tecnológica que tiene como objetivo velar por la integridad, disponibilidad y confidencialidad de los datos almacenados, así como de que exista una correcta gestión de sus recursos informáticos. Por tanto la empresa cuenta con un Departamento de Seguridad Informática con especialistas dedicados a velar por el resguardo de la información y de los usuarios a los que esta les ofrece servicios. Este departamento realiza auditorías planificadas a los sistemas operativos de los servidores anfitriones, las aplicaciones web que están de cara a los usuarios y a los sistemas que gestionan las bases de datos en todas sus entidades para velar por la integridad de los datos que se almacenan en estos activos informáticos.

Los distintos gestores usados por esta empresa son auditados por el grupo de Seguridad Informática (SI) siguiendo buenas prácticas de usabilidad, buscando errores de configuración así como posibles brechas de seguridad. Estas buenas prácticas son un conjunto de investigaciones realizadas por las compañías desarrolladoras de estos Sistemas Gestores de Bases de Datos (SGBD), así como empresas dedicadas a la SI a nivel internacional certificadas en términos de seguridad en Bases de Datos.

ETECSA se encuentra actualmente en un desarrollo vertiginoso de su infraestructura y la aplicación de tecnologías de punta en vista de igualarse a grandes proveedores de servicios de telecomunicaciones del mundo (2)

El proceso de auditoría implementado por el Departamento de Seguridad Informática de la División de Tecnologías de la Información de ETECSA resultaba engorroso y propenso a errores en diferentes momentos por la no existencia de una herramienta que permitiera la realización del mismo, razón por la que se le solicitó al Centro de Telemática (TLM) de la Universidad de las Ciencias Informáticas que desarrollara una aplicación que permitiera automatizar el proceso.

Como resultado de este convenio el proyecto AuditBD del centro TLM, desarrolló el Sistema de Auditorías a Sistemas Gestores de Bases Datos (SASGBD). SASGBD cuenta con un Módulo para auditar gestores de Bases de Datos y una herramienta portable. Este módulo se encargada de la gestión y evaluación del monitoreo realizado a las Bases de Datos (BD) por la herramienta portable y de conformar un informe final, resultado del proceso de auditoría.

El módulo de BD fue desarrollado con los requerimientos tecnológicos solicitados por el cliente, una herramienta de escritorio usando Java como lenguaje de programación y el framework Spring. Actualmente es utilizado por ETECSA y cumple con los objetivos para los que fue desarrollado. Por otra parte el Centro TLM en busca de una estandarización tecnológica, definió el uso de Python como lenguaje de desarrollo y Django como marco de trabajo. El tener una herramienta desarrollada en un lenguaje que se aparta de las líneas definidas por el Centro, puede traer consecuencias negativas. Teniendo en cuenta nuevas oportunidades de negocio que pudiesen surgir, y con ellas personalizaciones enmarcadas en la satisfacción del cliente, es ventajoso contar con productos de software desarrollados sobre la misma tecnología. Asimismo la capacitación recibida por el personal está enfocada a la estandarización y no al viejo marco de trabajo en el que se desarrolló el producto. De esta forma se logra un mayor aprovechamiento de los recursos humanos existentes en TLM. También es válido señalar que hoy día el personal que inició el desarrollo de SASGBD no forma parte del equipo de desarrollo con que cuenta TLM.

Después del análisis de la situación anterior se formula el siguiente **problema a resolver**: ¿Cómo contribuir a la estandarización tecnológica del Módulo de BD de SASGBD del Centro TLM?

Identificando como **objeto de estudio** Auditorías a Sistemas Gestores de Bases de Datos.

Para dar solución al problema planteado anteriormente se propone como **objetivo general** Adaptar el Módulo de BD de SASGBD para satisfacer la estandarización tecnológica del Centro de Telemática.

Y que el **campo de acción** El Módulo de BD de SASGBD del Centro de Telemática.

De acuerdo a todo lo antes expuesto se propone la siguiente **idea a defender**: El desarrollo del Módulo de BD de SASGBD en una versión web, permitirá contribuir a la estandarización tecnológica llevada a cabo por el Centro TLM.

De este objetivo general se desglosa en los siguientes **objetivos específicos**:

- ❖ Fundamentar la investigación a partir de la definición del marco conceptual alrededor del objeto de estudio.
- ❖ Dar mantenimiento al Módulo de BD de SASGBD para adaptarlo a la estandarización tecnológica en la que está inmerso el Centro Telemática.
- ❖ Diseñar la estrategia de pruebas para la verificación y validación de las funcionalidades definidas para el sistema.

Para desarrollar la aplicación se definieron un conjunto de **tareas de la investigación** que permiten darle solución al objetivo propuesto:

- ❖ Asimilación de la arquitectura tecnológica realizada en el Centro de Telemática como parte de la estandarización para el desarrollo de software.
- ❖ Análisis de los procesos de negocio involucrados en el funcionamiento del Módulo de BD de SASGBD
- ❖ Estudio y selección de las metodología, herramientas y tecnologías necesarias para el desarrollo de la versión web del Módulo de BD de SASGBD.
- ❖ Estudio y revisión de la documentación referente al Módulo de BD de SASGBD.

Durante la investigación y desarrollo se utilizaron diferentes métodos científicos para la obtención, procesamiento y toma de decisiones:

Métodos Empíricos:

- ❖ Observación: Se utiliza en el estudio del Módulo de BD de SASGBD para la comprensión del funcionamiento y procesos de la solución existente.

Métodos Teóricos:

- ❖ Analítico – Sintético: Se utiliza en el proceso de análisis y revisión de artículos y documentos relacionados con el tema, de donde se intentara extraer los elementos esenciales que contribuyan a profundizar más sobre la temática, y a acoplar de manera íntegra las distintas ideas generadas.

El presente trabajo se dividió en 3 capítulos:

- ❖ Capítulo 1 “Fundamentación Teórica”. En este capítulo se describen los principales conceptos a tratar en la investigación. Se define la metodología de desarrollo, herramientas y lenguajes de programación en la implementación de la herramienta.
- ❖ Capítulo 2 “Características y Diseño de la Herramienta”. En este capítulo se exponen las características funcionales de la herramienta mediante las Historias de Usuario (HU). Se define la propuesta de solución. Se estiman y planifican las HU para su implementación.
- ❖ Capítulo 3 “Diseño, Implementación y Pruebas de la Herramienta”. En este capítulo se define la arquitectura de la herramienta y los patrones de diseños empleados. Se redactan las Tarjetas CRC (Clase Responsabilidad Colaborador) y las Tareas de Ingeniería que responden a cada Historias de Usuario (HU). Se realizan las pruebas para verificar que la herramienta cumpla con las necesidades del cliente.

Capítulo 1 Fundamentación Teórica

1.1. Introducción

En el presente capítulo se abordan los principales conceptos referentes a la presente investigación. Además, se caracterizan la metodología, las herramientas y tecnologías que se emplean para el desarrollo de la solución informática.

1.2. Conceptos asociados a la Investigación:

Auditorías Informática

La auditoría informática es un proceso llevado a cabo por profesionales especialmente capacitados para el efecto, y que consiste en recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos. Permiten analizar de forma sistemática el uso de los recursos y los flujos de información dentro de una organización y determinar qué información es crítica para el cumplimiento de su misión y objetivos, identificando necesidades, duplicidades, costes, valor y barreras, que obstaculizan los flujos de información eficientes. Auditar consiste principalmente en estudiar los mecanismos de control que están implantados en una empresa u organización, determinando si los mismos son adecuados y cumplen determinados objetivos o estrategias, estableciendo los cambios que se deberían realizar para la consecución de los mismos. (3)

Objetivos de la auditoría Informática

- ❖ El análisis de la eficiencia de los Sistemas Informáticos.
- ❖ La verificación del cumplimiento del Reglamento sobre la seguridad y protección de la información oficial y el modo en que se aplicarán las normas de seguridad establecidas, Resolución 1, de 26 de diciembre de 2000 en el Decreto Ley 199 del Ministerio de Interior.
- ❖ La revisión de la eficaz gestión de los recursos informáticos.
- ❖ Control de la función informática.

Tipos de auditorías de Sistemas

Dentro de la auditoría informática destacan los siguientes tipos: (3)

- ❖ Auditoría de la gestión: la contratación de bienes y servicios, documentación de los programas.
- ❖ Auditoría de los datos: Clasificación de los datos, estudio de las aplicaciones y análisis de diagramas de flujo.
- ❖ Auditoría de las Bases de Datos: Controles de acceso, de actualización, de integridad y calidad de los datos.
- ❖ Auditoría de la seguridad: Referidos a datos e información verificando disponibilidad, integridad, confidencialidad, autenticación y no repudio.
- ❖ Auditoría de la seguridad física: Referido a la ubicación de la organización, evitando ubicaciones de riesgo, y en algunos casos no revelando la situación física de esta. También está referida a las protecciones externas y protecciones del entorno.
- ❖ Auditoría de la seguridad lógica: Comprende los métodos de autenticación de los sistemas de información.
- ❖ Auditoría de las comunicaciones: Se refiere a la auditoría de los procesos de autenticación en los sistemas de comunicación.
- ❖ Auditoría de la seguridad en producción: Frente a errores, accidentes y fraudes.

El tipo de auditoría que realizará la herramienta es la auditoría a Bases de Datos.

Auditoría a Bases de Datos

Es el proceso que permite medir, asegurar, demostrar, monitorear y registrar los accesos a la información almacenada en las bases de datos incluyendo la capacidad de determinar:

- ❖ Quién accede a los datos.
- ❖ Cuándo se accedió a los datos.
- ❖ Desde qué tipo de dispositivo/aplicación.
- ❖ Desde que ubicación en la Red.
- ❖ Cuál fue la sentencia SQL ejecutada.
- ❖ Cuál fue el efecto del acceso a la base de datos.

Objetivos Principales de las Auditorías a Bases de Datos:

- ❖ Impedir el acceso externo.
- ❖ Impedir el acceso interno a usuarios no autorizados.
- ❖ Autorizar el acceso sólo a los usuarios autorizados.
- ❖ Mantener trazas de uso y del acceso a bases de datos.
- ❖ Permitir investigaciones.

- ❖ General alertas en tiempo real.

1.3. Estudio de SASGBD:

Sistema para la Gestión de Auditorías a Sistemas Gestores de Bases de Datos (SASGBD):

SASGBD es un sistema para la realización de auditorías a Sistemas Gestores de BD utilizado por el personal del Departamento de Seguridad Informática de ETECSA. El sistema desarrollado por el centro TLM de la UCI, está destinado a automatizar el proceso de auditoría a Sistemas Gestores de Bases de Datos (SGBD) que se realiza en la empresa. El mismo cuenta con un Módulo de auditorías a BD y una herramienta Portable. El Portable es capaz de cargar las consultas y comandos a ejecutar en el servidor para auditar un gestor de BD, obtener los resultados y luego estructurarlos en un fichero XML para su posterior análisis por parte del Módulo de BD. El Módulo de BD es el responsable de las auditorías. Cuenta con una arquitectura en capas:

Arquitectura del Módulo de BD

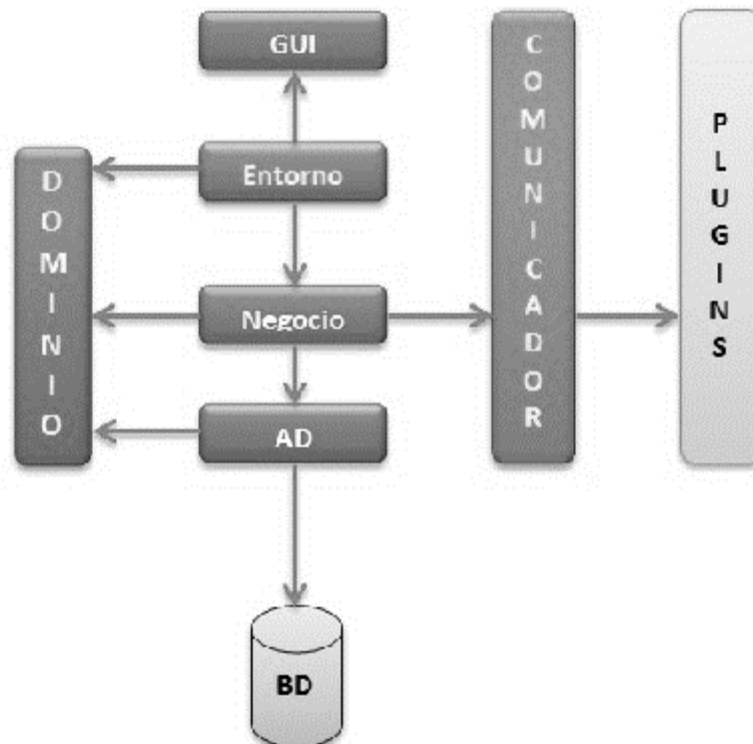


Ilustración 1 Arquitectura del Módulo de BD.

El Módulo soporta un sistema de plugins que lo hace flexible a la hora de agregarle soporte a nuevos SGBD. Al iniciar la aplicación el Servicio de Localización busca los plugins en la

carpeta “Plugins” de la estructura de carpetas de la herramienta, y verifica su compatibilidad con el sistema. En el momento de la instalación de un plugin nuevo el sistema almacena en la BD toda la configuración inicial de este. La información que se inserta consta de un gestor y sus versiones, una lista de indicadores y de consultas, los cuales el plugin trae consigo por defecto para la realización de auditorías a ese gestor. Los plugins además pueden ser desinstalados, pero esto no produce la eliminación de los datos del gestor para el cual se instaló. La gestión de las consultas es otra funcionalidad del sistema. Estas consultas pueden ser adicionadas a un gestor y para varias versiones pero es necesaria la existencia de un plugin instalado que valide la sintaxis de la sentencia SQL de la consulta a insertar. La búsqueda del plugin al insertar cada consulta se realiza en base al nombre del gestor y las versiones seleccionadas. Para la modificación al igual que en la adición es necesario de un plugin para la validación de la sentencia SQL. Las consultas se muestran en un árbol, el cual cuenta con 3 niveles (Gestores, Indicadores, Consultas), cada consulta será hija del Indicador al cual está asociada. El script de consultas para ejecutar la auditoría por el Portable, es exportado en un fichero XML como consecuencia de la selección del gestor, la versión y las consultas que serán utilizadas. El resultado devuelto por la auditoría es cargado en formato XML y evaluado por un plugin según el gestor y la versión que el fichero tiene en sus propiedades. El plugin arroja una matriz de resultados con los valores encontrados en la auditoría y una evaluación para cada uno de estos. El proceso de evaluación de la matriz es calculada mediante un método estadístico aprobado por el cliente, el cual tiene en cuenta la evaluación de cada resultado encontrado y el peso de la consulta asociada a esta. Según el proceso de realización de auditorías en el Departamento de Seguridad Informática de ETECSA, la evaluación de los resultados de una matriz de auditoría puede ser editada a consideración del supervisor. Los Informes Generales son los reportes finales de la realización de una auditoría, estos son creados a partir de una matriz de auditoría y son registrados en la BD, también es posible exportar en formato PDF los datos del informe.

1.4. Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el progreso de productos software. Las mismas definen quién está haciendo qué, cuándo y cómo alcanzar un objetivo específico, al proporcionar normas para el desarrollo eficiente del software con calidad. Actualmente existen dos grandes clasificaciones, las metodologías tradicionales y las ágiles. Estas últimas se basan

en la satisfacción del cliente, equipos de software pequeños con alta motivación y un mínimo de productos de trabajo de la Ingeniería de Software. (4)

Programación Extrema

La Programación Extrema (XP) es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código. Desarrollada por Kent Beck, XP surge como respuesta y posible solución a los problemas derivados del cambio en los requerimientos. Básicamente plantea que se trabaje directamente con el cliente haciendo pequeñas iteraciones y como resultado mini entregas cada tres semanas. Se utilizan las historias de usuario con el objetivo que cada una no puede demorarse más de una semana, permitiendo el trabajo en parejas, o sea, dos programadores por máquina, donde los programadores menos experimentados aprenden de los más experimentados. (5)

Tiene las siguientes fases:

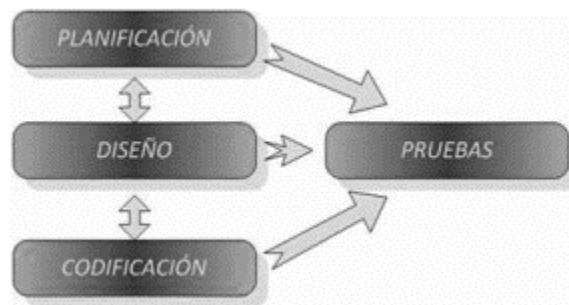


Ilustración 2. Fases de la Metodología XP (5)

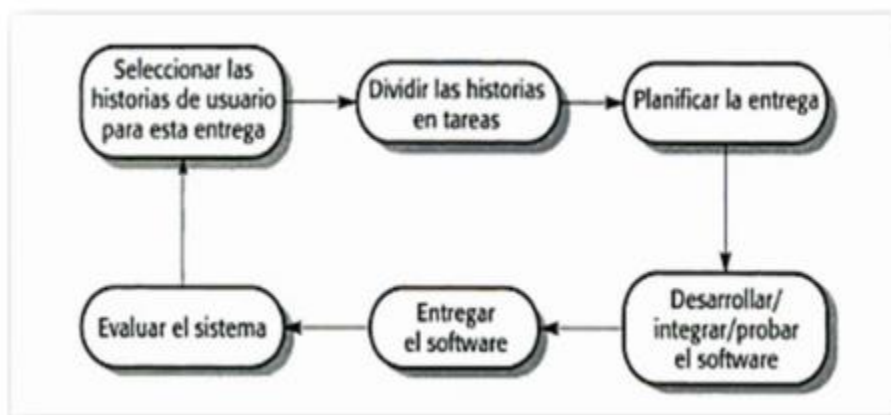


Ilustración 3. Ciclo de la Metodología XP. (5)

Para guiar el desarrollo de esta aplicación se emplea la metodología ágil XP por las características que posee de centrarse básicamente en la implementación de las soluciones y no en el soporte documental. Además la herramienta a implementar como parte de la investigación, debe ser desarrollada a corto plazo por un equipo pequeño de desarrollo. Logra que cada miembro del equipo, incluyendo el cliente esté listo para enfrentar cualquier cambio en el software, promoviendo el trabajo en equipo y el aprendizaje de los desarrolladores. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo y tiene como objetivo lograr la satisfacción del cliente.

1.5. Herramientas y Tecnologías informáticas:

Las Herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. En un sentido amplio del término, se puede decir que una herramienta es cualquier programa o instrucción que facilita una tarea. Es muy importante usar las herramientas adecuadas para cada tarea. En ese aspecto cada herramienta se crea y diseña para una o varias funciones determinadas, y por tanto se pueden hablar de muy diversos tipos de herramientas informáticas según el campo al que se dediquen. (6)

1.5.1. Lenguaje de programación del lado del servidor

Para el desarrollo del sistema se ha seleccionado Python que posee una sintaxis limpia y elegante que permite trabajar de una manera más cómoda y eficiente, a la vez ahorra una cantidad considerable de tiempo en el desarrollo de aplicaciones. Además es un lenguaje de encriptado independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

Este lenguaje se define por un grupo de características de las cuales se destacan las siguientes:

Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje, siempre y cuando exista un intérprete programado para él.

Interactivo: Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede

ayudar a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

Orientado a Objetos: La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

Funciones y librerías: Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas, números, archivos. Además, existen muchas librerías que se pueden importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .Zip. (7)

Se seleccionó Python por ser el lenguaje utilizado en el centro TLM en su versión 2.7.

1.5.2. Framework de Desarrollo

Django 1.4: Django como framework web implementado sobre el lenguaje de programación Python, bajo la licencia de Berkeley Software Distribution (BSD). Django proporciona estructura al código fuente, impulsando las buenas prácticas de programación web, lo que facilita al código legible y fácil de mantener. Acoge como particularidad la implementación del paradigma Model Template View (MTV), teniendo como objetivo brindar facilidad para la creación de sistemas web complejos (8). Proporciona abstracciones de alto nivel de patrones comunes del desarrollo web, atajos para tareas frecuentes de programación y claras convenciones sobre cómo resolver problemas. Su alta escalabilidad le posibilita manejar el crecimiento continuo de trabajo de manera fluida sin perder calidad en los servicios.

Jquery 1.9.1: Es un marco de trabajo JavaScript, facilita la creación de aplicaciones del lado del cliente. Además permite manipulación de documentos, gestión de eventos y animaciones. Está bajo la licencia MIT y la licencia GNU versión 2 la cual permite el uso en cualquier tipo de plataforma, personal o comercial. Además elimina la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. (9) Se utiliza para manejar todas las operaciones de las acciones del cliente.

Backbone 1.1.0: Backbone.js es un pequeño framework de desarrollo para el lenguaje de programación JavaScript, basada en el paradigma de diseño de aplicaciones Modelo Vista Controlador. Brinda una estructura a las aplicaciones web al ofrecer modelos con la unión clave-valor y eventos personalizados, contiene colecciones con una API de funciones

innumerables, puntos de vista con eventos de manipulación declarativa y lo conecta todo a su API (Interfaz de Programación de Aplicaciones) existente sobre una interfaz JSON REST (Transferencia de Estado Representacional). (10). Es utilizado por la herramienta en la construcción de tablas paginadas como otra variante de visualizar la información.

1.5.3. Lenguaje de programación del lado del cliente

Lenguaje de Marcas de Hipertexto 5 (HTML, Hyper Text Markup Language): Es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis, indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios. (11) Se utiliza para la construcción de las plantillas.

JavaScript: Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. (12) Se utiliza para cargar los datos, para la construcción de las tablas que se muestran por la herramienta y del árbol que contiene los parámetros necesarios para elaborar el script de revisión.

Hojas de Estilo en Cascada 3 (CSS3, Cascading Style Sheets): Es la versión más reciente de CSS el cual es un lenguaje de hojas de estilos, creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (13) Su empleo en el desarrollo de la herramienta web permitió colocar elementos en las páginas con mayor precisión.

1.5.4. Estructura de intercambio de datos entre cliente y servidor.

JSON (JavaScript Object Notation): En español notación de objetos JavaScript, es un formato de intercambio de datos estructurados entre todos los lenguajes de programación. Se basa en dos estructuras. Una de ellas es una colección de objetos, registros, estructuras, diccionarios, tablas hash, listas con clave, o matriz asociativa y la otra una lista ordenada de valores. En la mayoría de los lenguajes, esto se realiza como una matriz, vector, lista o secuencia. (14). Se utiliza para consumir los datos del servidor.

Ajax: Es un acrónimo JavaScript asíncrono + XML, permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya

que el intercambio de información con el servidor se produce en un segundo plano. (15) Se utiliza para realizar todas las peticiones de los datos del cliente hacia el servidor.

1.6. IDE (Entorno de Desarrollo Integrado)

Eclipse (Helios) v3.6.1:

Entorno de desarrollo multilenguaje, de código abierto y libre que se ejecuta sobre la máquina virtual de Java, con una comunidad muy activa de desarrolladores. Posee una arquitectura basada en plugins que permite añadir soporte para varios lenguajes, entre los cuales se encuentra Python el lenguaje de programación seleccionado para desarrollar el sistema. Además posibilita la integración de muchas librerías utilizadas por dicho lenguaje. Por todas estas características se propone la utilización de Eclipse.

1.7. Lenguaje Unificado de Modelado (UML, Unified Modeling Language)

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar todo el proceso de desarrollo de software. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (16)

1.8. Notación para el Modelado de Procesos de Negocio (BPMN, Business Process Modeling Notation)

Es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. El principal objetivo de BPMN es proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. Tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. (17) Se selecciona esta notación para modelar el proceso de negocio en el que se enmarca la presente investigación ya que provee una notación común para que los procesos puedan ser representados gráficamente de forma más clara, estandarizada y completa.

1.9. Herramienta CASE (Computer Aided Software Engineering)

Las Herramientas CASE son un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software. Entre las cuales se encuentra Visual Paradigm.

Visual Paradigm para UML

Es una herramienta que se utiliza para realizar el modelado de software. Se caracteriza por ser un sistema multiplataforma, distribuido bajo una licencia comercial y gratuita. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. (18) Debido a sus características se selecciona esta herramienta CASE en su versión 8.0 para el modelado del negocio y el diagrama de clases.

1.10. Gestor de Bases de Datos

PostgreSQL

Para la gestión de los datos de la aplicación se utiliza PostgreSQL en su versión 9.1, es un gestor de base de datos relacional orientado a objetos, libre y es utilizado por muchos programadores que realizan aplicaciones cliente servidor, complejas o críticas. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel. (19) Por estas características fue seleccionado como gestor de BD

1.11. Conclusiones parciales

En el presente capítulo fueron abordados los principales conceptos empleados en el desarrollo de la investigación. Se realizó un estudio del Módulo de BD de SASGBD así como de sus funcionalidades. Se estudiaron los conceptos, herramientas informáticas y metodología a utilizar para un mejor desarrollo de la herramienta. Se seleccionó XP como metodología para guiar el proceso de desarrollo, los lenguajes de programación Python 2.7 y JavaScript. Como framework de desarrollo Django 1.4 y JQuery1.9.1. Se escogió como herramienta CASE Visual Paradigm.

Capítulo 2: Características y Diseño de la Herramienta

2.1. Introducción

En el capítulo se describen las características de la herramienta y se detallan sus principales funcionalidades a desarrollar para cumplir con lo requerido por el cliente. Además se generan los artefactos correspondientes a la fase de Planificación de la metodología XP.

2.2. Procesos de negocio Gestión de Auditorías a Sistemas Gestores de Bases de Datos

Gestión de Auditorías es un conjunto de procesos por los cuales se controla el ciclo de vida de la información resultante de las auditorías realizadas a los SGBD de ETECSA, desde su obtención, hasta su disposición final en un informe resultado.

El proceso de negocio comienza una vez seleccionado el SGBD que se desea auditar, se procede por el supervisor a generar el script correspondiente para realizar la auditoría. Para conformar este script en formato XML (eXtensible Markup Language ('lenguaje de marcas extensible')) el supervisor debe seleccionar el Tipo gestor de BD, su versión, y una lista de consultas predefinidas que son las encargadas de probar cuan segura es la BD. El supervisor le entrega personalmente el script al administrador de la BD, que al ser ejecutado, se obtienen los resultados de la auditoría en otro script XML.

Este script es entregado por el administrador al supervisor para su análisis con una guía de revisión, lo cual permite valorar los parámetros del script resultante. Una vez importado dicho fichero se conforma y analiza la Matriz de Resultados. La Matriz contendrá una lista de valores encontrados los cuales son generados por las consultas contenidas en cada parámetro del script de revisión. Estos valores son evaluados mediante un cálculo usando los mismos métodos estadísticos, aprobados por el cliente, utilizados por el Módulo de BD de SASGBD, los cuales tienen en cuenta la evaluación de cada valor encontrado y el peso de la consulta asociada a esta. Según el proceso de realización de auditorías en el Departamento de Seguridad Informática de ETECSA la evaluación de los resultados de una matriz de auditoría puede ser editada a consideración del supervisor. Como resultado final se elabora un informe general que sintetiza los resultados de la auditoría.

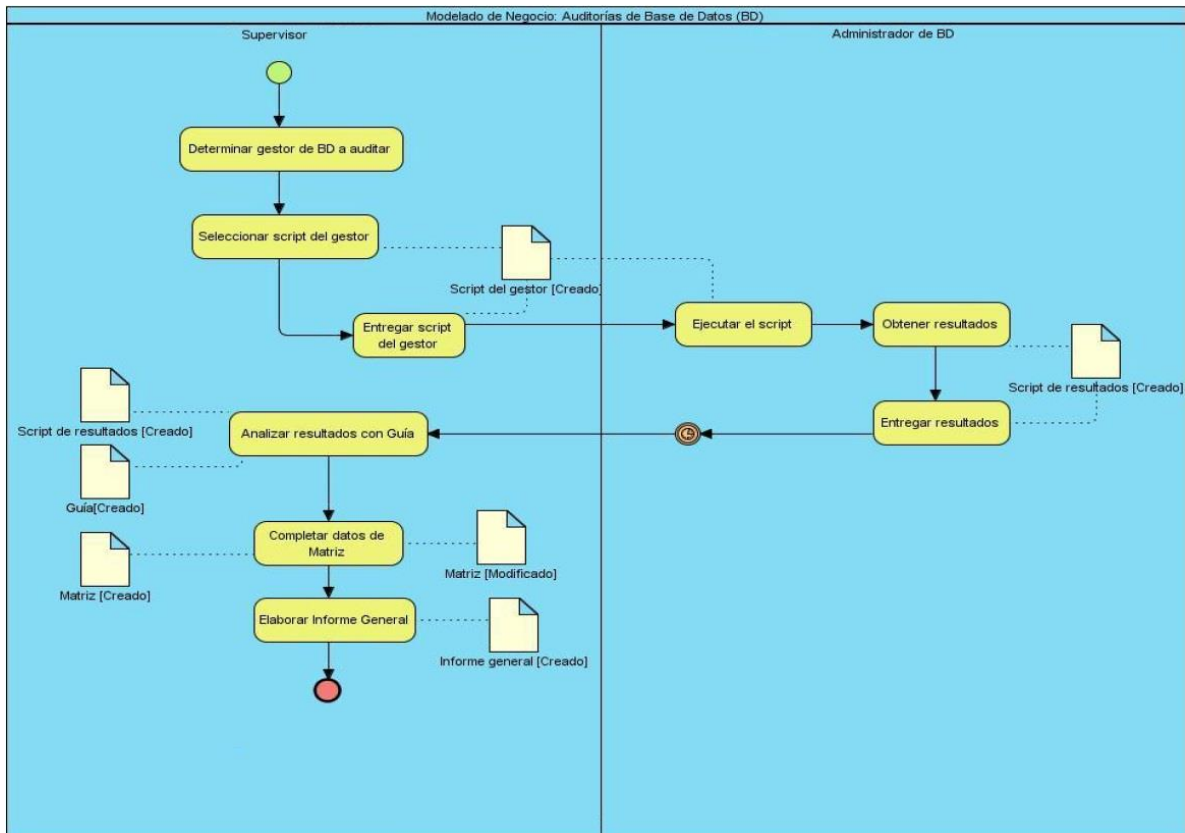


Ilustración 4. Modelo de Procesos de Negocio: Auditorías de Bases de Datos

2.3. Propuesta de Solución

La propuesta de solución está compuesta por una interfaz web desarrollada en Python usando el marco de desarrollo Django. La aplicación permitirá al supervisor crear un script de revisión en formato XML que contiene los datos a usar por la herramienta portable para realizar la auditoría, estos son:

- ❖ La versión del SO.
- ❖ La aplicación que utiliza el servidor.
- ❖ Un SGBD con su versión.
- ❖ Listado de indicadores los cuales tienen parámetros que verifican la configuración del servidor que se le va a realizar la auditoría.

La aplicación admitirá importar un fichero XML, resultado de la auditoría y así conformar la matriz de resultado. Este fichero resultado contiene:

- ❖ Dirección IP del servidor.
- ❖ Sistema Gestor de Base de Datos (SGBD).

- ❖ Versión del SGBD.
- ❖ Sistema Operativo (SO).
- ❖ Versión del SO.
- ❖ Indicadores con sus parámetros.
- ❖ Los valores resultados asociados a cada parámetro.

La matriz conformada, además de los datos anteriores del XML resultado, tendrá la evaluación del sistema para cada valor asociado a un parámetro, que puede ser o no reevaluado por el supervisor. Además contará con campos para redactar observaciones y recomendaciones.

Al concluir la matriz la aplicación brinda la posibilidad, de a partir de esta, conformar un informe resultado en formato PDF, que recoge todos los datos contenidos en la misma. La aplicación tendrá la opción de gestionar los usuarios que acceden a ella con sus diferentes roles.

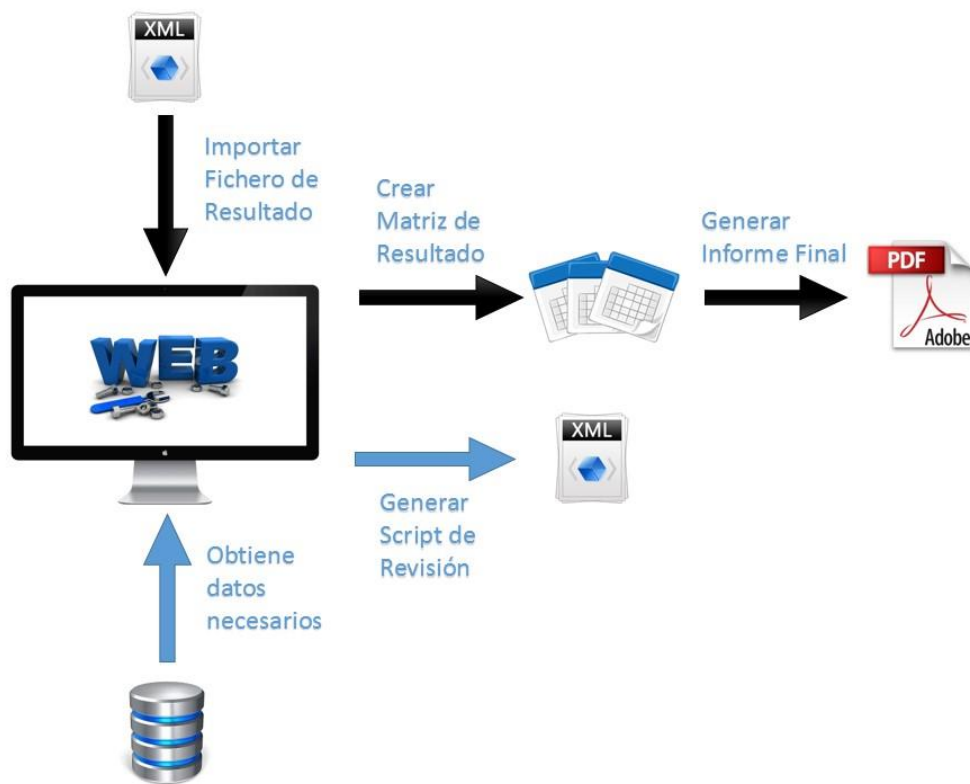


Ilustración 5. Propuesta de Solución.

2.4. Funcionalidades del GASGBD

Registrar	<ul style="list-style-type: none"> • Registrar entidad. • Registrar aplicación. • Registrar servidor. • Registrar indicador. • Registrar parámetro.
Auditoría	<ul style="list-style-type: none"> • Generar Script. • Importar Fichero de Resultado. • Concluir Matriz de Resultado. • Generar Informe.
Reporte	<ul style="list-style-type: none"> • Buscar Matriz de Resultado. • Buscar Informe.
Administración	<ul style="list-style-type: none"> • Cambiar contraseña. • Gestionar Usuario. • Gestionar Rol.

Tabla 1. Funcionalidades

2.5. Lista de Reserva de Producto

Para un correcto funcionamiento de la herramienta se deben tener en cuenta la lista de reserva de producto siguiente:

2.5.1. Usabilidad

Se requiere una preparación previa para operar con el sistema. Se exige un nivel medio o alto de conocimientos de informática, aunque el manejo de la aplicación es sencillo e intuitivo, lo que permite facilidad en el uso y comprensión del sistema por el usuario.

2.5.2. Hardware

Para la instalación de la aplicación se debe disponer de una computadora de 1 Gb de memoria RAM y 1 GB de espacio en disco duro.

2.5.3. Software

- PostgreSQL 9.1 como gestor de BD.
- Navegador Web.

2.5.4. Seguridad

El sistema deberá contar con una protección que incluya seguridad a nivel de usuarios, roles, contraseñas, privilegios; para garantizar la seguridad general de la información.

2.5.5. Interfaz de usuario

La aplicación propuesta conservará una interfaz sencilla dirigida a las personas que se relacionen con el sistema. Además se deben mostrar mensajes con información al usuario que guíen su interacción con la aplicación, estos mensajes pueden ser de información, alertas o errores.

2.6. Personas relacionadas

Se definen como personas relacionadas con la herramienta el supervisor y el auditor. Estos se encargaran de realizar la auditoría al SGBD.

2.7. Fase de Planificación

La primera fase de la metodología XP es la Planificación, la cual comienza con una serie de Historias de Usuario (HU), que se encargan de describir las características y funcionalidades del sistema que se quiere construir. Durante esta fase se estimará el esfuerzo de implementar cada HU. También se calcula el tiempo de cada iteración para el desarrollo del sistema.

2.7.1. Historias de Usuario

Para sustituir a los documentos de especificación funcional y a los casos de uso se utilizan las HU. Estas HU son escritas por el cliente en su propio lenguaje como descripciones cortas de lo que el sistema debe realizar. El tratamiento de las HU es muy dinámico y flexible, permite que en cualquier momento se puedan romper, reemplazar por otras más específicas o generales, añadirse nuevas o ser modificadas. Para ser implementadas las HU, el cliente y los desarrolladores se reúnen para detallar las funcionalidades de cada una. El tiempo de desarrollo ideal para una HU varía entre 1 y 3 semanas. (5)

Cada HU recoge los siguientes aspectos: (5)

Número: Número asignado a la HU.

Nombre de HU: Atributo que contiene el nombre de la HU.

Usuario: El usuario del sistema que utiliza o protagoniza la HU.

Prioridad en el negocio: Contiene el nivel de prioridad de la HU en el negocio. Es *Alta* en caso de que la HU sea indispensable en el negocio, *Media* en caso de que su realización o no afecte el negocio y *Baja* cuando no se considera una prioridad.

Riesgo de desarrollo: Contiene el nivel de riesgo en caso de no realizarse la HU. Es *Alta*, si el riesgo de no realizar la HU incide en el funcionamiento de la plataforma, *Media* si el riesgo de no realizarla es medianamente importante, y *Baja* en caso de que no se considere un riesgo tardar en la realización de la HU y no incida en el funcionamiento de la plataforma.

Puntos estimados: Este atributo es una estimación hecha por el equipo de desarrollo sobre el tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo, y un día equivale a 0.2 puntos.

Iteración asignada: Especifica la iteración a la que pertenece la HU correspondiente.

Descripción: Posee una breve descripción de lo que realizará la HU.

Historia de Usuario	
Número: 1	Usuario: Supervisor
Nombre de Historia de Usuario: Registrar	
Prioridad en negocio: Alta	Riesgo de Desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: El supervisor selecciona del menú "Registrar" la opción deseada. Esta puede ser "Entidad", "Servidor", "Aplicación", "Indicador" o "Parámetro".	
Observaciones: El supervisor tiene que estar previamente autenticado.	
Interfaz:	

Nombre	Dirección	Anfitrión
UCI	calle 4	Si
Copextel	calle 2	Si
TRD	calle 24	No

Tabla 2. Historia de Usuario # 1.

Las Historias de Usuario número 2 en adelante se encuentran en el [Anexo I](#).

2.7.2. Estimación de esfuerzo por Historia de Usuario

Por regla general que establece la Metodología XP en sus principios generales cada una de las Historias de Usuario necesita de una a tres semanas de desarrollo. Las estimaciones de esfuerzo se realizan en una escala de uno a tres puntos, siendo un punto la referencia a una semana excelente de programación.

Historia de Usuario	Punto de Estimación
1. Registrar	1
2. Generar script de revisión	1
3. Importar fichero de resultados	1
4. Concluir matriz	1
5. Generar Informe	1
6. Buscar Matriz	0.4

7. Buscar Informe	0.4
8. Cambiar Contraseña	0.2
9. Gestionar Usuario	0.2
10. Gestionar Rol	0.2

Tabla 3. Estimación de esfuerzo por HU.

2.7.3. Plan de Iteraciones

Para llevar a cabo la implantación de la herramienta se realizó una planificación de las HU anteriormente descritas, conformando 3 iteraciones para su ejecución. Con el objetivo de presentar funcionalidades al cliente al finalizar cada iteración, lo que permite que exista retroalimentación y este sea partícipe del proceso de desarrollo de software. Esto trae como resultado un producto de mayor calidad y más acorde a las especificaciones y necesidades del cliente.

2.7.3.1. Iteración 1:

En la iteración 1 se realizarán las HU del número 1 a la 3. Perteneciente a la implementación de las HU con prioridad alta, importantes en el desarrollo de la herramienta.

2.7.3.2. Iteración 2:

En la iteración 2 se realizarán las HU del número 4 a la 7. Perteneciente a la implementación de las HU con prioridad media donde se realiza todo lo referente a la elaboración del informe final.

2.7.3.3. Iteración 3:

En la iteración 3 se realizarán las HU del número 8 a la 10 Perteneciente a la implementación de las HU con prioridad baja, son las encargadas de la seguridad.

2.7.4. Plan de duración de las Iteraciones

Definidas las Historias de Usuarios y estimado el esfuerzo dedicado a la realización de cada una de ellas, se da paso a la planificación de la implementación de cada HU, estableciendo un total de 3 iteraciones.

Iteración	Historia de Usuario a Implementar	Duración Total
1	Registrar	3 semanas
	Generar script de revisión	
	Importar fichero de resultados	
2	Concluir matriz	2.8 semanas
	Generar Informe	
	Buscar Matriz	
	Buscar Informe	
3	Cambiar Contraseña	0.6 semanas
	Gestionar Usuario	
	Gestionar Rol	

Tabla 4. Plan de Iteración y entrega.

2.7.5. Plan de entrega

El plan de entrega detalla la culminación de cada iteración de la herramienta implementada.

Herramienta	Final de la iteración 1 (6/05/2015)	Final de la iteración 2 (26/05/2015)	Final de la iteración 1 (29/05/2015)
Herramienta de Gestión de Auditorías a Sistemas Gestores de Bases de Datos	GASGBD v0.1	GASGBD v0.2	GASGBD v1.0

Tabla 5. Plan de entrega de la Herramienta

2.8. Conclusiones parciales

Durante el desarrollo de este capítulo se contribuyó a lograr una mejor comprensión de la herramienta que se desea desarrollar. Se definieron y redactaron las historias de usuario,

artefacto que propone la metodología XP en su Fase de Planificación. Además se realizó una estimación y planificación del esfuerzo para la implementación de la herramienta y un plan de duración de las iteraciones.

Capítulo 3: Diseño, Implementación y Pruebas de la Herramienta

3.1. Introducción

En el presente capítulo se describen las fases de diseño y codificación de la metodología XP, correspondientes al diseño e implementación. Se presentan la arquitectura, se definen los patrones utilizados, además se representan las tarjetas Clase – Responsabilidad – Colaborador (CRC). También se especifican las Tareas de Ingeniería para describir en detalles cada una de las HU. Se muestran los estándares de codificación y la incidencia en las BD.

3.2. Patrón de arquitectura:

Modelo-Vista-Plantilla (Model Template View (MTV)) el cual es una modificación a la ya conocida Modelo Vista Controlador (MVC). El MTV no sigue ningún patrón de desarrollo en específico, se realizó de esa manera para hacer el framework Django, en el que está basado, más funcional.

El patrón MTV se ve frecuentemente en aplicaciones web, que hagan uso del framework Django, ya que fue adaptado en específico para este. El objetivo de definir patrones como este es principalmente hacer más fluida la comunicación entre desarrolladores, ya que permite organizar el trabajo de los mismos, realizando una separación estricta entre las partes de una aplicación, permitiendo realizar cambios en una parte en particular sin afectar el resto de la aplicación. (20)

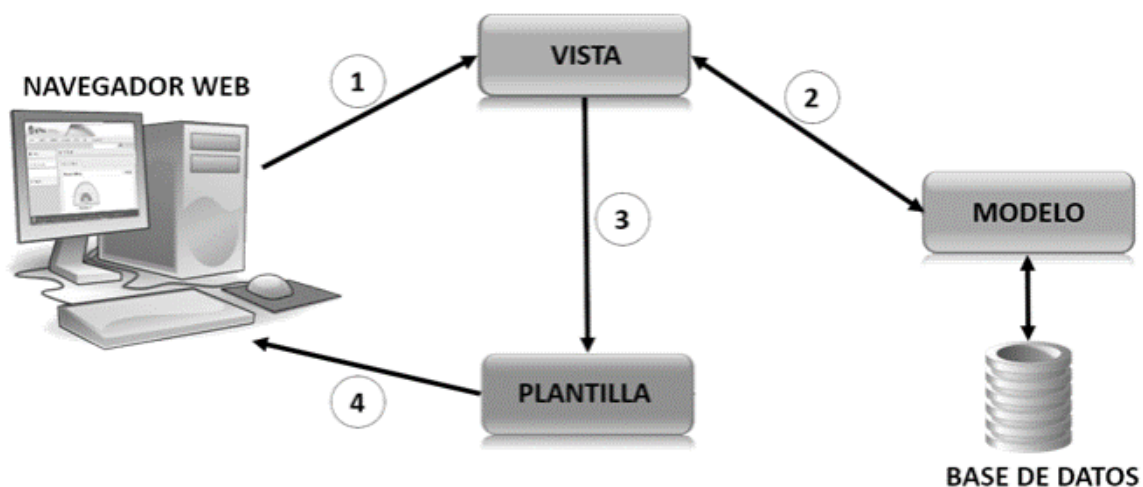


Ilustración 6. Arquitectura MTV.

1. El navegador manda una solicitud.
2. La vista interactúa con el modelo para obtener datos.
3. La vista llama a la plantilla.
4. La plantilla muestra la respuesta a la solicitud del navegador.

Separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- La Modelo se refiere al modelo, la capa de acceso de datos. Esta capa contiene todo lo referido a los datos: cómo acceder a ellos, cómo validarlos, qué comportamiento tienen y las relaciones entre ellos.
- La Plantilla se refiere a la capa de presentación. Esta capa contiene las decisiones relacionadas con la presentación cómo debería mostrarse algo en una página Web u otro tipo de documento.
- La Vista se refiere a la capa de la lógica de negocio. Esta capa contiene el acceso al modelo y delega en la(s) plantilla(s) apropiada(s). Puede pensar en ella como el puente entre los modelos y las plantillas.

Ventajas del uso de la arquitectura MTV:

- La posibilidad de tener diferentes vistas para un mismo modelo.
- La construcción de nuevas vistas sin necesidad de modificar el modelo subyacente.
- Agregar o quitar una funcionalidad no afecta a la aplicación.

3.3. Capas de la Arquitectura MTV

Se presentan los diagramas de paquetes que generan las capas de la arquitectura MTV, para un mejor entendimiento del marco de trabajo Django.

Capa Modelo:

Estas clases se encuentran dentro de la capa modelo que brinda la arquitectura MTV, encargadas de las consultas a la base de datos para la herramienta implementada.

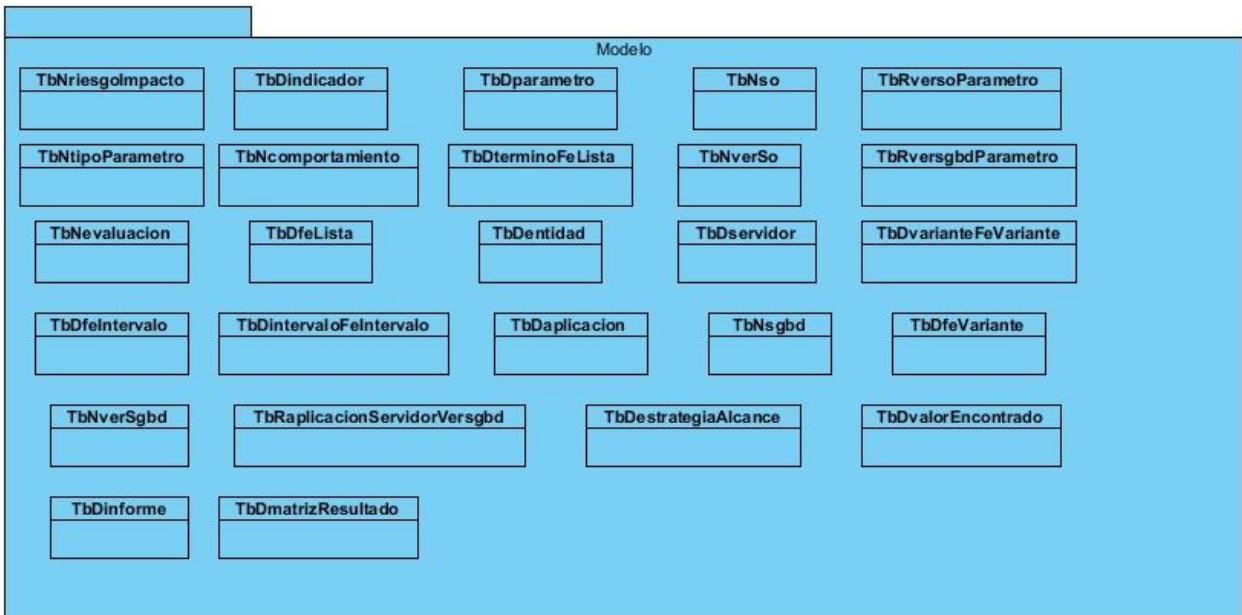


Ilustración 7. Capa Modelo.

Capa Vista:

Estas clases se encuentran dentro de la capa vista que brinda la arquitectura MTV, encargadas de generar el contenido de la plantilla a través de la interacción con la capa modelo para la herramienta implementada.

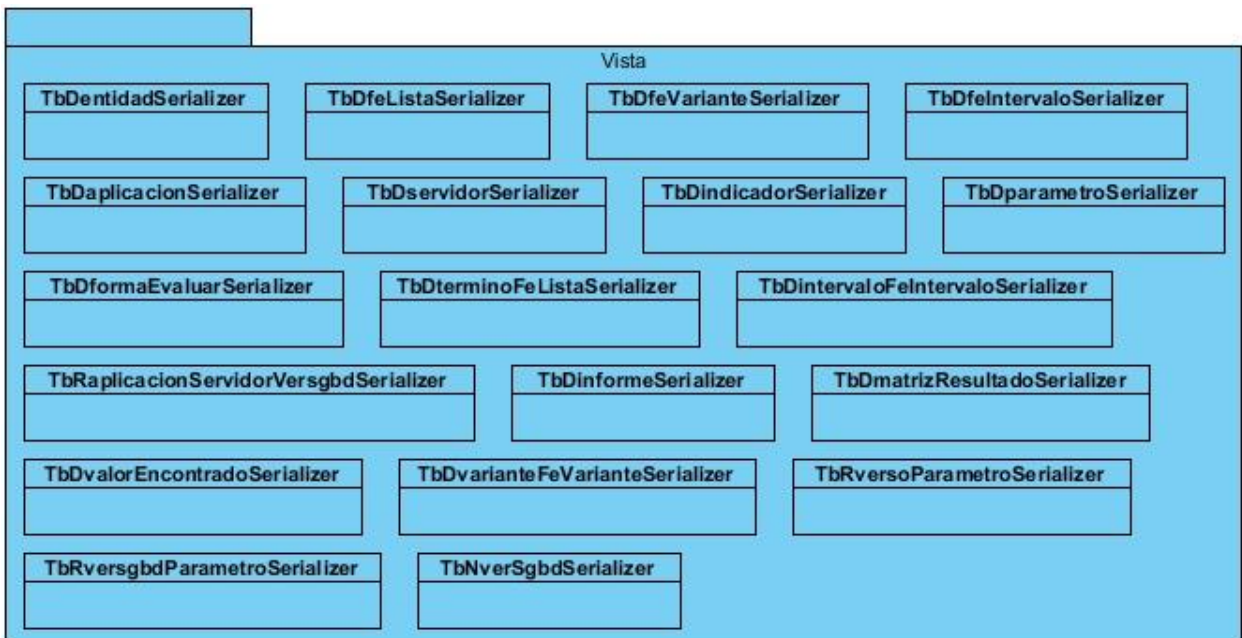


Ilustración 8. Capa Vista.

Capa Plantilla:

En la capa plantilla se encuentran las plantillas HTML heredando todas de jqgridGeneric.

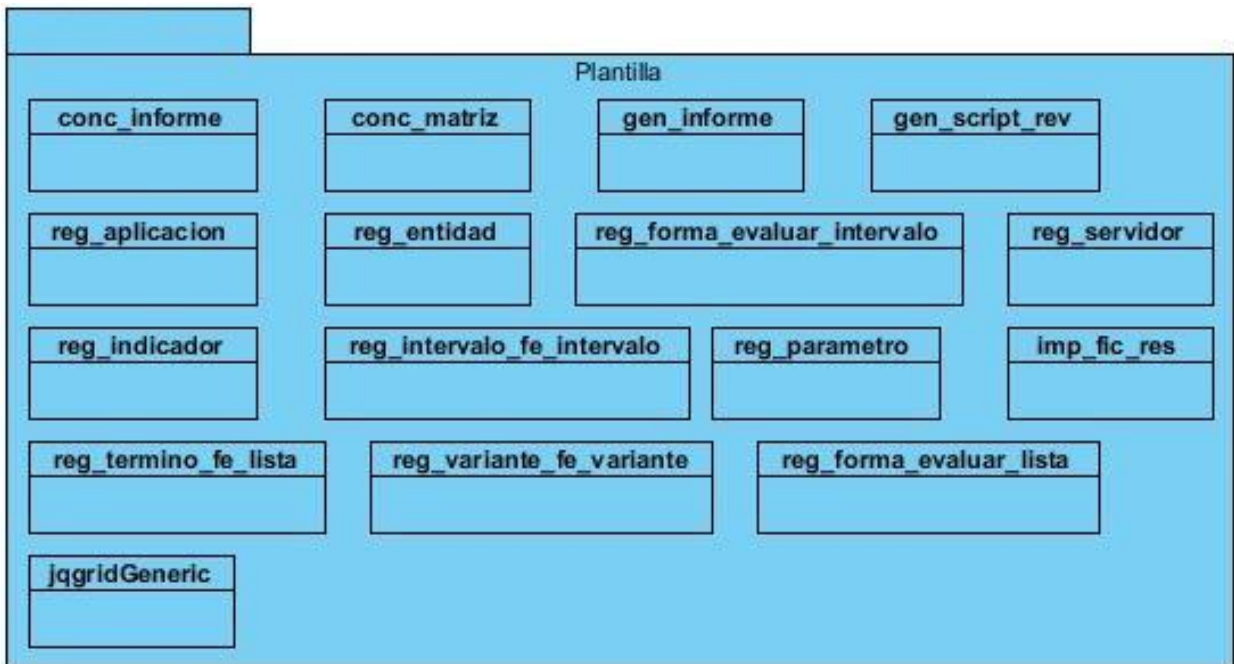


Ilustración 9. Capa Plantilla.

Relación entre las Capas de la Arquitectura:

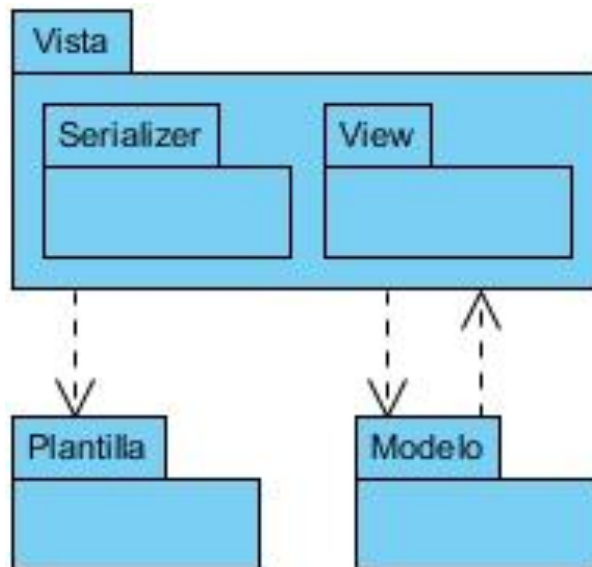


Ilustración 10. Diagrama de Paquetes.

3.4. Patrones de Diseño

Un patrón de diseño define un esquema de refinamiento de los subsistemas o componentes dentro de un sistema, o las relaciones entre estos. Este describe una estructura común y recurrente de componentes interrelacionados, que resuelve un problema general de diseño dentro de un contexto particular. (21)

3.4.1. GRASP

Acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. (22)

Experto:

Es un patrón encargado de asignar responsabilidades, es un principio básico que utiliza un diseño orientado a objetos. (22) Esto se puede evidenciar en la Capa Modelo donde las clases *TbDindicador*, *TbDparametro*, *TbDentidad*, *TbNso*, *TbNverSo*, *TbDservidor*, *TbDaplicacion*, *TbNsgbd*, *TbNverSgbd*, *TbDinforme*, *TbDmatrizResultado* y *TbDvalorEncontrado*, contienen la información necesaria para cumplir con sus responsabilidades.

Creador:

Guía la asignación de responsabilidades relacionadas con la creación de objetos. Es una tarea muy frecuente en los sistemas orientados a objetos. (22) Se evidencia al generar un script de revisión, se crea una instancia de la clase *Document* que manipula la estructura XML.

```
56 def genScriptRev(request):
57     data= request.POST.getlist('data[]')
58     id=request.POST['g']
59     sgbd=TbNsgbd.objects.get(id=id)
60     id=request.POST['v']
61     version=TbNverSgbd.objects.get(id=id)
62
63     xml = Document()
64     scriptRevision = xml.createElement("scriptRevision")
65     scriptRevision.setAttribute("idverSGBD", str(version.id))
66     scriptRevision.setAttribute("SGBD", str(sgbd.nombre))
67     scriptRevision.setAttribute("verSGBD", str(version.numero_version))
```

Ilustración 11. Ejemplo del Patron Creador.

Bajo Acoplamiento:

El Bajo Acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. (22)

3.5. Tarjetas CRC (Clase-Responsabilidad-Colaborador)

Forman parte de la fase de Diseño de la metodología XP y representan de manera organizada las diferentes clases en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

Clase genScriptRev	
Responsabilidades	Colaboradoras
<ul style="list-style-type: none"> Exportar script de revisión en formato XML. 	<ul style="list-style-type: none"> TbDindicadorSerializer TbDparametroSerializer TbNsgbdSerializer TbNverSgbdSerializer TbRversoParametroSerializer

Tabla 6. Tarjeta CRC: Clase genScriptRev.

Clase impFicheResult	
Responsabilidades	Colaboradoras
<ul style="list-style-type: none"> Importa el fichero XML resultado de la auditoría. Crear la matriz de resultados. 	<ul style="list-style-type: none"> ficheroResult matrizTableData TbDentidadSerializer

Tabla 7. Tarjeta CRC: Clase impFicheResult.

Clase regParametro	
Responsabilidades	Colaboradoras
<ul style="list-style-type: none"> Registra un parámetro en la BD. 	<ul style="list-style-type: none"> TbDindicadorSerializer TbNtipoParametroSerializer TbNriesgoImpactoSerializer

Tabla 8. Tarjeta CRC: Clase regParametro.

Clase regAplicacion	
Responsabilidades	Colaboradoras

<ul style="list-style-type: none"> • Registra una aplicación en la BD. 	<ul style="list-style-type: none"> • TbDentidadSerializer
---	--

Tabla 9. Tarjeta CRC: Clase regAplicacion.

3.6. Diseño de Base de Datos

La ilustración 11 contiene el modelo de la BD. La cual se encarga de almacenar toda la información utilizada por la herramienta.

Por una parte las tablas tb_DParametro, tb_DIndicador, tb_nSGBD y tb_nSO son las encargadas de guardar la información con la que se construye el script de revisión utilizado en la realización de la auditoría.

Las tablas tb_dValor_Encontrado, y tb_dMatriz_Resultado se encargan de guardar los datos referentes al fichero de resultados importado.

La tabla tb_dInforme su objetivo es guardar los informes finales, que son el resultado de todo el proceso de auditoría.

3.7. Tareas de Ingeniería

Según la metodología XP al terminar la implementación de cada HU se realizan pruebas que se le muestran al cliente para demostrar si cumplen con los requerimientos hechos para la misma. Este proceso rectifica las actividades definidas que realizan. Estas tareas recogen los diferentes aspectos:

Número de la tarea: Los números deben ser sucesivos.

Número Historia de Usuario: Número de la HU a la que pertenece la tarea.

Nombre Tarea: Nombre que identifica a la tarea.

Tipo de Tarea: Las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Definir).

Puntos Estimados: Tiempo estimado en días que se le asignará a su desarrollo.

Fecha Inicio: Fecha en que inicia el desarrollo de la tarea.

Fecha Fin: Fecha en que finaliza el desarrollo de la tarea.

Programador Responsable: Nombre y apellidos del programador.

Descripción: Breve descripción de la tarea.

A continuación se muestra una tabla con las tareas de ingeniería definidas y las HU correspondiente a cada una:

HU# 1 Registrar Aplicación	
Tareas de Ingeniería	
1	Registrar Aplicación.
2	Registrar Servidor.
3	Registrar Indicador.
4	Registrar Entidad.
5	Registrar Parámetro.

Tabla 10. Tareas de Ingeniería HU# 1.

Tareas de Ingeniería	
Número de la tarea: 1	Número HU: 1
Nombre Tarea: Registrar Aplicación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: <ul style="list-style-type: none"> • Se seleccionan los propietarios existentes en la base de datos. • Se crea y muestra la interfaz gráfica para la inserción de la aplicación. • Se verifican los datos a insertar. • Se inserta la aplicación en la base de datos y mostrar mensaje de inserción correcta. 	

Tabla 11. Tarea 1 de la HU# 1.

Las tareas de ingeniería correspondientes de la HU #2 a la 14 se encuentran en el [Anexo II](#).

3.8. Estándar de Codificación

Un estándar de codificación se utiliza para lograr un código claro y legible para los desarrolladores, y trabajar de forma coordinada dentro del proyecto, de acuerdo con el tipo de lenguaje que se utiliza y las normas que propone el mismo. (23)

3.8.1. Python

Propuestas de Mejora de Python (PEP0008 “Python Enhancement Proposals”) es una guía de estilos que facilita la lectura del código y la consistencia entre programas de distintos usuarios. (23)

Sangría:

Utilizar siempre 4 espacios y nunca mezclar tabuladores y espacios, ejemplo:

```
def sgbd(request):
    s = TbNsgbd.objects.all()
    data = serializers.serialize("json", s)
    return HttpResponse(data, content_type='application/json')
```

Ilustración 13. Ejemplo de Sangría (Python)

Tamaño máximo de línea:

Las líneas deben limitarse a un máximo de 79 caracteres.

Importaciones:

Las importaciones de distintos módulos deben estar en líneas diferentes y siempre al principio del fichero, ejemplo:

```
import json
import time
import xml.etree.ElementTree as ET

from django.http import HttpResponse
from django.core import serializers
from xml.dom.minidom import Document
from django.http import HttpResponse
```

Ilustración 14. Ejemplo de Importaciones (Python)

Espacios en blanco en expresiones:

Evitar los espacios en blanco extra dentro del paréntesis, después de una coma, antes del paréntesis de una llamada a una función, antes del paréntesis de un índice.

3.9. Pruebas

La metodología XP propone la fase de Prueba durante el desarrollo del software, estableciendo pruebas tantas como sean posible con el objetivo de reducir el número de errores no detectados durante la implementación, disminuir el tiempo entre la introducción de estos en el sistema y su detección; aumentar la seguridad y evitar efectos colaterales no deseados a la hora de realizar modificaciones en la aplicación. La metodología XP divide las pruebas del sistema en dos grupos:

3.10. Pruebas Unitarias

Son pruebas de caja blanca, realizadas constantemente por los programadores al concluir una parte de la implementación, el objetivo principal es verificar que las funcionalidades exigidas por el cliente se implementaron correctamente. Las pruebas unitarias se realizaron al concluir la implementación de cada funcionalidad utilizando la librería de Python (unit testing).

3.10.1. Iteración 1

Durante la iteración 1 se seleccionaron 11 fragmentos de código en los cuales no se detectaron errores.

```
-----  
Ran 11 tests in 0.310s  
  
OK  
Destroying test database for alias 'default'...
```

Ilustración 15. Pruebas Unitarias Iteración 1

3.10.2. Iteración 2

Durante la iteración 2 se seleccionaron 12 fragmentos de código en los cuales no se detectaron errores.

```
-----  
Ran 12 tests in 0.380s  
  
OK  
Destroying test database for alias 'default'...
```

Ilustración 16. Pruebas Unitarias Iteración 2

3.10.3. Iteración 3

Durante la iteración 3 se seleccionaron 1 fragmentos de código en los cuales no se detectaron errores.

```
-----  
Ran 1 test in 0.024s  
  
OK  
Destroying test database for alias 'default'...
```

Ilustración 17. Pruebas Unitarias Iteración 3

3.11. Pruebas de aceptación

Son pruebas de caja negra, especificadas por el cliente y se centran en las características y funcionalidades generales del sistema. Se derivan de las HU que han implementado como parte de la liberación del software.

- ❖ **Clases Válidas:** Se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas validas que hacen el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ❖ **Clases Inválidas:** Se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las posibles entradas inválidas q hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ❖ **Resultado Esperado:** Se hará una breve descripción del resultado que se espera, tanto para entradas validas como para entradas inválidas.
- ❖ **Resultado de la Prueba:** Se hará una breve descripción del resultado que se obtiene.
- ❖ **Observaciones:** Se hará la descripción de cualquier señalamiento o advertencia que sea necesario hacerle a la sección que se está probando.

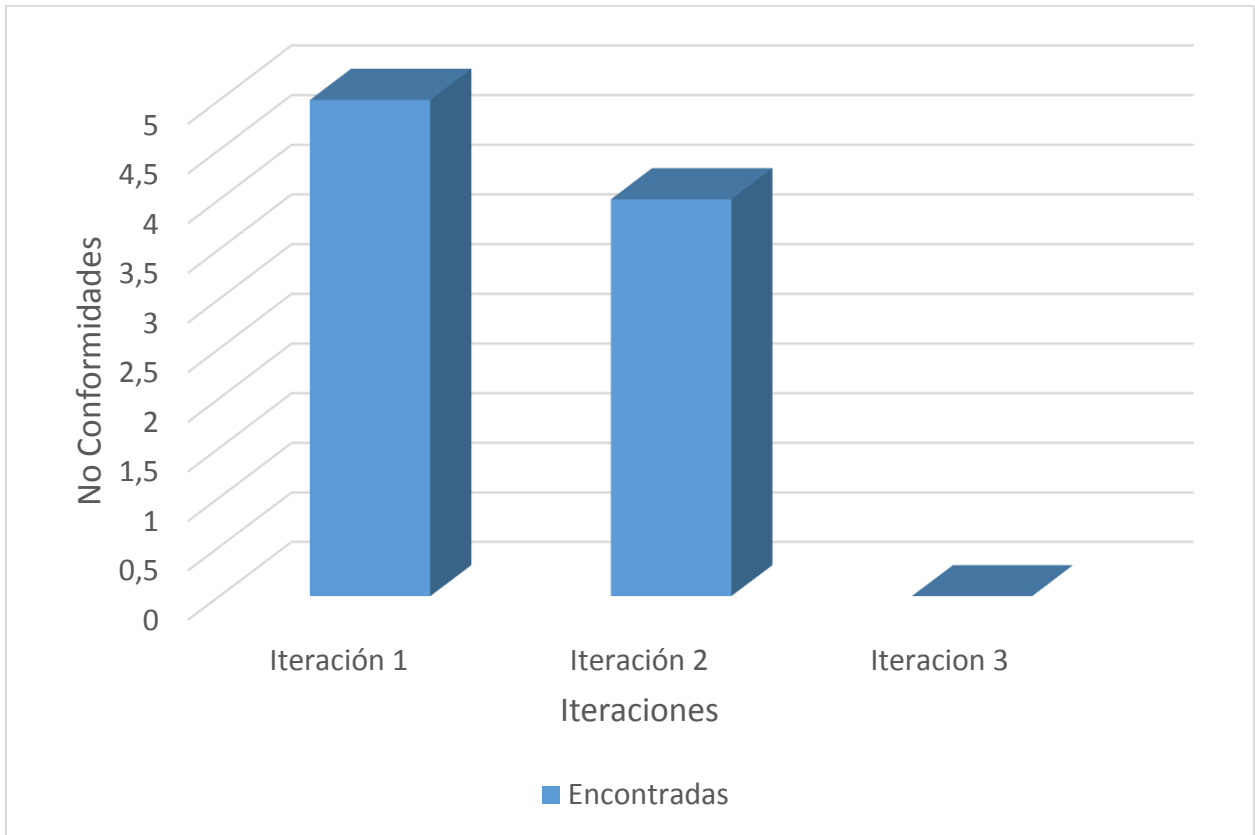


Ilustración 18: Pruebas de Aceptación

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
----------------	------------------	--------------------	------------------------	-------------

<p>El supervisor accede a la interfaz de la herramienta y selecciona del menú "Registrar" la opción "Aplicación y completa los campos que se muestran en la interfaz.</p>		<p>El sistema muestra el mensaje "La operación se ha realizado con éxito"</p>	<p>Satisfactorio</p>	
	<p>El supervisor accede a la interfaz de la herramienta y selecciona del menú "Registrar" la opción "Aplicación. Pero deja campos sin completar.</p>	<p>El sistema muestra un mensaje de error, "Este campo es obligatorio"</p>	<p>No se registra en la BD la aplicación. .</p>	

Tabla 12. Prueba de Aceptación #1: HU: Registrar Aplicación.

Las Pruebas de aceptación correspondientes a la HU 2 a la 6 se encuentran en el [Anexo III](#)

3.12. Conclusiones Parciales

Durante este capítulo se definieron los patrones diseño y crearon las tarjetas CRC. Se obtuvo un diseño de todas las clases que intervienen en la solución, permitiendo realizar la implementación del módulo. Se realizaron las pruebas unitarias empleando la librería de Python unit testing y las pruebas de aceptación donde todas las no conformidades fueron solucionadas.

Conclusiones Generales

Finalizada la investigación y desarrollada la versión web del Módulo de BD de SASGBD para contribuir con la estandarización tecnológica en el que está inmerso el centro TLM, se le da solución al problema planteado y cumplimiento al objetivo propuesto por el presente trabajo, el cual se ha cumplido satisfactoriamente, poniendo en práctica todas y cada una de las tareas propuestas para el desarrollo de esta versión. Por lo anteriormente expuesto se arriba a las siguientes conclusiones:

- ❖ Se realizó un estudio del Módulo de BD de SASGBD.
- ❖ Se desarrolló una versión web con las funcionalidades del Módulo de BD de SASGBD.
- ❖ Fueron realizadas pruebas unitarias y de aceptación para comprobar el correcto funcionamiento de la herramienta.
- ❖ La herramienta cumple con la estandarización tecnológica del centro TLM.

Luego del análisis anterior se puede afirmar que la presente investigación alcanzó el objetivo propuesto.

Recomendaciones

Se recomienda integrar la herramienta web a la plataforma de seguridad informática Xilema-PlatSi.

Referencias

1. Antunez Sanchez, Acides Francisco y Odoardo Hernandez, Noria. *Auditoría y seguridad informática. Realidades y Perspectivas en Cuba, una mirada del escenario actual*. 2010.
2. Empresa Empresa de Telecomunicaciones de Cuba S.A. *Empresa de Telecomunicaciones de Cuba S.A.* [En línea] 10 de 9 de 2012. [Citado el: 16 de 4 de 2015.] <http://www.etecsa.cu>.
3. Ramírez Rodríguez, German. *Sobre la auditoría informática y lopd de experiencia personal*. madrid : s.n., 2009.
4. Pressman, Roger. *Ingeniería de Software "Un Enfoque Práctico"*. s.l. : Mc Graw Hill, 2005.
5. Beck, Kent. *Extreme programming explained: embrace change* . Estados Unidos : Addison-Wesley, 2000. 201-61641.
6. Visuse. [En línea] 16 de 6 de 2010. [Citado el: 20 de 5 de 2015.] <https://visuse.wordpress.com/herramientas-y-tecnologias-utilizadas/>[En línea].
7. Lutz, Mark. *Learning Python, Fifth Edition*. 2013.
8. Kaplan-Moss, Jacob y Holovaty, Adrian. *El libro de Django*. s.l. : Jeremy Dunck, 2008.
9. The jQuery Foudation. jQuery UI . [En línea] 2014 . [Citado el: 22 de 5 de 2015.] <http://jqueryui.com>.
10. Backbonejs Projects. Backbonejs. [En línea] [Citado el: 22 de 5 de 2015.] <http://backbone.org>.
11. Kenny, Bill y Musciano, Chuck. *HTML. Guía Completa*. Mexico : McGraw-Hill , 2008.
12. Pérez, Javier Eguíluz. *Introducción a JavaScript*. . España : s.n., 2009.
13. Pérez, Javier Eguiluz. *Introducción a CSS*. [En línea] [Citado el: 25 de 5 de 2015.] <https://librosweb.es/libro/css/>.
14. ECMAScript. JSON. [En línea] 2014. [Citado el: 29 de 5 de 2015.] <http://json.org/>.
15. Pérez, Javier Equíluz. *Introducción a AJAX*. . España : s.n., 2008.
16. Jacobson, Ivar, Rumbaugh, James y Booch, Grady. *El lenguaje unificado de modelado*. s.l. : Addison Wesley, 2010.
17. Hitpass, Bernhard. *BPMN 2.0 Manual de Referencia y Guía Práctica*. Santiago de Chile : s.n., 2014.
18. Company Headquarters. visual-paradigm. [En línea] 2004. [Citado el: 29 de 2 de 2015.] <http://www.visual-paradigm.com>.
19. Martinez, Rafael. PostgreSQL. [En línea] 2009. [Citado el: 2 de 6 de 2015.] http://www.postgresql.org.es/sobre_postgresql..

20. Holovaty, Adrian y Kaplan-Moss, Jacob. *The Definitive Guide to django*. s.l. : Apress, 2010.
21. Oktaba, Hanna. Facultad de Ciencias, UNAM. [En línea] [Citado el: 2 de 6 de 2015.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
22. Larman, Craig. *UML y Patrones*. Canada : PEARSON, 2012.
23. van Rossum, Guido, Warsaw, Barry y Coghlan, Nick. Python Software Foundation. *PEP 0008 -- Style Guide for Python Code*. [En línea] 5 de 7 de 2001. [Citado el: 11 de 6 de 2015.] <https://www.python.org/dev/peps/pep-0008/>.

Bibliografía

- Downie, Nick. Chart.js | Open source HTML5 Charts for your website. [En línea] [Citado el: 20 de mayo de 2015.] <http://www.chartjs.org/>.
- Emilio A. Sánchez, Patricio Letelier, José H. Canós. Mejorando la gestión de historias de usuario en XP. Valencia España: Departamento de Sistemas Informáticos y Computación, 2014.
- geekytheory. [En línea] 13 de 10 de 2013 [Citado el: 22 de mayo de 2015]. <https://geekytheory.com/json-i-que-es-y-para-que-sirve-json/>.
- hipertextual.com. [En línea] 2014. [Citado el: 24 de mayo de 2015]. <http://hipertextual.com/archivo/pycharm-ide-python>.
- juan-garcia-carmona.blogspot.com. [En línea] 07 de 09 de 2012. [Citado el: 24 de mayo de 2015]. <http://juan-garcia-carmona.blogspot.com/2012/09/grasp-alta-cohesion-y-bajo-acoplamiento.html>.
- kioskea.net. [En línea] [Citado el: 25 de mayo de 2015] <http://es.kioskea.net/contents/224-patrones-de-diseno>.
- León, Eduardo. Tutorial Visual Paradigm for UML. [En línea] [Citado el: 25 de mayo de 2015]. <http://es.scribd.com/doc/36636137/Tutorial-Visual-Paradigm>.
- Metodologías Ágiles de desarrollo de software (XP) Fases. [En línea] 28 de junio de 2008. [Citado el: 27 de mayo de 2015]. http://boards5.melodysoft.com/UBV_INGS/metodologias-agiles-de-desarrollo-43.html.
- Penadés, Patricio Letelier y Ma. Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia: s.n., 2010.
- Prácticas de software. [En línea] 21 de 03 de 2011. [Citado el: 27 de mayo de 2015]. <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
- Python Software Foundation. Welcome to Python.org. [En línea] 2001-2014. [Citado el: 28 de mayo de 2015.] <https://www.python.org/doc/essays/blurb/>.
- Sommerville, Ian. Ingeniería del Software Séptima Edición. Madrid, España: Pearson Educación, S.A, 2005.

Anexos | Muestra de Historias de Usuario

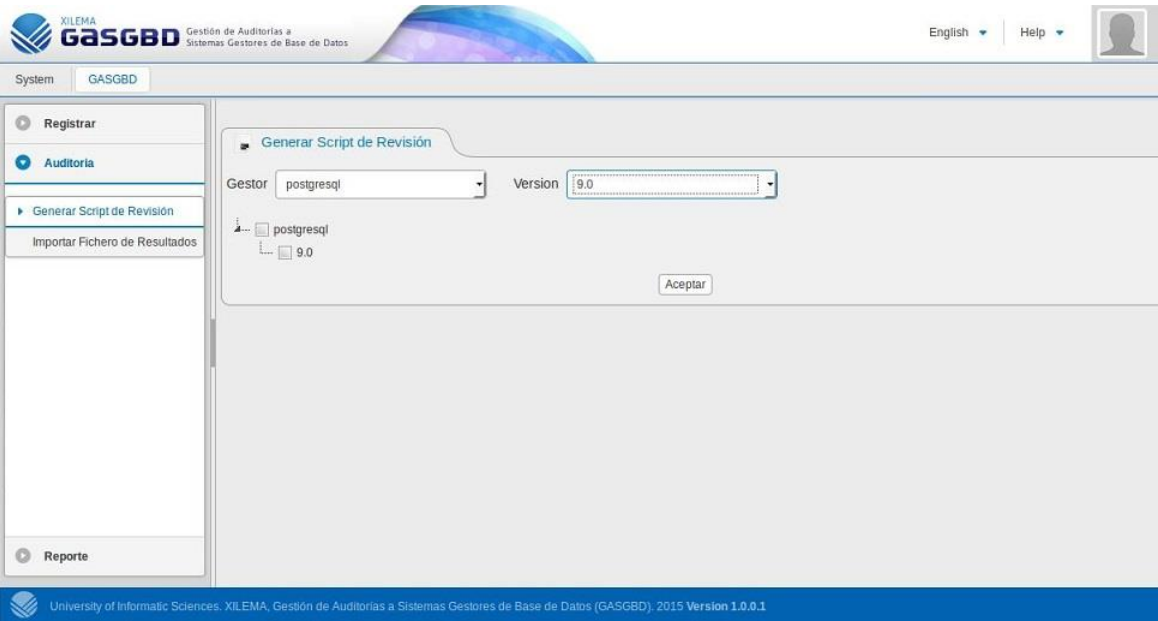
Historia de Usuario	
Número: 2	Usuario: Supervisor
Nombre de Historia de Usuario: Generar script de revisión	
Prioridad en negocio: Alta	Riesgo de Desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: El supervisor selecciona la opción “Generar Script de Revisión” del menú “Auditoría” y selecciona las opciones que desea exportar.	
Observaciones: El supervisor tiene que estar previamente autenticado.	
Interfaz:	
	

Tabla 13. Historia de Usuario # 2.

Historia de Usuario

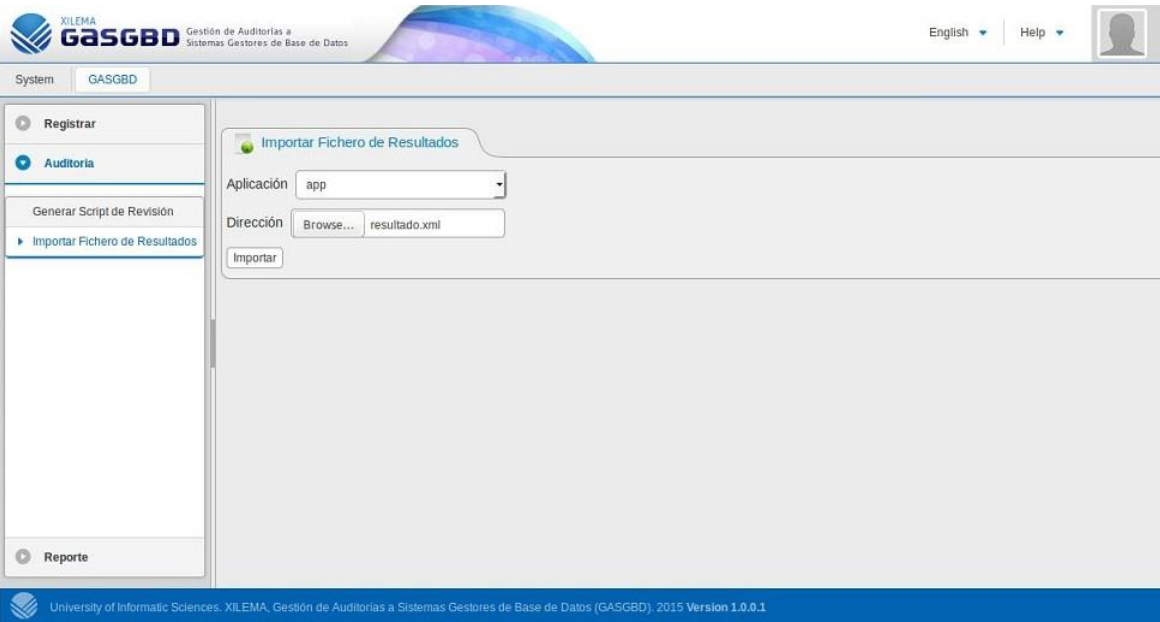
Número: 3	Usuario: Supervisor
Nombre de Historia de Usuario: Importar fichero de resultados	
Prioridad en negocio: Alta	Riesgo de Desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: El supervisor selecciona la opción “Importar Fichero de Resultado” del menú “Auditoría” y una vez importado se conforma la matriz de resultado.	
Observaciones: El supervisor tiene que estar previamente autenticado.	
Interfaz:	
	

Tabla 14. Historia de Usuario # 3.

Historia de Usuario	
Número: 4	Usuario: Supervisor
Nombre de Historia de Usuario: Concluir matriz	
Prioridad en negocio: Media	Riesgo de Desarrollo: Media

Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: El supervisor selecciona la opción “Concluir Matriz” del menú “Reporte” y selecciona la matriz a concluir.	
Observaciones: El supervisor tiene que estar previamente autenticado.	
Interfaz:	

Tabla 15. Historia de Usuario # 4.

Historia de Usuario	
Número: 5	Usuario: Supervisor
Nombre de Historia de Usuario: Generar Informe	
Prioridad en negocio: Media	Riesgo de Desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	

Descripción: El supervisor selecciona la opción “Generar Informe” del menú “Reporte” y selecciona las matrices que desea concluir.

Observaciones: El supervisor tiene que estar previamente autenticado.

Interfaz:

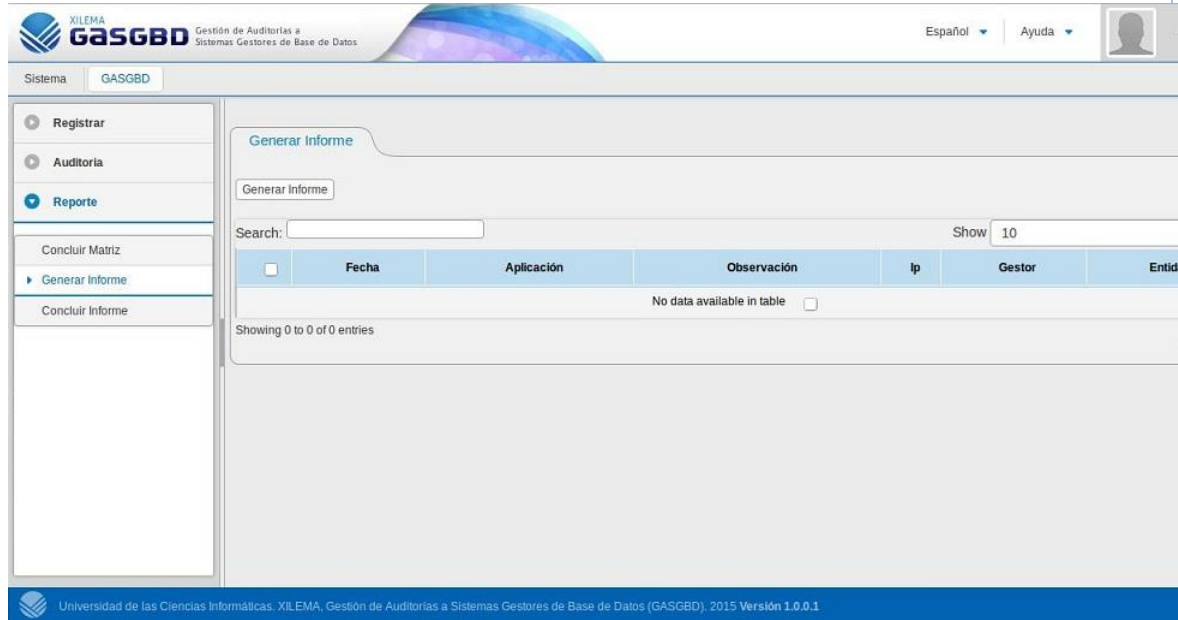


Tabla 16. Historia de Usuario # 5.

Anexos II Muestra de Tareas de Ingeniería

Tareas de ingeniería definidas y las HU correspondiente a cada una:

HU# 2 Generar Script de Revisión	
Tareas de Ingeniería	
1	Seleccionar los gestores existentes en la BD.
2	Seleccionar las versiones existentes en la BD a partir del gestor escogido.
3	Crear y mostrar interfaz gráfica para generar el script.
4	Seleccionar la ubicación donde se desea guardar el script.

Tabla 17. Tareas de Ingeniería HU# 2.

HU# 3 Importar Fichero de Resultado	
Tareas de Ingeniería	
1	Seleccionar las aplicaciones existentes en la BD.
2	Seleccionar la ruta del fichero a importar.
3	Analizar los datos del fichero importado.
4	Crear y mostrar interfaz gráfica con confirmación con las entidades anfitrionas existentes asociadas a los datos del fichero.
5	Realizar una evaluación con los resultados de la auditoría y se conforma la matriz correspondiente.
6	Crear y mostrar interfaz para la matriz conformada.
7	Verificar los datos a insertar.

8	Insertar datos en la BD.
9	Mostrar mensaje de confirmación.

Tabla 18. Tareas de Ingeniería HU# 3.

HU# 4 Concluir Matriz	
Tareas de Ingeniería	
1	Seleccionar los datos de la BD asociados a una matriz inconclusa.
2	Crear y mostrar interfaz con las matrices encontradas.
3	Crear y mostrar interfaz con los datos de la matriz a editar.
4	Validar los datos a insertar en la BD.
5	Insertar datos en la BD.
6	Crear y mostrar confirmación de matriz finalizada.
7	Calcular el riesgo resultante de la matriz finalizada.
8	Insertar datos en la BD.
9	Mostrar mensaje de operación satisfactoria y riesgo resultante de la matriz.

Tabla 19. Tareas de Ingeniería HU# 4.

HU# 5 Generar Informe	
Tareas de Ingeniería	
1	Seleccionar los datos de la BD asociados a una matriz finalizada.
2	Crear y mostrar interfaz con las matrices encontradas.
3	Crear y mostrar interfaz con los datos con vista a informe.
4	Validar los datos a insertar en la BD.

5	Insertar datos del informe en la BD.
---	--------------------------------------

Tabla 20. Tareas de Ingeniería HU# 5.

Tareas de Ingeniería	
Número de la tarea: 2	Número HU: 1
Nombre Tarea: Registrar Servidor.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: <ul style="list-style-type: none"> • Seleccionar los sistemas operativos existentes en la base de datos. • Seleccionar de la base de datos las versiones del sistema operativo escogido. • Seleccionar entidades existentes de la base de datos. • Seleccionar de la base de datos la dirección de la entidad a partir de la escogida. • Crear y mostrar interfaz gráfica para la inserción de un servidor. • Verificar los datos a insertar. • Insertar el servidor en la base de datos. • Mostrar mensaje de inserción correcta. 	

Tabla 21. Tarea 2 de la HU# 1.

Tareas de Ingeniería	
Número de la tarea: 3	Número HU: 1
Nombre Tarea: Registrar Indicador.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	

Descripción:

- Crear y mostrar interfaz gráfica para la inserción de un indicador.
- Verificar los datos a insertar.
- Insertar el indicador en la base de datos.
- Mostrar mensaje de inserción correcta.

Tabla 22. Tarea 3 de la HU# 1.

Tareas de Ingeniería	
Número de la tarea: 4	Número HU: 1
Nombre Tarea: Registrar Entidad.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción:	
<ul style="list-style-type: none">• Crear y mostrar interfaz gráfica para la inserción de una entidad.• Verificar los datos a insertar.• Insertar la entidad en la base de datos.• Mostrar mensaje de inserción correcta.	

Tabla 23. Tarea 4 de la HU# 1.

Tareas de Ingeniería	
Número de la tarea: 5	Número HU: 1
Nombre Tarea: Registrar Parámetro	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción:	
<ul style="list-style-type: none">• Seleccionar los indicadores existentes en la BD.	

- Seleccionar los tipos de parámetro existentes en la BD.
- Seleccionar los impactos existentes en la BD.
- Crear y mostrar interfaz gráfica para la inserción de una entidad.
- Verificar los datos a insertar.
- Insertar parámetro en la BD.
- Mostrar mensaje de inserción correcta.

Tabla 24. Tarea 5 de la HU# 1.

Tareas de Ingeniería	
Número de la tarea: 1	Número HU: 2
Nombre Tarea: Seleccionar los gestores existentes en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se seleccionan los gestores existentes en la BD.	

Tabla 25. Tarea 1 de la HU# 2.

Tareas de Ingeniería	
Número de la tarea: 2	Número HU: 2
Nombre Tarea: Seleccionar las versiones existentes en la BD a partir del gestor escogido.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se selecciona la dirección de la entidad escogida.	

Tabla 26. Tarea 2 de la HU# 2.

Tareas de Ingeniería	
Número de la tarea: 3	Número HU: 2

Nombre Tarea: Crear y mostrar interfaz gráfica para generar el script.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y muestra una interfaz gráfica para generar el script.	

Tabla 27. Tarea 3 de la HU# 2.

Tareas de Ingeniería	
Número de la tarea: 4	Número HU: 2
Nombre Tarea: Seleccionar la ubicación donde se desea guardar el script.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se selecciona la ubicación donde se desea guardar el script.	

Tabla 28. Tarea 4 de la HU# 2.

Tareas de Ingeniería	
Número de la tarea: 1	Número HU: 3
Nombre Tarea: Seleccionar las aplicaciones existentes en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se seleccionan las aplicaciones existentes en la BD.	

Tabla 29. Tarea 1 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 2	Número HU: 3
Nombre Tarea: Seleccionar la ruta del fichero a importar.	

Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se selecciona la ruta donde se encuentra el fichero.	

Tabla 30. Tarea 2 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 3	Número HU: 3
Nombre Tarea: Analizar los datos del fichero importado.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se analizan y confirman los datos del fichero cargado.	

Tabla 31. Tarea 3 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 4	Número HU: 3
Nombre Tarea: Crear y mostrar interfaz gráfica con confirmación con las entidades anfitrionas existentes asociadas a los datos del fichero.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y muestra una interfaz para dar confirmación de la entidad asociada a la cargada del fichero.	

Tabla 32. Tarea 4 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 5	Número HU: 3

Nombre Tarea: Realizar una evaluación con los resultados de la auditoría y se conforma la matriz correspondiente.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se realiza una evaluación con los resultados de la auditoría para conformar la matriz correspondiente.	

Tabla 33. Tarea 5 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 6	Número HU: 3
Nombre Tarea: Crear y mostrar interfaz para la matriz conformada.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y muestra una interfaz gráfica para la matriz conformada.	

Tabla 34. Tarea 6 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 7	Número HU: 3
Nombre Tarea: Verificar los datos a insertar.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se verifican los datos a insertar.	

Tabla 35. Tarea 7 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 8	Número HU: 3

Nombre Tarea: Insertar datos en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se insertan los datos en la BD.	

Tabla 36. Tarea 8 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 9	Número HU: 3
Nombre Tarea: Mostrar mensaje de confirmación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se muestra un mensaje de operación satisfactoria.	

Tabla 37. Tarea 9 de la HU# 3.

Tareas de Ingeniería	
Número de la tarea: 1	Número HU: 4
Nombre Tarea: Seleccionar los datos de la BD asociados a una matriz inconclusa.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se seleccionan los datos a partir de una matriz inconclusa.	

Tabla 38. Tarea 1 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 2	Número HU: 4

Nombre Tarea: Crear y mostrar interfaz con las matrices encontradas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y se muestra una interfaz de usuario con las matrices encontradas.	

Tabla 39. Tarea 2 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 3	Número HU: 4
Nombre Tarea: Crear y mostrar interfaz con los datos de la matriz a editar.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y se muestra una interfaz de usuario con los datos de la matriz seleccionada.	

Tabla 40. Tarea 3 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 4	Número HU: 4
Nombre Tarea: Validar los datos a insertar en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se verifican los datos a insertar.	

Tabla 41. Tarea 4 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 5	Número HU: 4

Nombre Tarea: Insertar datos en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se insertan los datos en la BD.	

Tabla 42. Tarea 5 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 6	Número HU: 4
Nombre Tarea: Crear y mostrar confirmación de matriz finalizada.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se muestra un mensaje de confirmación.	

Tabla 43. Tarea 6 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 7	Número HU: 4
Nombre Tarea: Calcular el riesgo resultante de la matriz finalizada.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se calcula el riesgo resultante de la matriz finalizada	

Tabla 44. Tarea 7 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 8	Número HU: 4
Nombre Tarea: Insertar datos en la BD.	

Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se insertan los datos en la BD.	

Tabla 45. Tarea 8 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 9	Número HU: 4
Nombre Tarea: Mostrar mensaje de operación satisfactoria.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se muestra un mensaje de operación satisfactoria.	

Tabla 46. Tarea 9 de la HU# 4.

Tareas de Ingeniería	
Número de la tarea: 1	Número HU: 5
Nombre Tarea: Seleccionar los datos de la BD asociados a una matriz finalizada.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se seleccionan los datos de una matriz finalizada.	

Tabla 47. Tarea 1 de la HU# 5.

Tareas de Ingeniería	
Número de la tarea: 2	Número HU: 5
Nombre Tarea: Crear y mostrar interfaz con las matrices encontradas.	

Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y muestra la interfaz con las matrices encontradas.	

Tabla 48. Tarea 2 de la HU# 5.

Tareas de Ingeniería	
Número de la tarea: 3	Número HU: 5
Nombre Tarea: Crear y mostrar interfaz con los datos con vista a informe.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se crea y muestra la interfaz con los datos con vista a informe.	

Tabla 49. Tarea 3 de la HU# 5.

Tareas de Ingeniería	
Número de la tarea: 4	Número HU: 5
Nombre Tarea: Validar los datos a insertar en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Programador responsable: Félix Duque López, Yassel Pérez Camacho.	
Descripción: Se verifican los datos a insertar en la BD.	

Tabla 50. Tarea 4 de la HU# 5.

Tareas de Ingeniería	
Número de la tarea: 5	Número HU: 5
Nombre Tarea: Insertar datos del informe en la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2

Programador responsable: Félix Duque López, Yassel Pérez Camacho.

Descripción: Se insertan los datos en la BD.

Tabla 51. Tarea 5 de la HU# 5.

Anexos III Muestra de Pruebas de Aceptación

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El supervisor selecciona del menú "Auditoría" la opción "Importar Fichero de Resultado". Selecciona una aplicación y un fichero de resultado formato XML.		Se muestra un mensaje "Matriz guardada satisfactoriamente".	Satisfactorio	
	El supervisor selecciona del menú "Auditoría" la opción "Importar Fichero de Resultado". Pero no selecciona un fichero de resultado.	El sistema muestra un mensaje de error "Debe seleccionar una aplicación y un fichero de resultado".	No se crea la matriz resultado.	

Tabla 52. Prueba de Aceptación #5: HU: Importar fichero de resultados.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
<p>El supervisor selecciona del menú "Reporte" la opción "Concluir Matriz".</p> <p>Selecciona una matriz de las listadas y se completan los campos "Evaluación de Usuario", "Recomendación", "Vista a Informe" y se selecciona la opción "Concluir Matriz"</p>		<p>Se muestra un mensaje "Matriz guardada y concluida satisfactoriamente"</p>	<p>Satisfactorio</p>	
	<p>El supervisor selecciona del menú "Reporte" la opción "Concluir Matriz".</p> <p>Selecciona una matriz de las listadas, pero no se completan los campos</p>	<p>El sistema muestra un mensaje de error "La matriz no se puede concluir aún".</p>	<p>No se concluye la matriz.</p>	

	<p>“Evaluación de Usuario”,</p> <p>“Recomendación”,</p> <p>“Vista a Informe” y se selecciona la opción “Concluir Matriz”</p>			
--	--	--	--	--

Tabla 53. Prueba de Aceptación #6: HU: Concluir Matriz.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
<p>El supervisor selecciona del menú “Auditoría” la opción “Importar Fichero de Resultados”. Selecciona una aplicación de las existentes y se selecciona el script de resultados que se desea importar.</p>		<p>Se muestra una interfaz con la matriz resultado creada a partir del script importado.</p>	Satisfactorio	
	<p>El supervisor selecciona del menú “Auditoría” la opción “Importar Fichero</p>	<p>El sistema muestra un mensaje de error “Debe seleccionar un fichero XML”.</p>	<p>No se construye la matriz de resultados.</p>	

	de Resultados". Selecciona una aplicación de las existentes pero no un archivo en formato XML.			
--	---	--	--	--

Tabla 54. Prueba de Aceptación #7: HU: Importar Fichero de Resultados.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación
El supervisor selecciona del menú "Reporte" la opción "Generar Informe". Seleccionan las matrices concluidas para generar el informe.		Se muestra una interfaz con los datos para construir un informe.	Satisfactorio	
	El supervisor selecciona del menú "Reporte" la opción "Generar Informe". Pero no selecciona ninguna matriz.	El sistema muestra un mensaje de error "Debe seleccionar al menos una matriz".	No se puede generar el informe.	

Tabla 55. Prueba de Aceptación #8: HU: Generar Informe.