



Universidad de las Ciencias Informáticas

Facultad 1

**Título: Solución informática para mejorar la calidad de los datos en el Sistema de
Gestión Universitaria.**

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores:

Dayliana Guerra Velázquez
Juan Daniel de la Vega Valenzuela

Tutores:

Ing. Lianet Liben Martínez
Ing. Dianela Galardy Tarragó
Ing. Yoan Carlos Machado Espinosa

La Habana, 2013.



Declaramos ser los autores del trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas titulado Solución informática para mejorar la calidad de los datos en el Sistema de Gestión Universitaria y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma el presente a los ____ días del mes de ____ del año 2013.

Dayliana Guerra Velázquez

Juan Daniel de la Vega Valenzuela

Ing. Lianet Liben Martínez

Ing. Dianela Galardy Tarragó

Ing. Yoan Carlos Machado Espinosa



Agradecimientos

A mis padres porque siempre me han dado todo cuanto han podido y porque confían en mí.

A toda mi familia porque desde que empecé en esta universidad me han apoyado y estado al pendiente de mí.

A mis compañeras de apartamento y en especial a Arianna, Celia e Ismeidy; las quiero mucho.

A todos mis compañeros de aula que me brindaron su apoyo para cada examen aunque sea en un pequeño detalle, en especial a Sael, Adrián, Heidi y Yudeikis.

Agradezco el apoyo de mis tutores de la asignatura Práctica Profesional, a todos los profesores del laboratorio que en algún momento me respondieron alguna duda y a los demás profesores que durante el transcurso de la carrera ayudaron a mi formación como profesional.

A mis tutores de tesis porque fueron de gran ayuda para la realización de este trabajo de diploma y porque desde el principio confiaron en mí.

A mi novio por el amor que me ha dado todo este tiempo y por haber aportado sus granitos de arena en este trabajo de diploma.

Agradezco a Dania Márquez por la acogida que me dio en su hogar y por sus buenos consejos.

A mi compañero de tesis por haberme mostrado el camino por el que hoy transito y porque sin él este trabajo de diploma no se hubiera podido realizar.

Agradezco sobre todo al Dios Todopoderoso que me sustenta a cada paso que doy y porque en Él fui hecha nueva.

Dayliana



Agradezco en primer lugar a mi Padre Celestial, por ser mi refugio y apoyo durante estos cinco años, a mis padres por su apoyo incondicional y por la educación que me dieron, a mi primos Pedrito y Tania porque sin su ayuda no hubiera llegado tan lejos, a mi tíos Manolo, Mirella y Juan Antonio por la confianza que depositaron en mí, a todas mi primas, a Maye, Miki, Patri, Yadira y Marbelis que me han apoyado en todo tiempo. A mi novia por permanecer a mi lado durante todo este largo tiempo, porque en ella encontré apoyo, ayuda y consejo, gracias por tu fidelidad, comprensión y amor, te amo.

A mi amigo Darian Sánchez Sarria, a Evidio, Gladis y Darielito un millón de gracias porque Uds. han sido como una familia para mí, eternamente agradecido por todo lo que han hecho por mí, los quiero.

A Roberto Pérez Ozeti mi viejo amigo, a mis vecinos Marilín y su hija Yanet, Teresita, Betty y Rafael eternamente agradecidos por cuidar de mis padres cuando estuvieron enfermos.

A mis hermanos en la fe Antonio, Annarelis, Eunice, Drucila, les doy todo mi agradecimiento.

A mis suegros Carlos y Darminda por el apoyo brindado.

A los amigos que hice en la universidad, Leo, Nio, Yoan, Dayliana, Heidy, Lenier, Sael, Diosbel, Carlos gracias por su amistad, Uds. son especiales.

A mis queridos tutores, un inmenso agradecimiento, por el tiempo que dedicaron para que este trabajo saliera adelante, por todo lo que me enseñaron.

A mi compañera de tesis por todo el empeño que puso en el trabajo, por su paciencia y perseverancia.

A todos mis profesores, en especial a Edistio, Daysniel, Angel por todo lo que me enseñaron.

A los profes Alexander y Norges un agradecimiento especial por la confianza que depositaron en mí, y por todo lo que me permitieron aprender en el proyecto.

A todos mis más profundos agradecimientos.

Juan Daniel



Dedicatoria

A Dios porque tuvo misericordia de mí y me sacó de las tinieblas cuando más lo necesitaba, por su infinita gracia y porque sin Él no hubiera llegado hasta aquí.

A mis padres, a mi hermanita y a mi abuela por ser las personas que más amo en la vida, porque siempre han estado conmigo a pesar de haber estado prácticamente separados durante estos cinco años.

A mi novio por su apoyo incondicional y por su amor verdadero, te amo...

Dayliana

A Dios.

A mis padres y familiares.

A mi novia.

A mis amigos.

Juan Daniel



Resumen:

El Centro de Informatización Universitaria (CENIA) de la Universidad de las Ciencias Informáticas (UCI), con el objetivo de informatizar las áreas de procesos claves de la universidad desarrolla el Sistema de Gestión Universitaria (SGU), este sistema tiene la necesidad de gestionar gran cantidad de datos y de hacer persistente la información que maneja. Actualmente el SGU maneja datos que en algunos casos poseen inconsistencia y son inestables, además no existe un mecanismo eficiente para la implementación y chequeo de las reglas del negocio. Por la existencia de estos problemas el objetivo de la presente investigación es desarrollar una solución informática para mejorar la calidad de los datos en el SGU mediante acciones que se ejecutan sobre la información almacenada cuando un usuario lo requiera o de manera automática. La solución fue desarrollada completamente utilizando tecnologías establecidas por el Grupo de Arquitectura del CENIA en el proceso de desarrollo de *software* con enfoque ágil basado en el nivel 2 de CMMI, donde el enfoque ágil está dado por el uso de prácticas de las metodologías ágiles Scrum y XP; todo esto contribuyó a la obtención de los artefactos propuestos por el proceso de desarrollo.

Palabras clave: calidad de los datos, datos, reglas del negocio.



Índice

Introducción	1
Capítulo 1. Fundamentación teórica	5
Introducción.....	5
1.1. Conceptos asociados a la investigación.....	5
1.1.1. Base de datos relacional.....	5
1.1.3. Características de la calidad de los datos	6
1.1.4. Reglas del Negocio.....	6
1.1.5. Técnicas de evaluación y corrección de datos.....	7
1.2. Herramientas para la mejora de la calidad de los datos	8
1.2.1. DQGuru (antes llamada Power MatchMaker)	8
1.2.2. Productos Data Quality de Uniserv	9
1.2.3. Data Cleaner	10
1.2.4. Trillium Software System	11
1.3. Valoración de las herramientas para mejorar la calidad de los datos.....	12
1.4. Tecnologías seleccionadas.....	13
1.4.1. Lenguajes	13
1.4.2. Herramientas	15
1.4.3. Proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI	17
Conclusiones parciales.....	19
Capítulo 2. Diseño de la solución.....	20
Introducción.....	20
2.1. Modelo del dominio.....	20
2.1.1. Descripción de las clases del dominio	21
2.2. Presentación de la propuesta de solución.....	21
2.3. Técnicas de obtención de requisitos	22
2.4. Definición de los requerimientos funcionales y no funcionales.....	23
2.4.1. Requisitos funcionales	23
2.4.2. Especificación de requisitos funcionales	25
2.4.3. Requisitos no funcionales	27
2.5. Descripción de la arquitectura y el diseño.....	28
2.5.1. Estilo arquitectónico.....	28
2.5.2. Modelo de despliegue.....	29
2.5.3. Patrones de diseño.....	31
2.5.4. Patrones de Base de datos	33



2.6.	Estándares de diseño	34
2.6.1.	Diagramación de las vistas	34
2.6.2.	Tipos de mensajes.....	36
2.7.	Mapa taxonómico	38
2.8.	Modelo de datos	39
	Conclusiones parciales.....	40
Capítulo 3.	Implementación y prueba.....	42
	Introducción.....	42
3.1.	Estándares de codificación	42
3.1.1.	Indentación, llaves de apertura y cierre, tamaño de líneas	42
3.1.2.	Conversión de nomenclaturas	43
3.1.3.	Sentencias de control	44
3.2.	Técnicas de programación.....	45
3.3.	Tratamiento de errores	46
3.4.	Validación de la solución propuesta.....	47
3.4.1.	Validación de requisitos	47
3.4.2.	Pruebas de <i>software</i>	49
3.4.3.	Resultado de las pruebas	51
	Conclusiones parciales.....	56
	Conclusiones	57
	Recomendaciones	58
	Bibliografía referenciada	59
	Bibliografía consultada.....	63
	Glosario de términos.....	64
	Anexos.....	65



Índice de Figuras

Figura 1: Modelo del dominio.	20
Figura 2: Mapa conceptual.	22
Figura 3: Arquitectura Tres Capas.	29
Figura 4: Diagrama de despliegue.	30
Figura 5: Ejemplo del uso del patrón Instancia única.	32
Figura 6: Ejemplo del uso del patrón Iterador.	33
Figura 7: Ejemplo del patrón Árbol fuertemente codificado.	34
Figura 8: Ejemplo del patrón Llaves subrogadas.	34
Figura 9: Áreas de la vista de presentación.	35
Figura 10: Áreas de la vista de gestión.	35
Figura 11: Áreas de la vista de funcionalidades.	36
Figura 12: Mensaje de información.	36
Figura 13: Mensaje de error en forma de ventana.	37
Figura 14: Mensaje de confirmación.	38
Figura 15: Mapa taxonómico.	39
Figura 16: Modelo físico de datos.	40
Figura 17: Ejemplo de indentación, llaves de apertura y cierre.	43
Figura 18: Ejemplo de variables.	43
Figura 19: Ejemplo de una clase.	43
Figura 20: Ejemplo de una función.	44
Figura 21: Ejemplo de una sentencia <i>if</i>	44



Figura 22: Ejemplo de una sentencia <i>if else</i>	44
Figura 23: Ejemplo de una sentencia <i>if else if else</i>	44
Figura 24: Ejemplo de una sentencia <i>for</i>	45
Figura 25: Ejemplo de una sentencia <i>while</i>	45
Figura 26: Ejemplo de una sentencia <i>try catch</i>	45
Figura 27: Resultado de las pruebas funcionales.....	52
Figura 28: Modelo lógico de datos.	66



Índice de Tablas

Tabla 1. Características de la calidad de los datos.	6
Tabla 2. Requisitos funcionales.	24
Tabla 3. Especificación de requisito. Crear tarea.	25
Tabla 4. Requisitos no funcionales.	27
Tabla 5. Casos de prueba de caja blanca. RF2.	51
Tabla 6. Diseño de casos de prueba. Crear tarea. Parte uno.....	53
Tabla 7. Diseño de casos de prueba. Crear tarea. Parte dos.....	54
Tabla 8. Modelo de entrevista.....	65
Tabla 9. Especificación de requisito. Mostrar configuraciones.	66
Tabla 10. Especificación de requisito. Crear configuración.	67
Tabla 11. Especificación de requisito. Modificar configuración.....	69
Tabla 12. Especificación de requisito. Eliminar configuración.	71
Tabla 13. Especificación de requisito. Mostrar objetos del negocio.....	73
Tabla 14. Especificación de requisito. Crear objetos del negocio.....	74
Tabla 15. Especificación de requisito. Modificar objeto del negocio.	79
Tabla 16. Especificación de requisito. Ver detalles de objeto del negocio.....	84
Tabla 17. Especificación de requisito. Eliminar objeto del negocio.....	85
Tabla 18. Especificación de requisito. Eliminar propiedad.....	87
Tabla 19. Especificación de requisito. Eliminar unión de tablas.	88
Tabla 20. Especificación de requisito. Mostrar reglas.	89
Tabla 21. Especificación de requisito. Crear regla.	90
Tabla 22. Especificación de requisito. Modificar_regla.....	100



Tabla 23. Especificación de requisito. Eliminar regla.	108
Tabla 24. Especificación de requisito. Ejecutar regla.	109
Tabla 25. Especificación de requisito. Mostrar tareas.	110
Tabla 26. Especificación de requisito. Modificar tarea.....	111
Tabla 27. Especificación de requisito. Eliminar tarea.	113
Tabla 28. Especificación de requisito. Ver detalles de tarea.	114
Tabla 29. Especificación de requisito. Mostrar reportes.	115
Tabla 30. Especificación de requisito. Ver detalles de reporte inicial.....	116
Tabla 31. Especificación de requisito. Ver detalles de reporte final.....	117
Tabla 32. Casos de prueba de caja blanca. RF1.....	118
Tabla 33. Casos de prueba de caja blanca. RF3.....	119
Tabla 34. Casos de prueba de caja blanca. RF4.....	119
Tabla 35. Casos de prueba de caja blanca. RF17.....	120
Tabla 36. Casos de prueba de caja blanca. RF18.....	121
Tabla 37. Casos de prueba de caja blanca. RF19.....	121
Tabla 38. Casos de prueba de caja blanca. RF20.....	122
Tabla 39. Diseño de casos de prueba. Mostrar configuraciones.	124
Tabla 40. Diseño de casos de prueba. Crear configuración.	124
Tabla 41. Diseño de casos de prueba. Modificar configuración.....	130
Tabla 42. Diseño de casos de prueba. Eliminar configuración.	137
Tabla 43. Diseño de casos de prueba. Mostrar objetos del negocio.	138
Tabla 44. Diseño de casos de prueba. Crear objeto del negocio. Parte uno.	139
Tabla 45. Diseño de casos de prueba. Crear objeto del negocio. Parte dos.	143



Tabla 46. Diseño de casos de prueba. Modificar objeto del negocio. Parte uno.....	146
Tabla 47. Diseño de casos de prueba. Modificar objeto del negocio. Parte dos.....	149
Tabla 48. Diseño de casos de prueba. Ver detalles de objeto del negocio.....	153
Tabla 49. Diseño de casos de prueba. Eliminar objeto del negocio.	154
Tabla 50. Diseño de casos de prueba. Eliminar propiedad.	156
Tabla 51. Diseño de casos de prueba. Eliminar unión de tablas.	158
Tabla 52. Diseño de casos de prueba. Mostrar reglas.	160
Tabla 53. Diseño de casos de prueba. Crear regla. Parte uno.	161
Tabla 54. Diseño de casos de prueba. Crear regla. Parte dos.	165
Tabla 55. Diseño de casos de prueba. Modificar regla. Parte uno.	169
Tabla 56. Diseño de casos de prueba. Modificar regla. Parte dos.....	172
Tabla 57. Diseño de casos de prueba. Eliminar regla.	175
Tabla 58. Diseño de casos de prueba. Ejecutar regla.	176
Tabla 59. Diseño de casos de prueba. Mostrar tareas.	177
Tabla 60. Diseño de casos de prueba. Modificar tarea.	178
Tabla 61. Diseño de casos de prueba. Eliminar tarea.	183
Tabla 62. Diseño de casos de prueba. Ver detalles de tarea.	185
Tabla 63. Diseño de casos de prueba. Mostrar reportes.	186
Tabla 64. Diseño de casos de prueba. Ver detalles de reporte inicial.	187
Tabla 65. Diseño de casos de prueba. Ver detalles de reporte final.....	188



Introducción

El desarrollo tecnológico alcanzado en las últimas décadas ha traído consigo un aumento en la automatización de procesos de la actividad humana diaria mediante el uso de *software*, estos constituyen el soporte lógico e inmaterial que permite a la computadora desempeñar tareas inteligentes.[1] Existen varias clasificaciones de *software* según la función que realizan, entre ellas se encuentran los de aplicación, que son programas diseñados para o por los usuarios con el fin de facilitar la realización de tareas específicas en la computadora. Desempeñan un papel primordial en el desarrollo individual y corporativo, estando presentes en todos los niveles de la sociedad tales como: la economía, el deporte, la medicina y la educación.

Actualmente las empresas se enfrentan a un crecimiento exponencial en cuanto a la cantidad y diversidad de datos a gestionar, debido a las tendencias tecnológicas en la captura y almacenamiento de datos, provocando que el desarrollo de aplicaciones informáticas orientadas a la gestión de las bases de datos se haya convertido en un fenómeno generalizado, que forma una parte indispensable dentro de la dinámica de desarrollo social. La afirmación anterior está fundamentada en lo que significa para cualquier entidad, ya sea una empresa o persona, el poder almacenar y gestionar de manera eficiente toda la información que circula por los escenarios con los que interactúa, donde han desempeñado un papel fundamental los Sistemas Gestores de Bases de Datos (SGBD), los que se caracterizan por estar dedicados a integrar distintos tipos de datos, compartirlos entre diferentes aplicaciones y usuarios, almacenarlos y posteriormente acceder a ellos de forma rápida y estructurada. El propósito general de estos sistemas es el de administrar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Al mismo tiempo que los SGBD han traído innumerables ventajas, los datos que estos manejan tienen un ciclo de vida mayor que el del *software* y durante el mismo están sometidos a procesos de actualización y modificación; procesos que son propensos a introducir errores en los datos, ya sea por fallas humanas o tecnológicas, provocando pérdidas en las características de la calidad de los datos.[2]

En Cuba se ha incrementado el desarrollo de soluciones informáticas que emplean como SGBD a PostgreSQL, respondiendo principalmente, a la política de migración a plataformas libres. La Universidad de las Ciencias Informáticas (UCI) se desempeña como principal impulsor de la aplicación de estas tecnologías en el país y con el objetivo de aumentar la productividad ha creado centros de desarrollo, dirigidos a la producción de *software* en diferentes áreas, uno de ellos es el Centro de Informatización Universitaria (CENIA), constituido por diferentes departamentos como el de Gestión Universitaria, que



tiene entre sus objetivos informatizar las áreas de procesos claves de la universidad, para lo cual desarrolla el Sistema de Gestión Universitaria (SGU), que como todo sistema de gestión tiene la necesidad de hacer persistente la información que maneja, así como la gestión constante de gran cantidad de datos, los que una vez capturados, procesados y almacenados deben ser un fiel reflejo de la realidad. Además es vital que dichos datos conserven en todo momento su calidad.

Actualmente el SGU maneja información que en algunos casos posee inconsistencia, producto de procesos externos tales como la migración, la replicación, la sincronización y la entrada manual de datos. La inconsistencia está presente cuando un dato que se encuentra almacenado varias veces, posee valores diferentes y por tanto si se consulta el mismo dato desde lugares distintos en el sistema, arroja resultados diferentes, constituyendo un error en la exactitud que deberían presentar los datos mostrados al usuario. Además el negocio manejado en este sistema se encuentra sujeto a constantes cambios que provocan inestabilidad en los datos almacenados. Los problemas mencionados anteriormente impiden o dificultan la toma de decisiones correctas basadas en información precisa; sumando a esto que para la implementación y chequeo de las reglas del negocio en las bases de datos las características brindadas por el SGBD utilizado no son suficientes.

A partir de lo antes expuesto se ha identificado el siguiente **problema científico**: ¿cómo mejorar la calidad de los datos almacenados en el Sistema de Gestión Universitaria?

En este contexto se define como **objeto de estudio**: calidad de los datos en Sistemas de Bases de Datos Relacionales. Enmarcado en el **campo de acción**: calidad de los datos en Sistemas de Bases de Datos Relacionales sobre el gestor de base de datos PostgreSQL.

Se define como **objetivo general** de la investigación: desarrollar una solución informática usando tecnologías libres para mejorar la calidad de los datos almacenados en el Sistema de Gestión Universitaria.

De donde se derivan los siguientes **objetivos específicos**:

- Caracterizar los principales elementos de la fundamentación teórica de la investigación.
- Diseñar una propuesta de solución configurable, que permita controlar la calidad de los datos a partir del cumplimiento de las reglas del negocio para la información generada en el Sistema de Gestión Universitaria.
- Desarrollar la propuesta de solución utilizando las herramientas y técnicas definidas.



- Validar la solución desarrollada.

Para dar cumplimiento a los objetivos específicos se trazan las **tareas de investigación**:

- Caracterización del proceso de mejora de la calidad de los datos a partir de la aplicación de reglas del negocio a nivel internacional y nacional.
- Análisis de sistemas que automaticen la mejora y mantenimiento de la calidad de los datos a partir de las reglas del negocio.
- Caracterización del proceso de desarrollo a utilizar.
- Caracterización de las herramientas que se utilizarán en el marco de trabajo.
- Análisis de técnicas de ingeniería de requisitos.
- Obtención de los requisitos de la propuesta de solución.
- Caracterización de la arquitectura y modelo de datos a usar.
- Diseño e implementación de la solución propuesta.
- Definición de tipos de pruebas a realizar.
- Aplicación de pruebas de *software* a la solución.

Para poder cumplir con las distintas tareas investigativas se emplearon los siguientes **métodos de investigación**:

Métodos Teóricos

- **Histórico-Lógico:** permitió conocer la evolución de las técnicas y herramientas para la mejora de la calidad de los datos y sus tendencias. Así como recopilar información para describir las tecnologías a utilizar.
- **Analítico-sintético:** se utilizó para el procesamiento de información y elaboración de conclusiones. Este método sirvió para analizar y comprender la teoría y documentación relacionada con el tema de investigación, permitiendo así extraer los elementos más importantes relacionados con el objeto de estudio.[3]

Métodos Empíricos

- **Entrevista:** ayudó a obtener información sobre el proceso de mejora de la calidad de los datos, comprender y precisar el problema a resolver, así como a validar la propuesta que se presentará al cliente.



- **Observación:** hizo posible detectar el insuficiente chequeo de las reglas del negocio usando el gestor de bases de datos PostgreSQL y los problemas de inestabilidad e inconsistencia en los datos, lo que ayudó al planteamiento del problema científico.[4]

Con la realización de la presente investigación se pretende obtener los siguientes **resultados:**

- Una solución informática que permita, aplicando reglas del negocio, mejorar la calidad de los datos almacenados en el Sistema de Gestión Universitaria.
- Documentación sobre la mejora de la calidad de los datos a partir de las reglas del negocio en el Sistema de Gestión Universitaria.

El presente trabajo, está estructurado en 3 capítulos, distribuidos de la siguiente forma:

Capítulo 1. Fundamentación Teórica: se realiza un análisis de la información existente acerca del seguimiento de la calidad de los datos en Sistemas de Bases de Datos Relacionales, abordando las características de las aplicaciones existentes en el mundo que responden al problema a resolver y por qué no son factibles. También se definen las tecnologías a utilizar para el desarrollo de la presente investigación.

Capítulo 2. Diseño de la solución: recoge la descripción y análisis de la solución propuesta, se describen las clases que modelan la solución y sus principales funcionalidades, así como los patrones usados, tanto de diseño como arquitectónicos. Se definen los requerimientos y especificaciones de usuarios.

Capítulo 3. Implementación y prueba: se describen los estándares de código fundamentales usados en el desarrollo de la herramienta y las validaciones realizadas a los requisitos antes de su implementación, así como las pruebas de *software* a la aplicación una vez terminada para verificar su correcto funcionamiento.



Capítulo 1. Fundamentación teórica

Introducción

En el presente capítulo se presentan los principales conceptos relacionados con la calidad de los datos en bases de datos relacionales. Se caracterizan algunas técnicas y herramientas que facilitan la mejora de la calidad de los datos mediante la automatización de acciones, así como algunos sistemas informáticos que gestionan y le dan seguimiento a la calidad de los datos. Además se describen las tecnologías a utilizar para el desarrollo de la solución dentro del entorno de desarrollo.

1.1. Conceptos asociados a la investigación

1.1.1. Base de datos relacional

Es una base de datos donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y donde todas las operaciones de la base de datos operan sobre estas tablas. Son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo. Estas bases de datos cumplen con el modelo relacional, que las representa en un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente.[5]

1.1.2. Calidad de los datos

En ocasiones la calidad de los datos es ignorada en las empresas y en el desarrollo, uso y mantenimiento de sus sistemas de información. Con demasiada frecuencia los datos se utilizan sin sentido crítico, sin consideración de los errores que pueden contener y esto puede conducir a resultados erróneos, información engañosa y decisiones desacertadas.[6]

Según el estándar ISO/IEC 25012 del 2008 la calidad de los datos se puede definir como: *“el grado en que las características de los datos guardan las condiciones y sugieren las necesidades cuando es usado bajo condiciones específicas”*. [7]

Otra definición es: *“el estado de integridad, validez, consistencia, puntualidad y exactitud que hace los datos apropiados para un uso específico”*. [8]

La calidad de los datos se mide según un conjunto de criterios o características definidas en el estándar antes mencionado.



1.1.3. Características de la calidad de los datos

La Tabla 1 muestra las características de la calidad de los datos que serán utilizadas en la presente investigación, según el estándar ISO/IEC 25012 del 2008, el cual las categoriza en 15 criterios considerados desde dos puntos de vistas:

- **Inherentes al dato:** la calidad de datos inherentes se refiere al grado en el cual las características de calidad de los datos tienen el potencial intrínseco para satisfacer las necesidades implicadas cuando el dato es usado bajo condiciones específicas.
- **Dependientes del sistema:** se refiere al grado en el cual la calidad del dato es enriquecida y preservada dentro de un sistema de cómputo cuando el dato es usado bajo condiciones específicas.[9]

Tabla 1. Características de la calidad de los datos.[10]

Característica	Descripción
Inherentes al dato	
Exactitud	El grado en el cual el dato tiene atributos que correctamente representan el valor adecuado del atributo intencionado de un concepto o evento en un contexto específico de uso.
Compleitud	El grado al cual el dato del sujeto asociado con una entidad tiene valores para todos los atributos esperados e instancias de entidad relacionadas en un contexto específico de uso.
Consistencia	El grado en el cual el dato tiene atributos que son libres de contradicción y son coherentes con otros datos en un contexto de uso específico.
Credibilidad	El grado en el cual el dato tiene atributos que son considerados verdaderos y creíbles por usuarios en un contexto de uso específico.

1.1.4. Reglas del Negocio

El Grupo de Reglas del Negocio (*BRG*, acrónimo de *Business Rules Group*), hace la siguiente definición general de regla del negocio: “una regla del negocio es una declaración que define o restringe algún aspecto del negocio; intenta definir, controlar o influenciar el comportamiento y la estructura del negocio”. [11]



Para el especialista en base de datos, es un requerimiento específico que se satisface mediante la definición de alguna característica de los datos.[12]

1.1.5. Técnicas de evaluación y corrección de datos.

Las técnicas de evaluación permiten el análisis y verificación de los datos, son necesarias para generar las métricas sobre el estado de la calidad de los datos analizados. La evaluación indicará la necesidad de tomar acciones correctivas para llevar a una organización hacia la calidad de los datos deseada.

1.1.5.1. Técnicas de evaluación de datos.

Verificación manual: consiste en revisar manualmente en la fuente original de la información, verificando todos y cada uno de los valores. Sólo la verificación manual puede, en teoría, localizar la totalidad de los errores. Tiene como desventajas que el proceso manual es susceptible de error, no garantiza la detección total y es muy lento y costoso. Para la mayoría de los casos no es práctico y no es posible aplicarlo.[13]

Técnicas analíticas: utilizan *software* y la habilidad del analista de calidad de datos para detectar los datos inexactos. Las técnicas analíticas se pueden aplicar teóricamente a bases de datos en producción. No pueden determinar si un valor es correcto al menos que puedan consultar una fuente alternativa para confirmar el valor. Las técnicas analíticas, bien aplicadas, identifican suficientes errores para dar una idea clara del estado de calidad de los datos. No pueden detectar todas las inexactitudes en los datos de una base de datos.

Perfilado de datos: es un proceso para examinar y evaluar los datos que existen en las fuentes de origen de una organización y recopilar estadísticas e información sobre los mismos, el propósito de dichas estadísticas es:

- Determinar qué datos pueden ser usados para otros propósitos.
- Conseguir métricas de calidad de datos que incluyen si los datos cumplen los estándares de la organización.
- Reducir el riesgo de integrar información a nuevas aplicaciones dado que se conoce su estado.
- Permite hacer un seguimiento de la calidad de los datos.

La entrada al proceso del perfilado de datos son los datos (exactos o inexactos) procedentes de bases de datos transaccionales y la información de metadato, por ejemplo un diccionario de datos.

Los métodos aplicados para analizar los datos son:



Descubrimiento: con *software*, se revelan las características de los datos a partir de los mismos. Es análogo a hacer minería de datos para reconstruir el metadato.

Pruebas asertivas: se formulan condiciones verdaderas (reglas) sobre los datos y se prueban con el *software*. Permite determinar donde difieren los datos del metadato y corregirlo.

El proceso reconstruye el metadato a partir del conocimiento real de la base de datos y genera un reporte de estado de la calidad de los datos, sobre el que se formulan recomendaciones. No corrige datos, sólo diagnostica e identifica anomalías documentadas en el repositorio de metadato.[14]

1.1.5.2. Técnicas de corrección de datos

Limpieza de datos: es un proceso cuyo principal objetivo es detectar y remover errores e inconsistencias en los datos, para proveer una mejor calidad de los mismos. Permite detectar entradas duplicadas, incompletas y establecer reglas para corregirlas. El proceso de limpieza de datos cuenta con las siguientes fases:

- Análisis de datos.
- Definición del flujo de trabajo de transformaciones y reglas de mapeos.
- Verificación.
- Transformación.
- Reemplazo de los datos sucios en las fuentes originales.[15]

1.2. Herramientas para la mejora de la calidad de los datos

Existen en el mercado varias herramientas que brindan soporte para la gestión de la calidad de los datos. A continuación se analizan algunas de ellas:

1.2.1. *DQGuru* (antes llamada *Power MatchMaker*)

Es una herramienta de limpieza de datos desarrollada en Java y multiplataforma, que *SQLPower* ha liberado convirtiendo la licencia en código abierto.

La herramienta ayuda a limpiar los datos, validar y corregir las direcciones, identificar y eliminar los duplicados y crear referencias cruzadas entre tablas de origen y de destino. Entre sus características están:

- Verificación de duplicados.
- Combinar duplicados y los datos relacionados.



- Genera tablas de referencias cruzadas para vincular los identificadores de origen del sistema a identificadores de la base de datos de destino.
- Amplio soporte para la transformación con conversión de mayúsculas, concatenación, sustitución de cadenas y sustitución de subcadenas por palabras.[16]

Es una herramienta muy útil para realizar procesos de limpieza, sobre todo si la cantidad de datos a revisar no es muy grande. Enfocada en corregir errores léxicos y eliminar duplicados. No permite definir reglas para un negocio específico, no genera información sobre los cambios hechos, y no opera si no hay mediación humana, la herramienta no tiene una interfaz intuitiva o fácil de entender.

1.2.2. Productos *Data Quality* de Uniserv

Uniserv es una empresa alemana que se encuentra entre los proveedores líderes de soluciones para calidad de datos. Ofrece *software* que puede emplearse internacionalmente así como servicios de aseguramiento de la calidad de los datos en las áreas de Inteligencia de Negocios, *CRM (Customer Relationship Management)*, Almacenes de Datos y Comercio Electrónico.

1.2.2.1. *Data Quality Explorer*

Data Quality Explorer es una herramienta con la que se realizan tareas correspondientes al perfilado de datos. El explorador permite comprobar el estado actual de los datos en situaciones como una migración, antes de una transferencia a otro sistema o antes de iniciar proyectos de gran envergadura relativos a la calidad de los datos. *Data Quality Explorer* sirve de ayuda no solo en la detección de errores, anomalías e inconsistencias en los datos, sino también a mejorar la calidad de los mismos. Se emplea para los análisis de datos financieros, de transacciones, estadísticos o de pedido. Entre sus funciones están:

- Análisis y exploración cooperativos de la calidad de los datos.
- Elaboración de perfiles basados en una perspectiva instantánea del conjunto de los datos captada en un momento dado.
- Documentación, gestión y seguimiento de problemas en la calidad de datos.[17]

1.2.2.2. *Data Quality Monitor*

Data Quality Monitor se encarga de comprobar la calidad de los datos en función de las reglas que se hayan fijado para un negocio, en caso de que se superen ciertos valores umbrales se genera un aviso. Se emplea en el análisis de la calidad de datos financieros, de transacciones, estadístico o de pedido.



Permite la elaboración, gestión y comprobación de reglas relativas a la calidad de los datos: reglas que se guardan en formato de Lenguaje Estructurado de Consulta (*SQL* por sus siglas en inglés) o como procedimientos almacenados. Es posible especificar valores umbrales ligados a las reglas, de forma que al ser superados se genere un aviso. Las reglas se agrupan en tareas que pueden ser ejecutadas de manera automática.

Además provee otras funcionalidades como:

- Aviso por correo electrónico cuando no se cumplan las reglas preestablecidas o se superen ciertos valores límites.
- Elaboración de estadísticas que revelen tendencias en la evolución temporal.
- Gestión de las diferentes versiones temporales de una regla en función de su ventana temporal de vigencia.[18]

1.2.2.3. *Data Quality Batch Suite*

Data Quality Batch Suite es considerada el eje central de todos los proyectos de limpieza de datos. Permite comprobar la calidad de los datos de clientes y en la mayoría de los casos depurarlos de forma completamente automática. Además, con la *Batch Suite* se podrá ejecutar la depuración de grandes cantidades de datos no relacionados con el cliente. Entre sus facilidades están:

- Módulos con funciones para la definición de reglas de control propias, relativas a los datos no relacionados con el cliente y para la ejecución automática de eliminación de grandes cantidades de datos no deseados (por ejemplo, datos financieros, de producto o de pedido).
- Reportes y confección de gráficos de los valores característicos que denotan la calidad.
- Análisis y formateo de direcciones postales.
- Detección de duplicados. [19]

Las soluciones antes mencionadas permiten la comprobación de la calidad mediante el uso de reglas y su agrupación en tareas, están orientadas específicamente a mejorar la calidad de los datos de carácter financiero, estadístico y comercial en las áreas de Inteligencia de Negocios, CRM, Almacenes de Datos y Comercio Electrónico, además son soluciones privadas que solo se pueden usar para base de datos que corren sobre el gestor Oracle.

1.2.3. *Data Cleaner*

Herramienta orientada a preparar los datos para cualquier proyecto en el que se deban aplicar técnicas de calidad de datos, es multiplataforma dado que está desarrollada en Java.



Incluye múltiples funcionalidades tales como:

- Monitoreo de datos: permite gestionar y establecer un seguimiento de la calidad de los datos.
- Limpieza de datos: a través de un conjunto de transformadores y analizadores permite estandarizar los valores utilizando sinónimos, conformar fechas en un formato de fechas único, analizar, extraer y normalizar la información mediante el uso de expresiones regulares o scripts.
- Perfil de datos: proporciona un conjunto de características de perfiles de datos, incluyendo la búsqueda de patrones, que muestra la distribución y frecuencia de los valores, la investigación de juegos de caracteres y trazar líneas de tiempo de las entradas de datos periódicos.[20]
- Permite visualizar mediante el uso de reportes los resultados de las acciones realizadas sobre los datos.

La herramienta permite detectar y borrar duplicados, comparar la información de diferentes fuentes de origen y crear un repositorio de datos maestros y correctos contra los que validar datos y visualizar los resultados en un reporte. A pesar de estas ventajas posee una curva de aprendizaje elevada, permite la construcción de reglas con muy pocos operadores de SQL, lo que limita la construcción de reglas complejas y no está orientada al trabajo con objetos del negocio sino directamente con la estructura de datos.

1.2.4. *Trillium Software System*

Es una solución de calidad de datos empresarial especialmente diseñada para el trabajo en conjunto de las unidades de negocio, para mejorar y gestionar la calidad de la información. A través de una interfaz intuitiva y fácil de navegar, los usuarios pueden cuantificar las condiciones de los datos, desarrollar reglas de negocio, implementar servicios de calidad de los datos y apoyar el gobierno de datos con cuadros de mando e informes que evalúan el cumplimiento de los datos contra las políticas y requerimientos de la organización.

Trillium Software System proporciona una gama completa de productos que ayudan a las empresas a crear y mantener de forma continuada, información de alta calidad a lo largo de todos sus sistemas empresariales tales como:

TS Discovery: ayuda a sus usuarios a entender los dominios informativos, patrones, formatos y relaciones, tal cual existen en la propia información. Además, permite comprobar la adecuación a determinadas reglas de negocio y estándares de datos definidos por la organización.



TS Quality. es el *software* líder mundial de calidad de datos. Ofrece todas las características y las herramientas necesarias para desarrollar procesos de calidad de datos dentro de las empresas. [21]

Trillium Software System posee las siguientes características:

- Conjunto completo de funciones de descubrimiento, generación de perfiles, análisis, normalización, conciliación y control de datos.
- Funciones de fichero amplias que permiten la reutilización de reglas, procesos y flujos de trabajo.
- *GUI* (Interfaz Gráfica de Usuario) basada en metodología para acelerar el desarrollo de calidad de los datos.
- Compatibilidad con los principales idiomas y datos múltiples en codificaciones tales como ASCII, EBCDIC y Unicode.
- Vistas interactivas de resumen, tablas y gráficos.
- Compatibilidad *API* (del inglés *Application Programming Interface*) para las principales normas: C, C++, C#, Java, Soap, .NET, XML.[22]

Entre los clientes más representativos de *Trillium Software System* se encuentran: *Microsoft*, *AT&T*, *British Telecom*, *FedEx*, y el Banco de Crédito de Perú. Esta herramienta es privativa y la plataforma comienza en los \$ 75.000, dependiendo de las funciones seleccionadas.

1.3. Valoración de las herramientas para mejorar la calidad de los datos

Las herramientas antes analizadas aunque brindan soporte para la gestión de calidad de los datos y poseen funcionalidades que sirven de aporte a la investigación, como la comprobación mediante el uso de reglas, la agrupación de reglas en tareas y la visualización de los resultados en reportes, carecen de características y funcionalidades que son importantes para lograr el objetivo de la presente investigación. Algunas de estas herramientas no permiten definir reglas para un negocio específico, son ineficientes para manejar gran volumen de información o están orientadas al trabajo en áreas específicas como la Inteligencia de Negocios y el Comercio Electrónico. *Data Cleaner* está limitada a la construcción de reglas del negocio de baja complejidad, además está orientada al trabajo directo sobre la estructura de datos, lo que puede provocar sobrecarga de la base de datos. Mientras que con el uso de *Trillium Software System* los usuarios pueden cuantificar las condiciones de los datos y desarrollar reglas de negocio, no cumple con el principio de soberanía tecnológica del país, la universidad y específicamente del CENIA.



Los sistemas estudiados aunque permiten de una forma u otra establecer un seguimiento de la calidad de los datos y cuentan con elementos que pueden ser útiles en el desarrollo de la solución, no satisfacen las condiciones necesarias para incluirse totalmente en la propuesta de solución.

1.4. Tecnologías seleccionadas

1.4.1. Lenguajes

Java 6.0

Hasta la fecha, la plataforma Java ha atraído a más de 9 millones de desarrolladores de *software*. Se utiliza en los principales sectores industriales y está presente en una amplia gama de dispositivos, computadoras y redes.

La versatilidad, portabilidad de plataformas y seguridad de la tecnología Java la convierten en ideal para la informática de redes. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.[23]

Algunas de las características de este lenguaje son:

- **Simple:** ofrece todas las funcionalidades de un lenguaje potente, pero sin las características menos usadas y más confusas de estos. Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++.
- **Orientado a objetos:** 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.
- **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.
- **Robusto:** realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.
- **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.



- **Dinámico:** se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales siempre que mantengan la *API* anterior.[24]

XML 1.0

XML (Lenguaje de Marcado Extensible) es más que un lenguaje, es un metalenguaje que permite definir otros lenguajes de marcas con objetivos diferentes. Por ese motivo se le llama extensible. Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes específicos. Aparece en 1997 como un subconjunto de *SGML* (*Structured Generalized Markup Language*, ISO 8879), lenguaje que permite especificar las reglas de etiquetado de documentos. XML permite representar información estructurada (en forma de documentos), de modo que pueda ser almacenada, transmitida, procesada, presentada e impresa por diferentes tipos de aplicaciones y dispositivos.

Las características más importantes de XML son las siguientes:

- Se almacena en formato de texto (no binario) lo cual hace que los documentos sean directamente entendibles. Es decir, los documentos tienen una estructura entendible tanto por los ordenadores como por las personas.
- Utiliza reglas de generación concretas y por tanto, los documentos son fácilmente procesables.
- Permite compartir información entre sistemas o fuentes de datos heterogéneas, por ejemplo, páginas web y distintas bases de datos. Mejora la compatibilidad entre aplicaciones.
- Es un estándar abierto no sujeto a ningún tipo de licencia.[25]

UML 2.0

El Lenguaje Unificado de Modelado (*UML*, por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprenden el desarrollo de *software*. Construido sobre los conceptos fundamentales del paradigma Orientado a Objetos. Entrega una forma de modelar elementos conceptuales como son procesos de negocio y funciones de sistema, además de elementos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de *software* reusables.[26]

Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder información para que los usuarios y desarrolladores comprendan las características de la aplicación, esto



garantiza la comunicación entre desarrolladores y que los clientes entiendan lo que se está realizando sin tener que adentrarse en el lenguaje técnico.

Mejora los tiempos totales de desarrollo (de 50 % o más) con una alta reutilización y minimización de costos.[27]

1.4.2. Herramientas

NetBeans 7.1

Es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento. *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

NetBeans IDE 7.1 es un IDE gratuito de código abierto para Oracle Solaris, Oracle Linux, otras distribuciones de Linux, Mac y Windows que permite a los desarrolladores crear, compilar, depurar y ejecutar rápidamente aplicaciones Web, empresariales, de escritorio y móviles utilizando principalmente Java, así como PHP y C/C++.

NetBeans IDE 7.1 es el primer IDE que admite las últimas especificaciones y estándares de la plataforma Java, incluido el soporte completo de desarrollo de JavaFX 2.0 y JDK7. Además, NetBeans IDE 7.1 ofrece mejoras considerables para crear interfaces gráficas de usuario *Swing* (*Swing GUI Builder*), admite CSS3 y herramientas para la depuración visual de interfaces de usuario (IU), tanto JavaFX como *Swing*.

Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. El código fuente está disponible para su reutilización de acuerdo con la *Common Development and Distribution License* (CDDL) v1.0 y la *GNU General Public License* (GPL) v2.[28]

Visual Paradigm 5.0

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 costo. Permite dibujar todos los tipos de diagramas de clases, realizar ingeniería inversa desde gestores de bases de datos, generación de bases de datos, código en lenguajes como Java o C++ desde diagramas y generar documentación. La herramienta también proporciona abundantes tutoriales de *UML*, demostraciones interactivas de *UML* y proyectos *UML*.[29]



Soporta los principales estándares de la industria tales como *UML*, *SysML* y *BPMN*. Ofrece un completo conjunto de herramientas de equipos de desarrollo de *software* necesario para la captura de requisitos, *software* de planificación, modelado de datos, es multiplataforma y está disponible de forma gratuita.[30]

Evolus Pencil 1.3.4

Desarrollado por la compañía Evolus Co., Ltd Pencil es una herramienta gratuita, de código abierto multiplataforma para hacer diagramas y prototipos de interfaz gráfica, puede ser instalado como un complemento del navegador Firefox. Posee un conjunto de componentes como íconos, cajas de textos y botones. Brinda la capacidad de editar texto enriquecido y trabajar a través de procedimientos estándares de dibujo, tales como alineación, escalado, rotación, dimensión y z-orden. Permite exportar el proyecto de manera directa en un solo archivo, contando con formatos como pdf, html y png.[31]

PostgreSQL-8.4

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia *BSD* (*Berkeley Software Distribution*) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos y Windows. Posee documentación muy bien organizada, pública y libre con comentarios de los propios usuarios.

Soporte originado para los lenguajes más populares del medio: PHP, C, C++, Perl, Python y Java. 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 y vistas materializadas). Altamente adaptable a las necesidades del cliente.

Dispone de procedimientos almacenados en numerosos lenguajes de programación, PL/PgSQL (similar al PL/SQL de Oracle), PL/Perl, PL/Python y PL/Tcl. Numerosos tipos de datos y la posibilidad de definir nuevos tipos. Además de los tipos de estándares en cualquier base de datos, dispone de tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML y matrices. Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido). *APIs* para programar en lenguajes como C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme y Qt.[32]



PgAdmin III 1.10.5

PgAdmin III es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (*EnterpriseDB Postgres, Plus Advanced Server y Greenplum Database*). Incluye:

- Interfaz administrativa gráfica.
- Herramienta de consulta SQL (con un EXPLAIN gráfico).
- Editor de código procedural.
- Agente de planificación SQL/shell/batch.
- Administración de Slony-I.

Diseñado para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris. Desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo y está disponible en más de una docena de idiomas. Es un *software* libre publicado bajo la licencia PostgreSQL.[33]

JUnit 3.8.2

Es un *Framework Open Source* para la automatización de las pruebas unitarias en los proyectos de *software*. Se utiliza para la elaboración de las pruebas de caja blanca, provee al usuario de herramientas, clases y métodos que facilitan la realización de las pruebas, de modo que asegura la consistencia y funcionalidad del sistema. En función de algún valor de entrada se evalúa el valor de retorno esperado, si la clase cumple con la especificación entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.[34]

1.4.3. Proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI

Con el propósito de ganar en competitividad y de asegurar la calidad de sus productos, la universidad decidió adoptar procesos para el desarrollo de aplicaciones informáticas a nivel de organización y después del análisis de varios estándares y modelos se decide utilizar CMMI (Modelo de Capacidad y Madurez Integrado), que es un modelo de calidad para mejorar los procesos de las organizaciones. Actualmente el CENIA lleva a cabo el proceso de desarrollo de *software* con un enfoque ágil basado en el nivel 2 de CMMI, donde su enfoque ágil se basa en la utilización de algunas prácticas de las metodologías Scrum y XP (Programación Extrema). La unión de este modelo de calidad junto con las prácticas garantiza



organización en los proyectos, ya que el modelo indica qué deben hacer los procesos para obtener y entregar productos y servicios de calidad y las prácticas declaran cómo llevarlos a cabo.

1.4.3.1. Ventajas del modelo CMMI

CMMI permite relacionar de forma más explícita las actividades gerenciales y de ingeniería con los objetivos del negocio. Posibilita expandir el alcance y la visibilidad hacia el ciclo de vida del producto y a las actividades de ingeniería, para asegurarse que el producto o servicio cumpla con las expectativas de los clientes. Las buenas prácticas que propone CMMI contribuyen a la disminución del tiempo de desarrollo y recursos invertidos en la corrección de defectos, mayor tolerancia al cambio e incremento de la capacidad de adopción y adaptación de nuevas tecnologías. El modelo tiene dos representaciones: la representación continua y la representación por etapas o niveles. La representación por etapas ofrece un enfoque sistemático y estructurado para mejorar los procesos paso a paso. Al conseguir cada etapa, se asegura que se ha dado un mejoramiento y que se han establecido las bases necesarias para iniciar la siguiente etapa. Cuenta con 5 niveles de madurez de los procesos, dentro de ellos está el nivel 2 o gestionado, compuesto a su vez por las siguientes 7 áreas de procesos (AP):

- Administración de Requisitos.
- Aseguramiento de la Calidad de Procesos y Productos.
- Planeación de Proyectos.
- Monitoreo y Control de Proyectos.
- Medición y Análisis.
- Administración de Configuración.
- Administración de Acuerdos con Proveedores.

Las AP ayudan a la mejora de los objetivos de la producción en la UCI. Basados en este modelo, el programa de mejora que se desarrolla actualmente en la universidad tiene como objetivo la definición e implementación de los procesos necesarios para cubrir estas 7 áreas de procesos, buscando con ello la reducción de varios de los problemas que están vigentes en la producción de *software* en la universidad.[35]

1.4.3.2. Prácticas de Scrum

- Reunión diaria.
- Planificación de la iteración o sprint.



- Revisión de la iteración o sprint.
- Pila del producto.
- Auto-organización.

1.4.3.3. Prácticas de XP

- Cliente in-situ.
- El juego de la planificación.
- Diseño simple.
- Pruebas.
- Estándares de programación.
- Refactorización.
- Propiedad colectiva del código.
- Integración continua.
- Entregas pequeñas.[36]

Conclusiones parciales

Las características de la calidad de los datos analizadas son la evidencia de todos los parámetros que debe tener determinada información para que cumpla con los niveles de calidad definidos por una entidad y para un uso específico. El estudio de las técnicas de evaluación y corrección de datos muestra lo relacionadas que están, pues la evaluación indica la necesidad de tomar acciones correctivas para llevar una organización hacia la calidad de los datos deseada. Con el estudio de los sistemas que actualmente le dan seguimiento a la calidad de los datos se adquirió una serie de conocimientos que son útiles para el desarrollo de la propuesta de solución de la presente investigación, aunque estos no reúnen los requisitos para ser incluidos completamente en la misma. Por otra parte el estudio de las tecnologías seleccionadas por el grupo de arquitectura del CENIA, permitió un acercamiento a los elementos del marco de trabajo y sus características, para hacer más fácil su uso durante el desarrollo de la propuesta de solución planteada.



Capítulo 2. Diseño de la solución.

Introducción

En el proceso de construcción de *software* una de las partes más difíciles del trabajo conceptual es decidir precisamente qué construir, de modo que se logre entender las necesidades del cliente y los demás involucrados. En el presente capítulo se caracteriza la solución propuesta, se describen las clases del dominio en cuestión y se especifican los requerimientos funcionales y no funcionales identificados a partir de la aplicación de técnicas de obtención de requisitos. Además se definen los patrones de diseño, de base de datos y el estilo arquitectónico a utilizar.

2.1. Modelo del dominio

Es el mecanismo fundamental para comprender el dominio del problema. Se utilizó para capturar y expresar conceptos u objetos del mundo real del dominio analizado, sus características y las relaciones entre dichos conceptos. El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema.[37]

A continuación se muestra el modelo de dominio que representa el seguimiento de la calidad en el SGU:

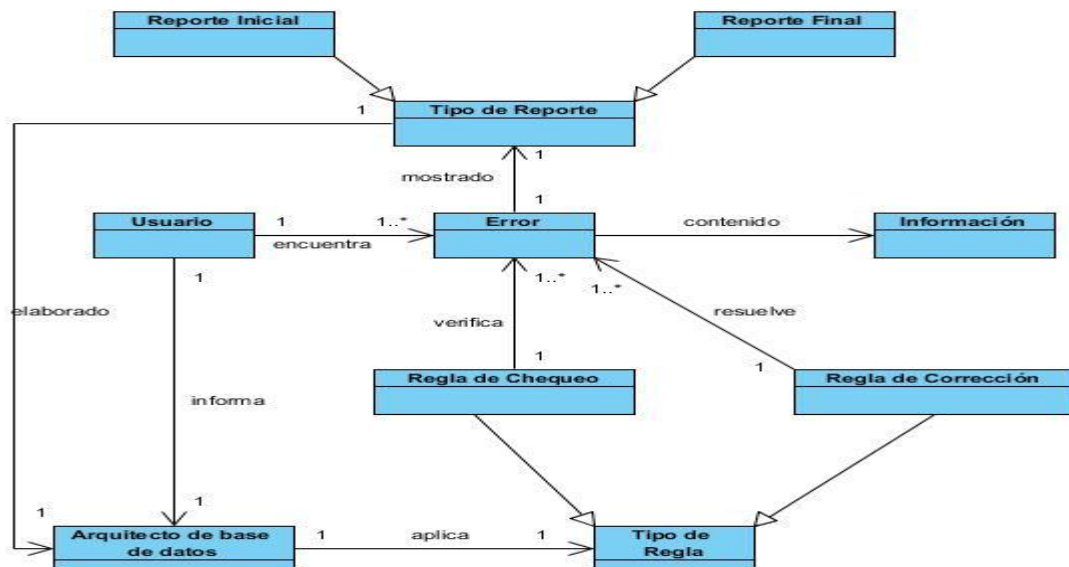


Figura 1: Modelo del dominio.



2.1.1. Descripción de las clases del dominio

Usuario: representa al usuario que detecta errores en la información que utiliza cuando interactúa con el SGU.

Error: representa la afectación de determinadas dimensiones de la calidad de los datos consultados por el usuario.

Información: representa la información a la que tiene acceso determinado usuario.

Arquitecto de base de datos: representa al encargado de identificar las causas del error detectado por un usuario y corregirlo. Además es el encargado de elaborar los reportes.

Regla de Chequeo: representa la regla que se aplica para verificar el error o los errores encontrados por un usuario.

Regla de Corrección: representa la regla que se aplica para darle solución a los errores detectados.

Reporte Inicial: permite mostrar la información con los errores encontrados.

Reporte Final: permite mostrar la información una vez corregidos los errores.

2.2. Presentación de la propuesta de solución

Por la situación existente en el SGU en cuanto a la afectación de determinadas dimensiones de la calidad de los datos almacenados, se decide la creación de la herramienta Policía de datos del SGU, que permitirá realizar acciones (seleccionar, eliminar, actualizar, insertar y transformar) sobre una fuente de datos o sobre un destino a partir de una fuente. La fuente y el destino estarán representados por objetos del negocio, los cuales podrán agrupar entidades que en la base de datos están separadas producto de la normalización. Los objetos brindan una mayor abstracción al usuario, permitiéndole que se centre en el negocio que maneja. El usuario podrá darle nombre a un objeto, definir sus propiedades (campos y tablas) y la forma en que se relacionan las tablas que representa a partir de la estructura de la base de datos que se analiza, esto conformará la configuración del objeto, la que permitirá llegar a los componentes físicos en la base de datos. A partir de los objetos configurados el usuario podrá construir reglas que pueden ser del negocio, de validación o de rectificación. Las reglas del negocio serán declaraciones que restringirán diferentes aspectos del negocio, responderán a flujos bien definidos, y se ejecutarán sobre la información. Las reglas de validación estarán orientadas a mejorar la exactitud de los datos almacenados y a tratar errores léxicos. Las reglas de rectificación estarán asociadas únicamente a la acción de transformación y su objetivo será verticalizar los datos de un objeto del negocio, lo cual consiste en verificar si un conjunto



de datos son atributos de determinada tabla representada por un objeto, de no ser así estos datos pasan a ser nuevos atributos de dicha tabla. Las reglas pueden agruparse en tareas programadas y están asociadas a acciones que podrán ejecutarse cuando un usuario lo requiera, o de manera automática. Los resultados de las acciones son mostrados en reportes, que contendrán además otros datos como la fecha en que se ejecutó la acción y la información antes y después de ejecutar la acción.

El siguiente mapa conceptual muestra los conceptos claves de la propuesta de solución:



Figura 2: Mapa conceptual.

2.3. Técnicas de obtención de requisitos

La Ingeniería de Requisitos (IR) es el área del conocimiento que explica cómo lograr la obtención, análisis, especificación y validación de los requisitos del *software*. Tiene como objetivo proporcionar el mecanismo adecuado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable y especificarla sin ambigüedades, así como evaluarla conforme esta se transforma en un sistema operacional.[38]

La IR ofrece técnicas para identificar y extraer los requisitos, dentro de las que se encuentran: entrevistas, cuestionarios, tormentas de ideas, desarrollo conjunto de aplicaciones (JAD), escenarios, prototipos, reuniones de facilitación o taller de requisitos y observación directa (Etnografía).



De las técnicas antes mencionadas para la captura de requisitos se utilizan las siguientes:

Tormenta de ideas: es una técnica de reuniones en grupo que permite generar ideas originales en un ambiente libre de críticas. Ayudó a generar una gran variedad de vistas del problema y a formularlo de diferentes formas al inicio del proceso de captura, cuando los requisitos son todavía muy difusos.[39]

Observación directa: es una técnica de observación que se puede utilizar para entender los requerimientos sociales y organizacionales. Se centra en los siguientes aspectos: la forma en la que las personas trabajan y no en como el sistema los hace trabajar. Mediante esta técnica se tuvieron en cuenta las características de los usuarios que van a interactuar con el sistema para hacerlo más cómodo y fácil.[40]

Prototipos: permite obtener una pequeña muestra de cómo sería el producto final una vez terminado. Ayudó a conocer la opinión de los usuarios y a rectificar algunos aspectos antes de llegar al producto final. Permite identificar elementos que falten dentro de los requisitos.[40]

JAD: se utilizó para promover la cooperación y el trabajo en equipo entre usuarios y analistas. Consiste en realizar sesiones en las que participan usuarios expertos del dominio junto a analistas de *software* para aprovechar la dinámica de grupos, apoyadas por elementos visuales de comunicación y comprensión de soluciones.[40]

2.4. Definición de los requerimientos funcionales y no funcionales

Se define un requisito como un atributo necesario en el sistema, una declaración que identifica una capacidad, característica o factor de calidad de un sistema con el fin de tener valor y utilidad para un cliente o usuario; con lo que se pretende cumplir determinadas necesidades o restricciones operativas y que contribuyen a solucionar un problema en un entorno real.[41]

2.4.1. Requisitos funcionales

Declaran los servicios que el sistema debe proporcionar. Dependen del tipo de *software* que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización al redactar los requerimientos. Describen la interacción entre el sistema y su ambiente independientemente de su implementación. El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema.[42]

A continuación se muestran los requerimientos funcionales definidos para la propuesta de solución con sus prioridades respectivas:



Tabla 2. Requisitos funcionales.

No	Descripción	Prioridad
RF1	Mostrar configuraciones	Media
RF2	Crear configuración	Alta
RF3	Modificar configuración	Alta
RF4	Eliminar configuración	Media
RF5	Mostrar objetos del negocio	Media
RF6	Crear objeto del negocio	Alta
RF7	Modificar objeto del negocio	Alta
RF8	Ver detalles de objeto del negocio	Baja
RF9	Eliminar objeto del negocio	Media
RF10	Eliminar propiedad	Baja
RF11	Eliminar unión de tablas	Baja
RF12	Mostrar reglas	Media
RF13	Crear regla	Alta
RF14	Modificar regla	Alta
RF15	Eliminar regla	Media
RF16	Ejecutar regla	Alta
RF17	Mostrar tareas	Media
RF18	Crear tarea	Alta



RF19	Modificar tarea	Alta
RF20	Eliminar tarea	Media
RF21	Ver detalles de tarea	Baja
RF22	Mostrar reportes	Media
RF23	Ver detalles de reporte inicial	Baja
RF24	Ver detalles de reporte final	Baja

2.4.2. Especificación de requisitos funcionales

Permite expresar detalladamente las funcionalidades de un *software*, de modo que puedan ser revisados y aceptados por todos los involucrados. El objetivo de la especificación de requerimientos es documentar toda la información relativa al dominio de la aplicación teniendo en cuenta el alcance del proyecto y el receptor de la información. Es el producto del trabajo final que genera la IR.[43]

A continuación se muestra uno de los requerimientos especificados, para consultar las restantes especificaciones remitirse a los Anexos, Tabla (9 - 31).

Tabla 3. Especificación de requisito. Crear tarea.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF18.]	Crear tarea.	Permite crear una tarea. Se muestran los campos a llenar y una vez creada la tarea se muestra un mensaje de información "El elemento ha sido creado satisfactoriamente".	Alta	Alta
Prototipo				



Prototipo Crear tarea.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre	character varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.
Fecha de ejecución	character varying	Permite seleccionar la fecha para ejecutar la tarea. Campo obligatorio.
Hora	character varying	Permite seleccionar la hora en que se desea ejecutar la tarea. Campo obligatorio.
Minutos	character varying	Permite especificar los minutos exactos en que se va a ejecutar la tarea. Campo obligatorio.
AM	booleano	Permite seleccionar la sección del día AM, para ejecutar la tarea.
PM	booleano	Permite seleccionar la sección del día PM, para ejecutar la tarea.
Reglas	character varying	Permite asociar un grupo de reglas a la tarea que se está creando. Campo obligatorio.



Observaciones	<ol style="list-style-type: none">1. En caso que la tarea ya exista se muestra el mensaje de error: <i>“El elemento ya existe”</i>.2. En caso de cancelar la acción se muestra el mensaje de advertencia: <i>“¿Está seguro de realizar la acción?”</i>.3. En caso de que se introduzcan datos incorrectos (solo se admiten letras y guión bajo) se muestra el mensaje de color rojo: <i>“Entre letras y guión bajo”</i> encima del nombre del campo indicando el tipo de error.4. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: <i>“Campo requerido”</i> encima del nombre del campo que debe ser llenado obligatorio.5. Si se trata de introducir más de los caracteres permitidos para los campos de texto el sistema no permite seguir introduciéndolos.6. Interactúa con esta acción el administrador de base de datos.
----------------------	---

2.4.3. Requisitos no funcionales

No se refieren directamente a las funciones específicas que debe expresar el sistema, sino a restricciones del sistema y del proceso de desarrollo que se debe utilizar; las restricciones pueden ser de distintos tipos, por ejemplo relacionadas con propiedades del sistema como seguridad, fiabilidad y usabilidad. O relacionadas con cuestiones organizacionales como uso de tecnologías o estándares específicos.[44] A continuación se muestran las restricciones que debe cumplir el sistema:

Tabla 4. Requisitos no funcionales.

Requisitos no funcionales (RNF)	
Usabilidad	
RNF1	La herramienta estará dirigida a arquitectos de base de datos con un nivel medio o superior de informática.
RNF2	La herramienta debe ser soportada en plataformas Linux y Windows.
RNF3	Debe estar instalada la máquina virtual de Java JRE 6.0 o superior.



RNF4	La máquina cliente debe tener mínimo 512 MB de memoria RAM.
Interfaz	
RNF5	La herramienta debe contar con una interfaz amigable, con colores suaves, donde el usuario pueda orientarse fácilmente.
RNF6	La comunicación entre el cliente y el servidor de base de datos debe ser a través del protocolo TCP/IP.
RNF7	La comunicación entre la computadora de chequeo y el servidor de base de datos debe ser a través del protocolo TCP/IP.
Restricciones del diseño	
RNF8	Lenguaje de programación Java.
RNF9	NetBeans 7.1 como Entorno de desarrollo integrado.
RNF10	PostgreSQL 8.4 como Sistema Gestor de Base de Datos.
Seguridad	
RNF11	La herramienta almacenará las contraseñas de las bases de datos que maneja de manera encriptada.

2.5. Descripción de la arquitectura y el diseño

2.5.1. Estilo arquitectónico

La arquitectura Tres Capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva del problema a resolver. Los roles indican el tipo y la forma de interacción con otras capas y las responsabilidades la funcionalidad que implementan. Descompone los servicios de forma que la mayoría de las interacciones ocurran solo entre capas vecinas. Cada nivel agrega las responsabilidades y abstracciones del nivel inferior. Separa de forma clara la funcionalidad de cada capa y las capas inferiores no tienen dependencias de las capas superiores.

La siguiente figura muestra una representación de la arquitectura Tres Capas:

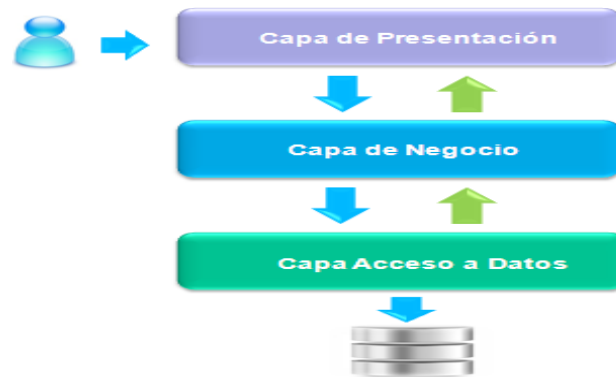


Figura 3: Arquitectura Tres Capas.

Capas

- **Capa de Presentación:** es la parte que ve el usuario, donde se le muestran las interfaces para que interactúe con el programa comunicándole y recolectando la información suministrada por el usuario en un mínimo de proceso. Se comunica únicamente con la capa de negocio llevando y trayendo los datos o registros necesarios; es la interfaz gráfica del programa y debe ser lo más amena posible para una mejor comunicación con el usuario.
- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es donde se establecen todos los procesos que deben realizarse. Es una capa intermedia, entre la capa de presentación y la capa de datos, puesto que se relaciona con ambas y procesa la información devuelta por la capa de datos.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.[45]

2.5.2. Modelo de despliegue

Describe la distribución física del sistema entre los nodos de computación sobre los que se va a instalar el sistema. Se utiliza un nodo para identificar cualquier servidor, terminal de trabajo u otro *hardware* que se utiliza para desplegar componentes en el ambiente de producción.[46]



El diagrama de despliegue del Policía de datos del SGU mostrado en la siguiente figura está compuesto por una computadora cliente que se conecta al servidor de base de datos a través de la familia de protocolos TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet) y una computadora de chequeo que se conecta al servidor de base de datos también mediante el protocolo TCP/IP:

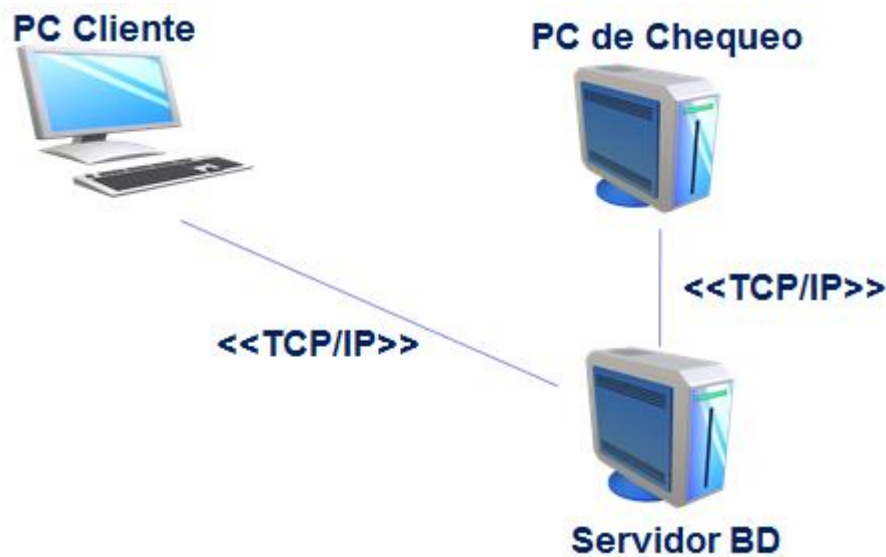


Figura 4: Diagrama de despliegue.

Descripción de los elementos del modelo de despliegue

- **Computadora Cliente:** su función es acceder a la herramienta e interactuar con la misma según sus necesidades. Al estar la aplicación desarrollada sobre Java, la máquina cliente necesita disponer de muy pocas prestaciones puesto que solo requiere de la máquina virtual de java para levantar la aplicación y realizar las operaciones necesarias.
- **Servidor de Base de Datos:** es el encargado de proveer la información para crear las configuraciones de los objetos del negocio y de almacenar la información generada en la creación de reglas, tareas y reportes. Debe estar corriendo sobre PostgreSQL 8.4.
- **Computadora de Chequeo:** su función es ejecutar una aplicación como un servicio del sistema operativo, el mismo ejecutará las tareas programadas en la máquina cliente sobre el servidor de base de datos.



- **<<TCP/IP>>**: La familia de protocolos de Internet es un conjunto de protocolos de red en la que se basa Internet y que permite la transmisión de datos entre redes de computadoras.

2.5.3. Patrones de diseño

Se aplican a un elemento específico del diseño como un agregado de componentes para resolver algún problema de diseño, relaciones entre los componentes o los mecanismos para efectuar la comunicación de componente a componente.[47]

2.5.3.1. Patrones para asignar responsabilidades (GRASP)

Experto: el objetivo es asignar una responsabilidad al experto en información, que es la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Se evidencia en las clases modelos que son las que cuentan con los métodos necesarios para realizar tareas sobre la base de datos.

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Su propósito fundamental es encontrar un creador que se conecte con el objeto producido en cualquier evento.

Se evidencia en la clase *CRegla* que es la encargada de crear una instancia de la clase *MRegla*.

Alta Cohesión: asignar una responsabilidad de modo que la cohesión siga siendo alta. Caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Se evidencia en la clase *Serialize* que cuenta únicamente con las funcionalidades necesarias para serializar y deserializar objetos.

Bajo Acoplamiento: asignar una responsabilidad para mantener bajo acoplamiento (una clase con bajo acoplamiento no depende de muchas otras). Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño.

Está presente en la clase *Xml* que no depende de otras para su funcionamiento.

Controlador: asignar la responsabilidad del manejo de mensajes de los eventos de un sistema a una clase que represente el sistema global, una empresa u organización o el papel de una persona.

Este patrón se ve en las clases controladoras, estas son las que se encargan de obtener datos y brindárselos a las vistas.[48]



2.5.3.2. Patrones GOF

Existen tres tipos de estos patrones: de creación, estructurales y de comportamiento.

Los patrones **de creación** abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas.

- **Instancia única:** garantiza que únicamente se cree una instancia de una clase y provee un punto de acceso global a ella. Todos los objetos que utilizan una instancia de esa clase usan la misma instancia.[49]

Este patrón se evidencia en la clase `Xml`, la cual es utilizada por varias clases y para evitar que sea creada en todas las clases que tienen dependencias de ella se crea una instancia de la clase y se provee un punto de acceso global a la misma, en este caso el método `GetXml()`, además para evitar que se creen otras instancias su constructor es privado.

```
static private Xml xm1 = null;
private Xml() {
}
static public Xml getXm1() {
    if (xm1 == null) {
        xm1 = new Xml();
    }
    return xm1;
}
```

Figura 5: Ejemplo del uso del patrón Instancia única.

Los patrones **estructurales** describen cómo las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

- **Iterador:** define una interfaz que declara métodos para acceder secuencialmente a los objetos de una colección. Una clase que accede a una colección solamente a través de una interfaz independiente de la clase que implementa la interfaz.[49]

El uso del patrón se evidencia en el uso de la *API* de java (`java.util.Iterator`), que provee los métodos necesarios para iterar sobre una colección sin que esta tenga la necesidad de exponer su representación interna.



```
public Element BuscarObjeto(String objeto, String dir) throws JDOMException, IOException {
    Element object = null;
    Document doc = AbrirXML(dir);
    Element raiz = doc.getRootElement();
    List<Element> hijosraiz = raiz.getChildren();
    boolean encontrado = false;
    Iterator iterador = hijosraiz.iterator();
    while (iterador.hasNext() && !encontrado) {
        Element obj = (Element) iterador.next();
        if (obj.getName().equals(objeto)) {
            object = obj;
            encontrado = true;
        }
    }
    return object;
}
```

Figura 6: Ejemplo del uso del patrón Iterador.

Los patrones **de comportamiento** estudian las relaciones de llamadas entre los diferentes objetos normalmente ligados con la dimensión temporal.

Mediador: define un objeto que encapsula la forma en la cual interactúan otros. Utiliza la lógica en un objeto para manejar cambios de estado de otros objetos en lugar de distribuir la lógica sobre los otros objetos, dando como resultado una implementación más cohesiva de la lógica y decrementando las relaciones entre otros objetos. Promueve el acoplamiento débil.[49]

Este patrón se ve en las controladoras que funcionan como mediadoras entre las clases de las vistas y los modelos o acceso a datos.

2.5.4. Patrones de Base de datos

- **Árbol fuertemente codificado:** a cada nodo se le asocia una entidad. En términos de base de datos las relaciones que existen entre las tablas pueden ser de uno a uno, de uno a muchos, de muchos a muchos etc., en este caso las relaciones son de uno a muchos (n). El árbol fuertemente codificado es utilizado para representar jerarquías donde es bien conocida la estructura.[50]

La utilización del patrón está presente en la base de datos de la herramienta en la siguiente situación: la tabla *tb_nestado*, representa los estados que puede tomar una tarea (almacenadas en la tabla *tb_dtarea*) y un estado puede estar asociado a varias tareas, una tarea puede estar asociada a varios reportes (almacenados en *tb_dreporte*).



Figura 7: Ejemplo del patrón Árbol fuertemente codificado.

- **Patrón Llave subrogada:** se decide generar una llave primaria única para cada entidad en vez de usar un atributo identificador en el contexto dado. Normalmente se usa enteros en columnas de identidad o *GUID (Globally Unique Identifier)* que están demostrados que no se repiten o con una probabilidad extremadamente baja. Permite que las tablas sean más fáciles de consultar por el identificador dado que se conoce el mismo tipo de todos en cada tabla.[50]

La utilización del patrón está presente en la base de datos en varias tablas, por ejemplo en la tabla *tb_dtarea*, en vez de usar el campo *nombre_tarea* como llave se decide utilizar un campo numérico auto incrementable llamado *id_tarea*:

sq_policia.tb_dtarea	
id_tarea	int4
nombre_tarea	varchar(2147483647)
fecha_ejecucion	timestamp(29)
id_estado	int4

Figura 8: Ejemplo del patrón Llaves subrogadas.

2.6. Estándares de diseño

2.6.1. Diagramación de las vistas

- Vista de presentación: es la primera vista que se le muestra al usuario que va a interactuar con la herramienta, donde podrá acceder al servicio que desee y estará dividida en las 2 áreas que se enumeran a continuación:
 - 1- Área de menú.



2- Logotipo de la aplicación.

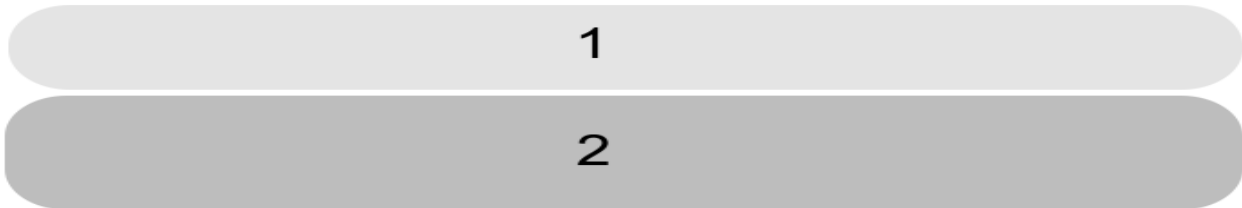


Figura 9: Áreas de la vista de presentación.

- Vista de gestión: se muestra luego de que el usuario selecciona un servicio y permite seleccionar una acción a realizar sobre el servicio, estará dividida en las 4 áreas siguientes:

- 1- Área de título.
- 2- Área de acciones externas.
- 3- Área de contenido.
- 4- Área de acciones de contexto.

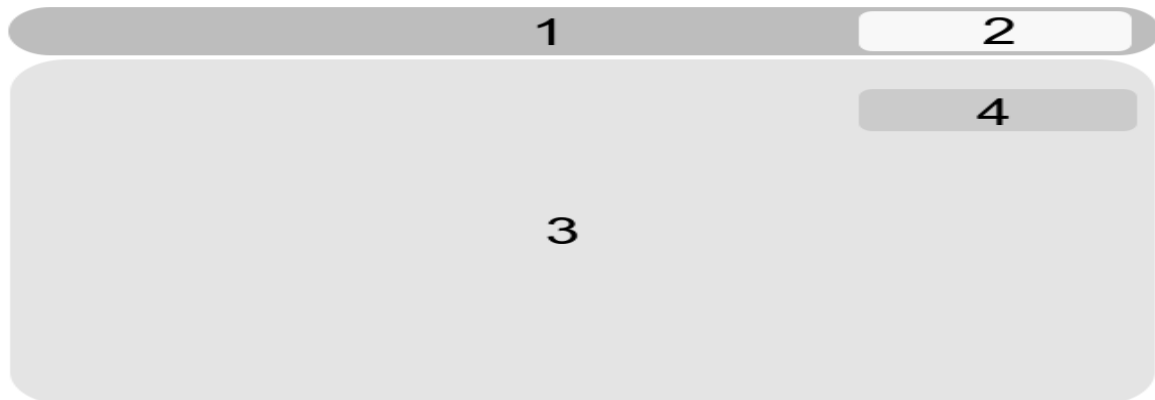


Figura 10: Áreas de la vista de gestión.

- Vista de funcionalidades: se muestra cuando el usuario selecciona una acción de contexto en la vista de gestión y permite realizar la acción seleccionada, estará dividida en las 3 áreas siguientes:
 - 1- Área de título.
 - 2- Área de acciones externas.
 - 3- Área de gestión de acciones.

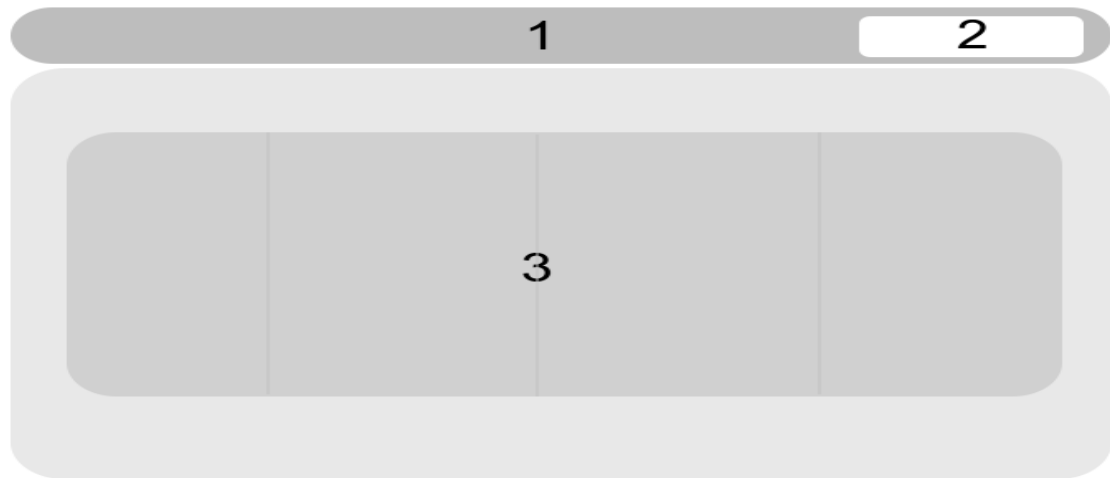


Figura 11: Áreas de la vista de funcionalidades.

2.6.2. Tipos de mensajes

Dentro de los tres tipos de mensajes definidos se encuentran: mensajes de información, mensajes de error y mensajes de confirmación. A continuación se describen cada uno de ellos:

Mensajes de información: se utilizan para brindar una información al usuario y se muestran cuando se crea, se modifica, se elimina o se ejecuta un elemento.

- El elemento ha sido creado satisfactoriamente.
- El elemento ha sido modificado satisfactoriamente.
- El elemento ha sido eliminado satisfactoriamente.
- El elemento ha sido ejecutado satisfactoriamente.

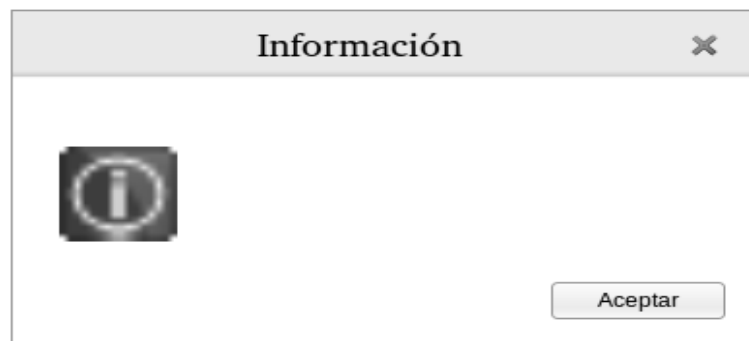


Figura 12: Mensaje de información.



Mensajes de error: se utilizan para mostrarle al usuario que ha realizado una acción incorrecta o que ha ocurrido un fallo en el sistema. Estos mensajes pueden mostrarse encima del componente o en forma de ventana. Dentro de los mensajes se encuentran:

En forma de ventana:

- El elemento ya existe.
- Error desconocido.
- Error de base de datos.
- Los parámetros de conexión no son válidos.
- Origen de datos no válido.

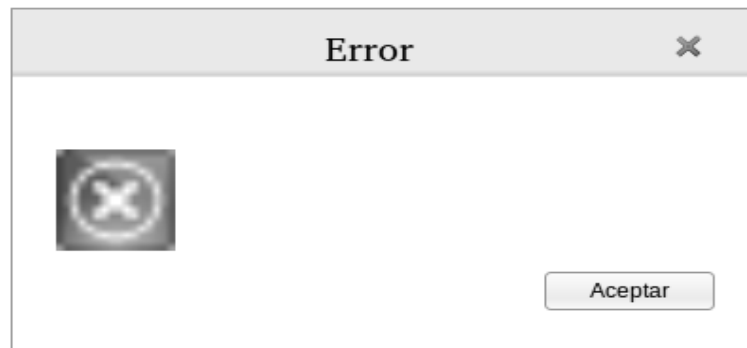


Figura 13: Mensaje de error en forma de ventana.

Encima del componente:

- Campo requerido.
- Entre letras y guión bajo.
- Entre solo letras, números y guión bajo.
- No más de 30 caracteres.
- El directorio no existe.

Mensajes de confirmación: son los que se utilizan cuando es necesario asegurarse que el usuario desea realizar una acción deseada, por ejemplo cuando se va a cancelar una acción, es necesario asegurarse que eso es lo que desea. Dentro de los mensajes están:

- ¿Está seguro de realizar la acción?



- ¿Está seguro que desea salir?

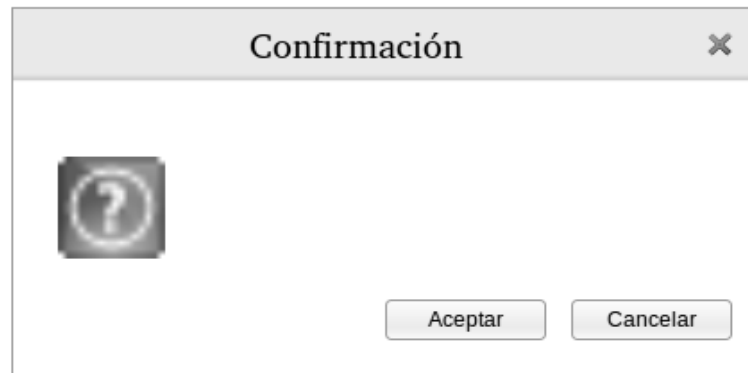


Figura 14: Mensaje de confirmación.

2.7. Mapa taxonómico

El mapa taxonómico permite clasificar la información de manera que se puede lograr una distribución jerárquica de los conceptos fundamentales. Haciendo uso de la herramienta para mapas taxonómicos *MindManager* se distribuyeron las funcionalidades de la aplicación de la siguiente manera:

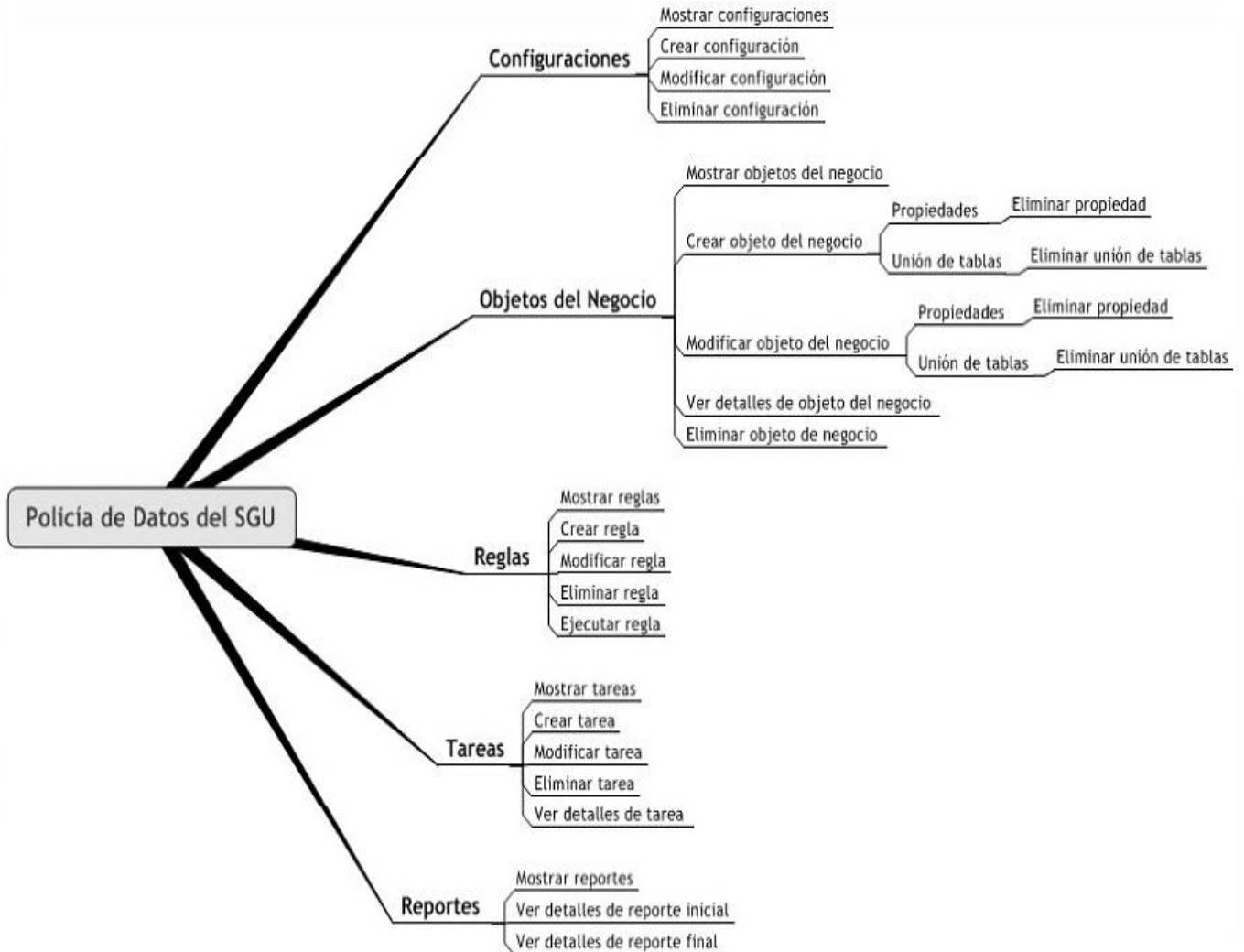


Figura 15: Mapa taxonómico.

2.8. Modelo de datos

El diseño de una base de datos se descompone en tres etapas: diseño conceptual, lógico y físico. La etapa del diseño lógico tiene como objetivo transformar el esquema conceptual en un modelo de datos para un sistema de gestión de bases de datos determinado. La salida de esta etapa es el esquema lógico global y la documentación que lo describe. Todo ello es la entrada para la etapa que viene a continuación, el diseño físico. Mientras que en el diseño lógico se especifica qué se guarda, el diseño físico especifica



cómo se guarda. Incluye escoger las estructuras de archivos determinadas donde almacenar las tablas o relaciones de la base de datos y asegurar el acceso a las relaciones de forma rápida y segura.[51]

En la Figura 16 se muestra el modelo físico de la base de datos de la aplicación. Para ver el modelo lógico dirigirse a los Anexos, Figura 28:

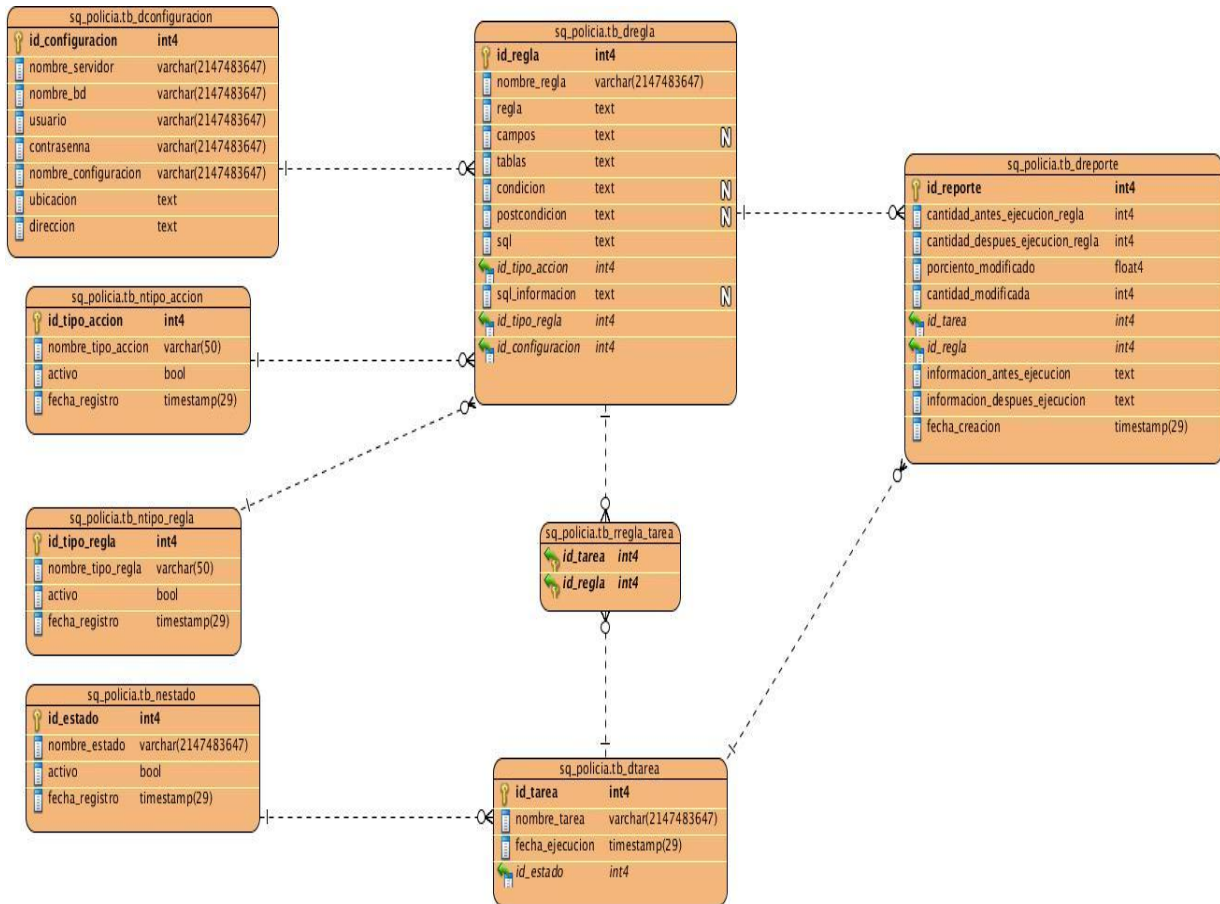


Figura 16: Modelo físico de datos.

Conclusiones parciales

La representación del modelo del dominio sirvió para comprender el funcionamiento del proceso de seguimiento de la calidad de los datos en el SGU. Las técnicas de captura de requisitos utilizadas facilitaron la definición y comprensión de las funcionalidades del sistema, lo que permitió realizar el levantamiento de los requisitos funcionales y no funcionales en su totalidad. A través de las



especificaciones de los requisitos se documentó toda la información relativa al dominio de la aplicación, garantizando la revisión y aceptación por todos los involucrados. La descripción de los estándares de la arquitectura y el diseño permite que se cuente con los elementos necesarios para la implementación del Policía de datos del SGU.



Capítulo 3. Implementación y prueba.

Introducción

En el desarrollo de un sistema informático después de documentar toda la información relativa al dominio de la aplicación mediante las especificaciones de requisitos y luego de describir los estándares fundamentales de la arquitectura y el diseño comienza la implementación del sistema, para lo cual los programadores se apoyan en los artefactos generados de las actividades anteriores. Una vez culminado el proceso de implementación se procede a aplicar un conjunto de pruebas que permiten validar la eficiencia y el rendimiento de la aplicación informática.

En el presente capítulo se describen los estándares de código fundamentales usados en la implementación de la herramienta y las validaciones realizadas a los requisitos antes de su implementación, así como las pruebas a la aplicación una vez terminada para verificar su correcto funcionamiento.

3.1. Estándares de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que otros programadores puedan identificar las variables, las funciones o métodos. Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto para un lenguaje de programación permite que todos los participantes lo puedan entender en menos tiempo y que cualquier persona que se desempeñe como codificador de dicho lenguaje pueda interpretar de manera eficiente la escritura del código.[52]

3.1.1. Identación, llaves de apertura y cierre, tamaño de líneas

Se deben emplear cuatro espacios como unidad de indentación y los tabuladores deben ser exactamente cada 8 espacios. La llave de apertura « { » aparece al final de la misma línea de la sentencia de declaración. La llave de cierre « } » empieza una nueva línea indentada ajustada a su sentencia de apertura correspondiente, excepto cuando no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura « } ». No usar más de 80 caracteres por línea. En la Figura 17 se puede observar un ejemplo.



```
343     public void ModificarAtributo(String nombreOld, String nombreNew,  
344                                 String atributo, String id, String dir)  
345     {  
346         //...  
347     }
```

Figura 17: Ejemplo de indentación, llaves de apertura y cierre.

3.1.2. Conversión de nomenclaturas

Variables

La declaración de las variables se rige por la nomenclatura *CamelCase* en su variante *LowerCamelCase*, donde la primera letra de cada una de las palabras es minúscula. Las variables locales se inicializan donde se declaran. La única razón para no inicializar una variable donde se declara es si el valor inicial depende de cálculos posteriores. En la Figura 18 se puede encontrar un ejemplo:

```
String nombreConfiguracion;
```

Figura 18: Ejemplo de variables.

Clases

Los nombres de las clases deben ser sustantivos y su declaración se rige por la nomenclatura *CamelCase* en su variante *UpperCamelCase*, donde la primera letra de cada una de las palabras es mayúscula. En la Figura 19 se puede encontrar un ejemplo:

```
public class ReglaVisual {  
}
```

Figura 19: Ejemplo de una clase.

Funciones

Los métodos deben ser verbos seguidos de paréntesis, no se deja ningún espacio en blanco entre el nombre de un método y los paréntesis. Los parámetros de las funciones se separan por coma y después de cada coma se pone un espacio. La declaración de las funciones se rige por la nomenclatura *CamelCase* en su variante *UpperCamelCase*. En la Figura 20 se puede encontrar un ejemplo:



```
public Document CrearXML(String direccion) throws IOException {
    Document docNuevo = new Document();
    Format format = Format.getRawFormat();
    XMLOutputter xmlOutputter = new XMLOutputter(format);
    File archivo = new File(direccion);
    xmlOutputter.output(docNuevo, new FileOutputStream(archivo));
    return docNuevo;
}
```

Figura 20: Ejemplo de una función.

3.1.3. Sentencias de control

Las sentencias de control usan siempre llaves « {} ». Entre las sentencias de control y el paréntesis inicial siempre debe existir un espacio, de igual manera entre el paréntesis de cierre y la llave de apertura.[53]

En las Figuras (21-26) se muestran ejemplos de las sentencias de control:

Sentencias *if*

```
if (elemento.getTipo() == TipoElemento.FuncionAgregada) {
    existe = true;
}
```

Figura 21: Ejemplo de una sentencia *if*.

Sentencias *if-else*

```
if (objeto.getChild("tabla") == null) {
    tables = new Element("tabla");
    objeto.addContent(tables);
} else {
    tables = objeto.getChild("tabla");
}
```

Figura 22: Ejemplo de una sentencia *if else*.

Sentencias *if else-if else*

```
if (elemento.getTipo() == TipoElemento.0) {
    cond += " OR ";
} else if (elemento.getTipo() == TipoElemento.Y) {
    cond += " AND ";
} else {
    cond += "$$" + elemento.getValor() + "$$";
}
```

Figura 23: Ejemplo de una sentencia *if else if else*.



Sentencias *for*

```
for (int i = 0; i < configuraciones.length; i++) {  
    modelo.addElement(configuraciones[i][0]);  
}
```

Figura 24: Ejemplo de una sentencia *for*.

Sentencias *while*

```
while (iter.hasNext() && !encontrado) {  
    Element atributo = (Element) iter.next();  
    if (atributo.getAttributeValue("id").equals(id)) {  
        iter.remove();  
        encontrado = true;  
    }  
}
```

Figura 25: Ejemplo de una sentencia *while*.

Sentencias *try catch*

```
if (jComboBox1.getSelectedIndex() != 0) {  
    try {  
        CargarObjetos();  
    } catch (JDOMException ex) {  
        JOptionPane.showMessageDialog(this, Excepciones.getExcepcion().Mensaje(TipoExcepcion.XML),  
            "Error", JOptionPane.ERROR_MESSAGE);  
    } catch (IOException ex) {  
        JOptionPane.showMessageDialog(this, Excepciones.getExcepcion().Mensaje(TipoExcepcion.IO),  
            "Error", JOptionPane.ERROR_MESSAGE);  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(this, Excepciones.getExcepcion().Mensaje(TipoExcepcion.Desconocido),  
            "Error", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Figura 26: Ejemplo de una sentencia *try catch*.

3.2. Técnicas de programación

Una técnica de programación es una metodología que se debe seguir y tomar en cuenta a la hora de programar. Son normas que se deben asumir para lograr la estandarización de la programación, lo que implica una disminución de costos, independencia del programador y seguridad.[54]

A continuación se explica la técnica de programación utilizada para el desarrollo de la aplicación:

Programación orientada a objetos (POO)



La POO es un paradigma que utiliza objetos como elementos fundamentales en la construcción de la solución, se trata de una técnica que aumenta considerablemente la velocidad de desarrollo de los programas gracias a la reutilización de los objetos.

La POO cuenta con mecanismos relacionados entre sí como la herencia, abstracción, polimorfismo, acoplamiento, encapsulamiento y cohesión. La herencia organiza y facilita el polimorfismo y el encapsulamiento, sirve para crear objetos que incorporen propiedades y métodos de otros objetos, lo que permite construir objetos a partir de otros sin tener que reescribirlo todo. La abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real, mediante ella se puede formar un conjunto de clases que permitan modelar la realidad o el problema que se quiere resolver. El polimorfismo garantiza la abstracción para definir un código que sea compatible con objetos de varios tipos. El encapsulamiento suele usarse en conjunto con la ocultación y significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad al mismo nivel de abstracción, esto permite aumentar la cohesión de los componentes del sistema.[55]

3.3. Tratamiento de errores

La presencia de errores durante el desarrollo de *software* es bastante común, pero lo importante no es no equivocarse sino poder detectar los errores a tiempo, debido a que cuanto más tarde se descubran más costosos será corregirlos. El tratamiento de errores debe estar enfocado tanto a los errores que se producen a la hora de los usuarios introducir datos, como a los errores que puedan ser generados por el comportamiento incorrecto de los componentes internos.

En el Policía de datos del SGU el seguimiento que se le da a los datos introducidos por los usuarios es mediante validaciones que se les asignan a los campos de entrada de datos, garantizando que se muestre el tipo de error sobre el campo donde se introdujeron datos incorrectos. Para darle seguimiento a los errores internos de la aplicación hay que tener en cuenta el lenguaje de programación.

En Java una excepción señala un error y es generalmente supuesta porque los errores en este lenguaje de programación están limitados a ser más bien una regla por definición. Una ventaja de darle seguimiento a los errores es poder señalar y separar el tratamiento de los errores mediante excepciones y poder cumplir una respuesta a errores particulares. La gestión de excepciones de Java lleva la gestión del



error en tiempo de ejecución al mundo orientado a objetos. Una excepción de Java es un objeto que describe una condición excepcional que se ha producido en un fragmento de código.

En java el mecanismo para la gestión de excepciones consiste en el uso de bloques *try/catch*, para lo cual se colocan las instrucciones que podrían provocar problemas dentro de un bloque *try* y se colocan a continuación uno o más bloques *catch*, de tal forma que si se provoca un error de un determinado tipo, se salta al bloque *catch* capaz de gestionar ese tipo de error específico. El bloque *catch* contendrá el código necesario para gestionar ese tipo específico de error. Suponiendo que no se hubiesen provocado errores en el bloque *try*, nunca se ejecutarían los bloques *catch*.^[56]

3.4. Validación de la solución propuesta

3.4.1. Validación de requisitos

La validación de requisitos es la actividad de la IR que trata de mostrar que éstos realmente definen el sistema que el cliente desea; además revisa que no se haya omitido ninguno, que no sean ambiguos, redundantes o inconsistentes. Como resultado final de la validación de requisitos se obtiene un listado de problemas y acciones recomendadas, que permite darle seguimiento y eliminar los problemas detectados. La validación de requisitos es importante pues de ella depende que no existan elevados costos de mantenimiento para el *software* desarrollado, ya que el costo de arreglar un problema en los requerimientos haciendo un cambio en el sistema es mucho mayor que reparar los errores de diseño o los de codificación.^[57]

3.4.1.1. Criterios de validación de requisitos

En el proceso de desarrollo de *software* con enfoque ágil basado en el nivel 2 de CMMI se definen varios criterios que permiten validar los requerimientos del cliente, donde se responde a varias interrogantes que validan si un requerimiento se aprueba o no:

- ¿El proveedor del requerimiento es un proveedor válido?
- ¿El requerimiento está identificado como único?
- ¿El requerimiento es modificable?
- ¿El requerimiento no es ambiguo?
- ¿El requerimiento está completo?
- ¿El requerimiento es congruente con otros requerimientos relacionados?



- ¿El requerimiento puede ser implementado?
- ¿El requerimiento puede ser probado?
- ¿El resultado de la evaluación de impacto es positivo?
- ¿El requerimiento está correcto?
- ¿El requerimiento es traceable?

3.4.1.2. Técnicas de validación de requisitos

Ian Sommerville propone tres técnicas de validación de requisitos que se pueden utilizar en conjunto con la verificación de los requisitos o de forma individual. A continuación se explican cada una de ellas:

- **Revisión de requisitos:** los requisitos se analizan sistemáticamente por un equipo de revisores expertos. Con esta técnica se pudo revisar el documento de requisitos en cuanto a anomalías y omisiones. La revisión de los requisitos se puede hacer de manera formal o informal, en este caso la revisión se realizó de manera informal. Mediante la revisión informal se trataron los requisitos con la mayor cantidad de involucrados posibles para confirmar que los requerimientos documentados son realmente lo que inicialmente deseaban y permitió detectar problemas simplemente hablando del sistema con los involucrados, sin necesidad de llevar a cabo una reunión para una revisión formal.
- **Construcción de prototipos:** consiste en realizar un modelo ejecutable del sistema para que clientes y usuarios finales puedan experimentar con él y determinar si cumple sus necesidades reales. Se utilizó con el propósito de que los usuarios adquirieran una experiencia directa con las interfaces de la aplicación, de manera que pudieran definir si eran de su agrado. La utilización de esta técnica permitió a los involucrados en el sistema adquirir nuevas ideas para los requisitos, facilitando que se encontraran áreas fuertes y débiles en el *software*, garantizando su mejoramiento o la proposición de nuevos requerimientos del sistema, permitió además comprobar la corrección y completitud de la especificación de requisitos.
- **Generación de casos de prueba:** el método existente para comprobar la verificabilidad de un requisito es que sea posible definir uno o varios casos de prueba para dicho requisito, esta técnica consiste en la definición de casos de prueba que permitan verificar el cumplimiento de los requisitos funcionales. Si un caso de prueba es difícil o imposible de diseñar quiere decir que los requerimientos serán difíciles de implementar y deberían ser considerados nuevamente. Esta



técnica se utilizó con el objetivo de descubrir defectos en el sistema y al mismo tiempo mostrar que el sistema satisface los requerimientos planteados.[58]

3.4.1.3. Proceso de verificación de los requisitos.

En conjunto con la validación de los requerimientos se llevó a cabo una verificación sobre los requisitos, donde se comprobaron los siguientes parámetros:

Validez: se tuvo en cuenta el punto de vista de todos los usuarios, con esto se pudieron identificar funciones adicionales así como perfeccionar las existentes.

Consistencia: se aseguró que no hubiera restricciones o descripciones contradictorias en una misma función del sistema.

Completitud: se pudieron incluir requerimientos que definen todas las funciones y restricciones propuestas por el usuario del sistema.

Realismo: se aseguró que se pudieran implementar todos los requerimientos haciendo uso de la tecnología actual.

Verificabilidad: Se escribieron un conjunto de pruebas que demostraron que el sistema cumple con cada requisito especificado.[57]

Al concluir el proceso de validación y verificación se detectaron algunos errores, entre los cuales se encuentran:

- Interpretación incorrecta de las funcionalidades y características solicitadas por el cliente.
- Descripciones insuficientes de algunos requerimientos.
- Componentes innecesarios en los prototipos de interfaz.
- Funcionamiento poco usable para el cliente en el diseño de los prototipos de interfaz.
- Faltas de ortografía en las especificaciones de los requisitos.
- Definición incorrecta en la definición de la prioridad de implementación para varios requisitos.

Después de haber aplicado los criterios definidos para validar requisitos del cliente en conjunto con las técnicas de validación explicadas y la verificación de requisitos, se detectaron una serie de problemas a los cuales se les dio seguimiento y fueron eliminados en su totalidad, esto permitió que fuera aprobado el 100% de los requisitos, dando paso a la implementación de la aplicación.

3.4.2. Pruebas de software



Las pruebas de *software* son los procesos que permiten verificar y revelar la calidad de un sistema informático antes de su puesta en marcha. Permiten comprobar el grado de cumplimiento de las especificaciones iniciales del sistema y no pueden asegurar la ausencia de defectos, sólo pueden demostrar que existen defectos en el *software*. [59]

Pruebas unitarias: el objetivo es detectar errores en los datos, lógica y algoritmos. Se enfoca en ejecutar cada unidad mínima a ser probada (ej.= una clase), lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. Busca asegurar que el código funciona de acuerdo con las especificaciones, para esto se usa el método de pruebas de caja blanca para cada unidad. Los casos de prueba deben diseñarse de forma tal que se recorran todos los caminos de ejecución posible dentro del código bajo prueba, por lo tanto el diseñador debe construirlos con acceso al código fuente de la unidad a probar. En la prueba de caja blanca se prueban las rutas lógicas del *software* y la colaboración entre componentes, al proporcionar casos de prueba que ejerciten conjuntos específicos de condiciones, bucles o ambos. Este tipo de prueba se realizó de manera automática.

Pruebas del sistema: el objetivo en este nivel de pruebas es verificar el ingreso, procesamiento y recuperación apropiada de datos, así como la implementación adecuada de las reglas del negocio. Para la realización de las pruebas del sistema se usa el método de pruebas de caja negra, que son pruebas funcionales sin acceso al código fuente de las aplicaciones, donde se trabaja con entradas y salidas.

- **Pruebas funcionales:** son el proceso de probar una entrega del sistema que será distribuida a los clientes. El objetivo de estas pruebas es comprobar que el sistema satisface sus requerimientos y para esto deben demostrar la entrega de la funcionalidad especificada, rendimiento y confiabilidad y que no falla durante su uso normal. Para identificar los valores de los datos de entrada se utilizó la técnica de particiones o clases de equivalencias, esta técnica consiste en ejecutar el flujo básico de las funcionalidades utilizando datos válidos e inválidos para verificar que los resultados esperados ocurren cuando se utiliza un dato válido, que los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido y que cada regla de negocio es aplicada adecuadamente. Las clases de equivalencia se identifican examinando cada condición de entrada y dividiéndola en dos o más grupos. Se definen dos tipos de clases de equivalencia: clases válidas, que no son más que entradas válidas al programa y clases no válidas, que son valores de entrada erróneos.



Pruebas de aceptación: En este nivel de pruebas se tiene como objetivo comprobar si el producto final está listo para su uso operativo en el entorno del usuario final. A diferencia de los restantes tipos de pruebas, las pruebas de aceptación son realizadas por los usuarios expertos y en este caso se utilizó el tipo de prueba Beta, que consiste en revisar la aplicación desde el lugar de trabajo del cliente, es decir, en un ambiente no controlado por los desarrolladores. Para esto se utilizó el método funcional, lo que permitió probar la aplicación a nivel de interfaz haciendo uso de entradas y salidas de *software*. [60]

3.4.3. Resultado de las pruebas

Pruebas unitarias

Haciendo uso de la librería JUnit se aplicaron las pruebas de caja blanca en 4 iteraciones, donde en la primera iteración se detectaron 16 no conformidades que fueron corregidas en su totalidad. En una segunda iteración se encontraron 8 no conformidades que igualmente fueron resueltas. En la tercera iteración se encontraron 2 no conformidades que fueron resueltas. Finalmente en la cuarta iteración no se detectaron no conformidades obteniéndose un resultado satisfactorio. A continuación se muestra el ejemplo de una prueba realizada. Ejemplos de otros resultados se muestran en los Anexos, Tabla (32-38).

Tabla 5. Casos de prueba de caja blanca. RF2.

Prueba estructural o caja blanca	Código caso de prueba: GENIA_PRE_PUCPCB-CC: RFR2.
Probador: Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: Los datos de entrada serán los atributos de la configuración a crear.	
Datos de entrada:	Objeto nulo.
Tipo de dato esperado:	Vacío.
Función de evaluación: <pre>public void testAdicionarConfiguracion() throws Exception { System.out.println("AdicionarConfiguracion"); Configuracion configuracion = null; CConfiguracion instance = CConfiguracion.getConfiguracion(); instance.AdicionarConfiguracion(configuracion); }</pre>	
Evaluación del caso de prueba: Satisfactorio.	



Pruebas funcionales

Para validar la aplicación se realizaron pruebas de caja negra en 4 iteraciones, obteniéndose los siguientes resultados en cada una de las iteraciones realizadas:

- En la primera iteración se obtuvieron 14 no conformidades. Entre las que se encuentran la no visualización de los mensajes de error encima del nombre de algunos componentes y la no indicación de las acciones de contexto cuando se posicionaba el cursor encima de la acción. Estas no conformidades fueron corregidas.
- En la segunda iteración se detectaron 9 no conformidades. Dentro de las que se encuentran: La no validación para la entrada de datos incorrectos en algunos campos de entrada de datos. La no visualización del mensaje de advertencia para la acción de contexto Eliminar. Estas no conformidades también fueron rectificadas.
- En una tercera iteración se detectaron 5 no conformidades: errores ortográficos, que fueron rectificadas.
- Ya en una cuarta iteración el sistema estaba libre de no conformidades, cumpliendo con los requisitos funcionales especificados.

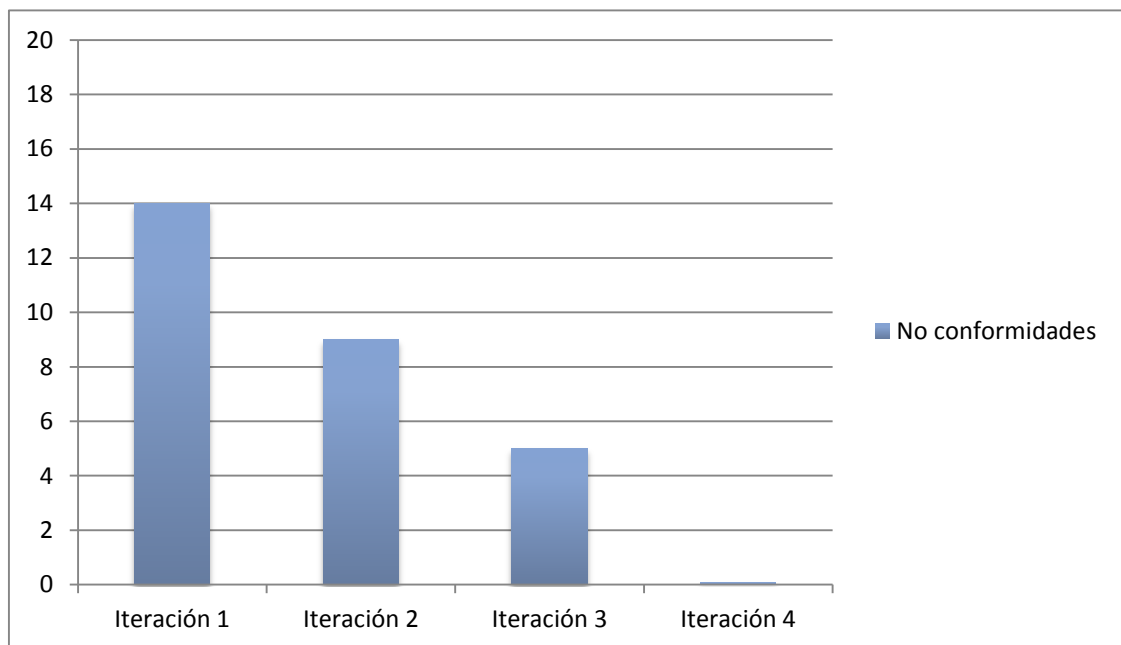


Figura 27: Resultado de las pruebas funcionales.



A continuación se muestra el diseño de casos de prueba correspondiente a la funcionalidad Crear tarea. Los restantes diseños de casos de pruebas pueden ser consultados en Anexos, Tabla (39 - 65).

Tabla 6. Diseño de casos de prueba. Crear tarea. Parte uno.

Escenario	Descripción	Nombre	Fecha de ejecución	Hora	Minutos	AM
EC 1.1 Insertar datos correctamente.	Mediante este escenario se introducen datos correctos en el sistema para crear una tarea.	V	V	V	V	V
		Tarea_nueva	lunes 8 de abril de 2012	1	1	AM
		V	V	V	V	V
		Tarea_de_chequeo	lunes 8 de abril de 2012	1	1	AM
		V	V	V	V	V
		Tarea_controladora	martes 9 de abril de 2012	5	10	Deshabilitado
		V	V	V	V	V
EC 1.2 Insertar elementos repetidos.	Mediante este escenario se introducen datos para crear una tarea que ya existe en el sistema.	I	v	V	V	V
		Tarea_nueva	lunes 8 de abril de 2012	1	0	AM
		I	V	V	V	V
		Tarea_de_chequeo	viernes 9 de octubre de 2012	7	18	AM
EC 1.3 Insertar	Mediante este	I	I	V	V	V



datos incompletos.	escenario no se introducen todos los datos para crear una tarea.	Vacío	Vacío	4	30	AM
		V	V	I	I	V
		Tarea_bb	sábado 30 de marzo de 2013	2	Vacío	Deshabilitado
		V	V	V	V	V
		Se introduce un texto que tenga como mínimo 1 caracter y como máximo 30, donde las palabras están separadas por guión bajo.	sábado 21 de abril de 2013	9	20	AM
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para crear una tarea.	I	V	V	V	V
		Tarea 123 #\$\$%	martes 13 de noviembre de 2012	6	52	AM
		I	V	V	V	V
		Tarea_de_control_asdffgtyh_drtyuuijkl	martes 13 de noviembre de 2012	10	10	AM
EC 1.5 Cancelar operación.	Mediante este escenario se cancela la creación de una tarea.	NA	NA	NA	NA	NA

Tabla 7. Diseño de casos de prueba. Crear tarea. Parte dos.

PM	Reglas	Respuesta del sistema	Flujo central
V	V	El sistema actualiza el	-El usuario una vez dentro de la vista
Deshabilitado	regla1	listado y muestra el	principal de la aplicación selecciona Tareas



V	V	mensaje de información	en las opciones del menú.
Deshabilitado	regla1	"El elemento ha sido creado satisfactoriamente".	-El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona la acción de Crear, llena los datos correctamente y presiona el botón Aceptar.
V	V		
PM	regla1 regla2		
V	V		
PM	regla2		
V	V	El usuario presiona el	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona la acción de Crear. -El usuario introduce datos de una tarea que ya está en el sistema y presiona el botón Aceptar.
Deshabilitado	regla1	botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".	
V	V		
Deshabilitado	regla1 regla2 regla5		
V	V	El sistema muestra	
Deshabilitado	regla2 regla6	encima del nombre del componente que debe ser llenado obligatorio	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona la acción de Crear, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
V	V	el mensaje de color rojo: "Campo requerido".	
PM	regla3 regla5		
V	I		
Deshabilitado	Vacio		
V	V	El sistema muestra	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas
Deshabilitado	regla7	encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	



V	V	El sistema no permite la entrada de más de 30 caracteres.	existentes. -El usuario selecciona la acción de Crear, introduce datos incorrectos y presiona el botón Aceptar.
Vacío	regla1 regla2		
NA	NA	El sistema muestra el mensaje de confirmación: ¿Está seguro que desea salir?	-El usuario una vez en la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que pueden realizar y el listado de tareas existentes. -El usuario selecciona la acción de Crear, llena o no los datos y presiona el botón Cancelar.

Pruebas de aceptación

Después de haber presentado la propuesta de solución al arquitecto de base de datos del SGU y al jefe del Departamento de Gestión Universitaria, que constituyen los usuarios finales. Se obtuvo como resultado la aprobación para el uso y explotación de la herramienta, lo cual quedó contemplado en el Acta de Aceptación del Cliente.

Conclusiones parciales

En el presente capítulo se mostraron los estándares de código utilizados, que facilitaron el trabajo del programador a la hora de implementar el sistema. Se realizó la validación de la aplicación primeramente llevando a cabo la validación de los requisitos funcionales y no funcionales y aplicando diferentes pruebas para asegurar la fiabilidad y demostrar la calidad del sistema una vez terminado. Durante todo el proceso de validación se encontraron errores de inconsistencias en las características y funcionalidades de la aplicación, lo que permitió considerar recomendaciones y darle seguimiento a los problemas detectados hasta que finalmente fueron erradicados en su totalidad logrando la entrega eficiente del Policía de datos del SGU.



Conclusiones

Una vez realizada la fundamentación teórica que sustentó la presente investigación, definidas las características del Policía de datos del SGU y efectuado su desarrollo y validación, se obtuvieron resultados que permiten plantear las siguientes conclusiones:

- Con el estudio de los sistemas que dan seguimiento a la calidad de los datos en el mundo quedó demostrado que aunque aportaron los conocimientos necesarios para una mayor comprensión del objeto de estudio y para sentar las bases de la investigación no cumplen con las necesidades actuales del SGU, por lo que quedaron descartados para ser utilizados.
- La validación de los requisitos y las pruebas realizadas, permitieron comprobar que los requerimientos definidos en el sistema son los que realmente el cliente desea, asegurar que el código funciona de acuerdo con las especificaciones y verificar la implementación adecuada de las reglas del negocio.
- El desarrollo del Policía de datos dotó al SGU de una solución configurable que controla el cumplimiento de las reglas del negocio para la información generada, lo que permite alcanzar y mantener altos parámetros de las características de la calidad de los datos almacenados.

Debido a lo anteriormente planteado se puede decir que se cumplió el objetivo principal de la investigación, obteniéndose los resultados esperados.



Recomendaciones

- Añadir nuevos operadores en la construcción de reglas del negocio para enriquecer el funcionamiento de la aplicación.
- Integrar la solución con la herramienta *DBAnalyzer* realizada en la UCI, que permite verificar la calidad de los datos mediante cálculos estadísticos.



Bibliografía referenciada

1. *Esparza Cruz, N. and C. Maldonado Camposano, IMPLEMENTACIÓN DE UN SOFTWARE INFORMATIVO DE TEMAS DE TESIS REALIZADAS Y EN PROCESO EN LA FACULTAD DE ADMINISTRACIÓN DE FINANZAS E INFORMÁTICA DE LA UNIVERSIDAD TÉCNICA DE BABAHOYO. 2011. p. 16.*
2. *Busso, D.E., Calidad de Datos. 2012, Universidad Tecnológica Nacional: Córdoba. p. 5.*
3. *Dr. Claudio Alberto Rivera Rodríguez, et al., Metodología de la investigación. 2009. 94.*
4. *Fisicoquímica. El método científico. [Página web] 2009 2013 [cited 2012 diciembre]; Available from: <http://es.scribd.com/doc/14002983/Metodo-Cientifico-2009-Con-Ejemplos>.*
5. *Miranda, A.R.T. Bases de Datos. [Página Web] 2007 [cited 2012 25 de octubre de 2012]; Elmasri 4ta.edicion:[Available from: <http://basededatosrelacionales.blogspot.com/2007/10/concepto-de-base-de-datos-relacionales.html>.*
6. *Busso, D.E., Calidad de Datos. 2012, Universidad Tecnológica Nacional: Córdoba. p. 2.*
7. *Martín Gutiérrez, P., Data Warehouse: marco de calidad. 2012: p. 146.*
8. *Busso, D.E., Calidad de Datos. 2012, Universidad Tecnológica Nacional: Córdoba. p. 3.*
9. *Martín Gutiérrez, P., Data Warehouse: marco de calidad. 2012: p. 147,148.*
10. *Gutiérrez, P.M., Data Warehouse: Marco de Calidad. 2012.*
11. *Group., B.R. Business Rules Group. 2013 [cited 2013 20 de enero]; Available from: http://www.businessrulesgroup.org/first_paper/br01c1.htm#s1e.*
12. *Saffirio, M. Mario Saffirio Tecnologías de Información y Gestión de Procesos de Negocios (BPM). 2011 [cited 2013 22 de enero del 2013]; Available from: <http://msaffirio.wordpress.com/2011/08/20/reglas-de-negocio-business-rules/>.*
13. *Busso, D.E., Calidad de Datos. 2012, Universidad Tecnológica Nacional: Córdoba. p. 16.*
14. *Busso, D.E., Calidad de Datos. 2012, Universidad Tecnológica Nacional: Córdoba. p. 17,19.*
15. *RAHM, E., DATA CLEANING PROBLEMS AND CURRENT APPROACHES. 2011. p. 2,3 y 12.*
16. *Díaz, J.C. (2009) Data Quality: un vistazo a las soluciones Open Source existentes.*
17. *Uniserv. Uniserv. [cited 2013 20 de enero del 2013]; Available from:*



- <http://www.uniserv.com/es/products/data-quality-solutions/data-profiling-data-quality.php>.
18. Uniserv. Uniserv. [cited 2013 21 de enero del 2013]; Available from: <http://www.uniserv.com/es/products/data-quality-solutions/data-monitoring-data-quality.php>.
 19. Uniserv. Uniserv. [cited 2013 22 de enero del 2013]; Available from: <http://www.uniserv.com/es/products/data-quality-solutions/data-cleansing-data-quality.php>.
 20. Interface, H. Datacleaner. [cited 2013 2 de febrero del 2013]; Available from: <http://www.datacleaner.org/>.
 21. Eniac. Eniac. [Página Web] [cited 2012 5 de diciembre de 2012]; Available from: <http://www.eniac.com/productos/trillium.htm>.
 22. Software, H.-H.T., Trillium Software System® Descripción General del Producto. 2010.
 23. Oracle. java. [cited 2013 7 de febrero del 2013]; Available from: <http://www.java.com/es/about/>.
 24. Taller, W. Manual de Java. [Página web] 2011 [cited diciembre 2012]; Available from: <http://www.webtaller.com/manual-java/caracteristicas-java.php>.
 25. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. Observatorio Tecnológico. [Página Web] [cited 2013 enero]; Available from: <http://recursostic.educacion.es/observatorio/web/es/software/programacion/675-xml>.
 26. Educate Ahora, I. Buenas tareas. 2013 [cited 2013 10 de enero del 2013]; Available from: <http://www.buenastareas.com/ensayos/Caracteristicas-Principales-De-Uml/2234495.html>.
 27. patrycarmenate. buenastareas. 2011 [cited 2013 11 de enero del 2013]; Available from: <http://www.buenastareas.com/ensayos/Lenguaje-Unificado-Uml/2747273.html>.
 28. Oracle. Oracle. 2012 [cited 2013 25 de enero del 2013]; Available from: <http://www.oracle.com/lad/corporate/press/pr-lad-23-jan-2012-1498968-esa.html>.
 29. FDM, E.d.D.d. Free Download Manager. [Página Web] 2012 [cited 2013 16 de enero]; Available from: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
 30. Overview, V.P.C. Visual Paradigm. [Página Web] [cited 2012 9 de diciembre]; Available from: <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.
 31. Company, E. Pencil Project. [Página web] 2012 [cited 2013 14 de enero]; Available from: pencil.evolus.vn/.



32. Guerrero, R.M. PostgreSQL-es. [Blog] 2009 [cited 2013 10 de enero]; Available from: http://www.postgresql.org.es/sobre_postgresql.
33. Argentina, G.d.U.d.P.d. ArPug [Página web] 2010 [cited 2012 11 de diciembre]; Available from: <http://www.arpug.com.ar/trac/wiki/PgAdmin>
34. Gómez, R. Pruebas software con Junit 4 y Eclipse. [Página web] 2010 20/01/2010 [cited 2013 marzo]; Available from: <http://www.slideshare.net/siticfje/pruebas-software-con-junit>.
35. Blanco, K.R., et al., Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas, in Revista Cubana de Ciencias Informáticas. 2012, Ediciones Futuro: La Habana.
36. Dr Roberto Hernández Sampieri, D.C.F.C., Dra Pilar Baptista Lucio, Metodología de la Investigación Cuarta Edición. 2006, México.
37. Larman, C., MODELO DEL DOMINIO, in UML y Patrones. 2ª Edición. 2003. p. 23.
38. Pressman, R., Ingeniería de requisitos, in Ingeniería del Software, Un Enfoque Práctico. p. 157.
39. Amador Durán Toro and B.B. Jiménez, Metodología para la Elicitación de Requisitos de Sistemas Software. 2000, Sevilla. 78.
40. Rojas, M.C.J.c.O., Técnicas para la obtención de requerimientos. 2012, Secretaría de Educación Pública.
41. Rodríguez, M.D.I.H. Gestión de Requisitos. [Página web] 2012 [cited 2013 diciembre]; Available from: <http://www.monografias.com/trabajos92/gestion-requisitos/gestion-requisitos.shtml>.
42. Sommerville, I., Parte II. Requerimientos, in Ingeniería del Software. 2005, Pearson Educación, S.A: Madrid. p. 110.
43. Pressman, R., Ingeniería de requisitos, in Ingeniería del Software, Un Enfoque Práctico. p. 160.
44. Sommerville, I., Parte II. Requerimientos, in Ingeniería del software S. Pearson Educación, Editor. 2005: Madrid, España. p. 111.
45. Cesar de la Torre Llorente, et al., Guía de arquitectura N_Capas orientado al dominio con. net 4.0(Beta). 2010, España. 433
46. Ayazo, C.A.T. Diagrama de despliegue. [Página web] 2013 [cited 2013 marzo]; Available from: <http://prezi.com/svxjuygcrpse/diagrama-de-despliegue-uml/>.
47. Pressman, R., Ingeniería del diseño, in Ingeniería del Software, Un Enfoque Práctico. p. 270.



48. *Informáticos, D.d.L.y.S., Patrones de Asignación de Responsabilidades (GRASP), in Patrones de Asignación de Responsabilidades (GRASP), escuela técnica superior de ingeniería informática. p. 27.*
49. *planet, G.t. Patrones GoF. [Página web] 2011 [cited 2013 febrero]; Available from: <http://geektheplanet.net/5462/patrones-gof.shtml>.*
50. *Blaha, M. Patterns of Data Modeling. 2010 [cited 2013 20 de abril del 2013]; Available from: http://www.ecured.cu/index.php/Patrones_de_dise%C3%B1o_de_bases_de_datos.*
51. *eHow. Tres fases del diseño tradicional de bases de datos. [Página web] 2013 [cited 2013 abril]; Available from: http://www.ehowenespanol.com/tres-fases-del-diseno-tradicional-bases-datos-info_197446/.*
52. *Daniel José Salas Álvarez, M.e.I. Estándares de codificación Java. [Página web] 2010 [cited 2013 marzo]; Available from: <http://www.aves.edu.co/ovaunicor/recursos/view/265>.*
53. *Vieites, L., Convenciones de código para el lenguaje de programación JAVA™. 2007.*
54. *Centro de Bachillerato Tecnológico industrial y de servicios # 122, C. Principios de Programación (DFD). [Página web] 2010 28 de diciembre de 2010 [cited 2013 marzo]; Available from: <http://plataforma.cbtis122.net/mod/page/view.php?id=1195>.*
55. *Ciberaula. Programación orientada a objetos. [Página web] 2010 [cited 2013 marzo]; Available from: http://www.ciberaula.com/articulo/tecnologia_orientada_objetos/.*
56. *Muñoz, J., Excepciones, in Desarrollo y Gestión de Aplicaciones Web con Tecnología Java. 2013. p. 44.*
57. *Sommerville, I., Parte II Requerimientos, in Ingeniería del Software. 2005, Pearson Education, S.A: Madrid. p. 144.*
58. *Sommerville, I., Parte II Requerimientos, in Ingeniería del Software. 2005, Pearson Education, S.A: Madrid. p. 145-146.*
59. *Cibertec. Pruebas de Software. [Página web] 2012 [cited 2013 abril]; Available from: <http://code.google.com/p/cibertec/downloads/detail?name=Pruebas%20de%20Software.pdf>.*
60. *García, I.M.L., Pruebas de software. 2013: México.*



Bibliografía consultada

1. Dr Roberto Hernández Sampieri, D.C.F.C., Dra Pilar Baptista Lucio, *Metodología de la Investigación Cuarta Edición*. 2006, México.
2. González, Y.I., *Descubrimiento de llaves foráneas en colecciones de datos: reglas de inclusión*. Serie Científica, 2010. **3**(10).
3. Martínez Fernández, J.L., *Introduciendo Semántica en un Proceso de Desarrollo Software a través de Reglas de Negocio*. 2010, Telecomunicacion.
4. Asenjo, J.S., *Sistemas Gestores de Bases de datos*, J. Sánchez, Editor. 2009.
5. Evans, C.C. *yaml*. Available from: <http://www.yaml.org/>
6. Briano, F. *Introducción a Spring Framework Java* 29 de noviembre del 2010; Available from: <http://picandocodigo.net/2010/introduccion-a-spring-framework-java/>.
7. *casidiablo.net*. *Procesando XML en Java*. 2008; Available from: <http://casidiablo.net/procesamiento-xml-java-xerces/>.
8. King, G. *Documentación de referencia de Hibernate*. 2004; Available from: http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html_single/.
9. León, A.R.S., *Compendio de consultas útiles al catálogo de postgresQL*. 2011: La Habana.
10. Mary Beth Chrissis, Mike Konrad, and Sandy Shrum, in *CMMI, Guía para la integración de procesos y la mejora de productos*. 2009, Pearson Educación, S.A. p. 368.
11. ERIKA CAMACHO, FABIO CARDESO, and GABRIEL NUÑEZ, *ARQUITECTURAS DE SOFTWARE*. 2004. p. 68.
12. Ansejo, J.S. *apuntes completos, Sistemas gestores de Bases de Datos*. [Página web] 2009 [cited 2013 febrero]; Available from: <http://ubuntuone.com/p/sqt/>.
13. Riquelme, M.S.L. and M.M.C. Morales, *Sistema Gestor de Base de Datos Relacionales*. 2010. p. 14.
14. Tedeschi, N. *¿Qué es un Patrón de Diseño?* [Página web] 2013 [cited 2013 abril]; Available from: <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.



Glosario de términos

API: es el conjunto de funciones y procedimientos (métodos, en la programación orientada a objetos) que ofrece una biblioteca para ser utilizada por otro *software* como una capa de abstracción.

Código procedural: representa el código de un programa estructurado en componentes, que reciben el nombre de procedimientos, subrutinas o funciones.

GUID: se usan para distinguir interfaces de componentes de *software* diferentes. Esto significa que dos versiones de un componente pueden tener exactamente el mismo nombre, pero ser diferente de cara al usuario en caso de que el *GUID* haya cambiado.

Identación: se utiliza para mejorar la legibilidad del código fuente por parte de los programadores, teniendo en cuenta que los compiladores o intérpretes raramente consideran los espacios en blanco entre las sentencias de un programa.

Metadato: proporciona la información mínima necesaria para identificar un recurso; incluye la información sobre el contexto, contenido y control de un dato.

Serialización: consiste en un proceso de codificación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato más legible como XML.



Anexos

Tabla 8. Modelo de entrevista.

MODELO DE ENTREVISTA
Nombre(s):
Apellidos:
Cargo que Ocupa:
Departamento al que pertenece:
Preguntas
<ol style="list-style-type: none">1. ¿En qué consiste el proceso de seguimiento de la calidad de los datos?2. ¿Cuáles son las características fundamentales de este proceso en el Sistema de Gestión Universitaria (SGU)?3. ¿Qué usuarios se encargan de darle seguimiento a la calidad de los datos en el SGU?4. ¿Actualmente existe algún mecanismo para llevar a cabo el seguimiento de la calidad de los datos en el SGU?5. ¿Qué limitaciones presenta el seguimiento de la calidad de los datos actualmente?6. ¿Qué desearía aportar a la forma en que se realiza el seguimiento de la calidad de los datos para hacer el trabajo más eficiente?

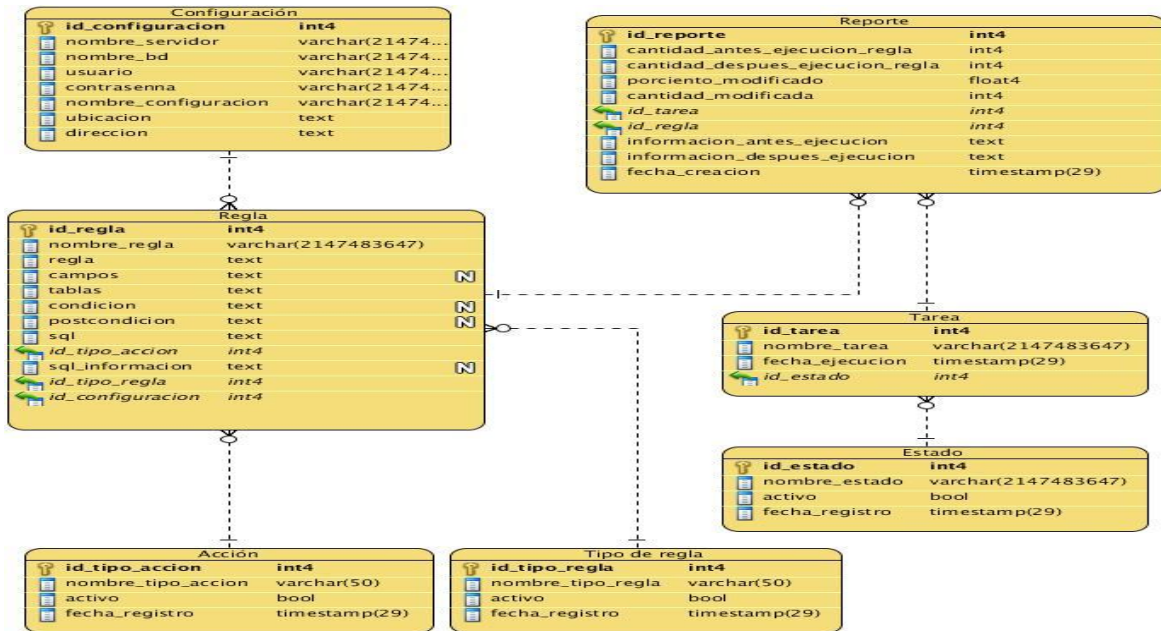


Figura 28: Modelo lógico de datos.

Tabla 9. Especificación de requisito. Mostrar configuraciones.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF1.]	Mostrar configuraciones.	Permite mostrar un listado con las configuraciones existentes. Se muestra la opción de Crear una configuración, así como Modificar y Eliminar una configuración seleccionada. El usuario puede abandonar la ventana presionando el botón Salir o por la cruz que se encuentra en la parte superior derecha.	Media	Media
Prototipo				



<div style="border: 1px solid gray; padding: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> Mostrar configuraciones - X </div> <div style="text-align: right; margin-top: 10px;"> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 33%;">Nombre</th> <th style="width: 33%;">Dirección</th> <th style="width: 33%;">Base de datos</th> </tr> </thead> <tbody> <tr> <td>aa</td> <td>/home/jdaniel/Desktop/aa.xml</td> <td>s</td> </tr> <tr> <td>e</td> <td>/home/jdaniel/cron/e.xml</td> <td>entre</td> </tr> <tr> <td>prueba</td> <td>/media/JD1/conf/prueba.xml</td> <td>prueba</td> </tr> </tbody> </table> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Salir"/> </div> </div>			Nombre	Dirección	Base de datos	aa	/home/jdaniel/Desktop/aa.xml	s	e	/home/jdaniel/cron/e.xml	entre	prueba	/media/JD1/conf/prueba.xml	prueba
Nombre	Dirección	Base de datos												
aa	/home/jdaniel/Desktop/aa.xml	s												
e	/home/jdaniel/cron/e.xml	entre												
prueba	/media/JD1/conf/prueba.xml	prueba												
Prototipo Mostrar configuraciones.														
Campos	Tipos de Datos	Reglas o Restricciones												
Observaciones	<ol style="list-style-type: none"> Interactúa con esta acción el administrador de base de datos. 													

Tabla 10. Especificación de requisito. Crear configuración.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF2.]	Crear configuración.	Permite crear una configuración. Se muestran los campos a llenar y una vez creada la configuración se muestra un mensaje de información "El elemento ha sido creado satisfactoriamente".	Alta	Alta
Prototipo				



Crear configuración - X

Nombre de configuración:* <input style="width: 90%;" type="text"/>	Ubicación:* <input style="width: 90%;" type="text"/>	<input type="button" value="Examinar"/>
Nombre de servidor:* <input style="width: 90%;" type="text"/>	Base de datos:* <input style="width: 90%;" type="text"/>	
Contraseña:* <input style="width: 90%;" type="text"/>	Usuario:* <input style="width: 90%;" type="text"/>	<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>

Prototipo Crear configuración.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre configuración	character varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.
Ubicación	character varying	Indica la dirección donde va a guardarse la configuración que se está creando. Solo admite la dirección de un directorio que sea válido. Campo obligatorio.
Nombre servidor	character varying	Indica el servidor donde está la base de datos a la que se le va a hacer la configuración. Admite una dirección ip o un nombre de dominio. Campo obligatorio.



Base de datos	character varying	Indica la base de datos dentro del servidor. Admite un rango de 1 a 30 caracteres con valores alfanuméricos y guión bajo. Campo obligatorio.
Usuario	character varying	Representa el nombre del usuario requerido para acceder a la base de datos donde se va a crear la configuración. Admite un rango de 1 a 30 caracteres con valores alfanuméricos y guión bajo. Campo obligatorio.
Contraseña		Indica la contraseña requerida para acceder a la base de datos. Admite cualquier caracter. Campo obligatorio.
Observaciones		<ol style="list-style-type: none">1. En caso que la configuración ya exista se muestra un mensaje de error “<i>El elemento ya existe</i>”.2. En caso de cancelar la acción se muestra un mensaje de advertencia “<i>¿Está seguro de realizar la acción?</i>”.2. En caso de que se introduzcan datos incorrectos se muestra y un mensaje en rojo encima del nombre del campo indicando el tipo de error.3. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: “<i>Campo requerido</i>” encima del nombre del campo que debe ser llenado obligatorio.4. Si se trata de introducir más de los caracteres permitidos para los campos el sistema no permite seguir introduciéndolos.5. Interactúa con esta acción el administrador de base de datos.

Tabla 11. Especificación de requisito. Modificar configuración.



Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF3.]	Modificar configuración.	Permite modificar una configuración. Se muestran los campos a llenar y una vez modificada la configuración se muestra un mensaje de información "El elemento ha sido modificado satisfactoriamente".	Baja	Alta
Prototipo				
Prototipo Modificar configuración.				
Campos	Tipos de Datos		Reglas o Restricciones	
Nombre configuración	character varying		Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.	
Ubicación	character varying		Indica la dirección donde va a guardarse la configuración que desea modificar. Solo admite la dirección de un directorio que sea válido. Campo obligatorio.	



Nombre servidor	character varying	Indica el servidor donde está la base de datos cuya configuración desea modificar. Admite una dirección ip o un nombre de dominio. Campo obligatorio.
Base de datos	character varying	Indica la base de datos dentro del servidor. Admite un rango de 1 a 30 caracteres con valores alfanuméricos y guión bajo. Campo obligatorio.
Usuario	character varying	Representa el nombre del usuario requerido para acceder a la base de datos donde se va a modificar la configuración. Admite un rango de 1 a 30 caracteres con valores alfanuméricos y guión bajo. Campo obligatorio.
Contraseña		Indica la contraseña requerida para acceder a la base de datos. Admite cualquier carácter. Campo obligatorio.
Observaciones	<ol style="list-style-type: none"> 1. En caso que la configuración ya exista se muestra un mensaje de error <i>“El elemento ya existe”</i>. 2. En caso de cancelar la acción se muestra un mensaje de advertencia <i>“¿Está seguro de realizar la acción?”</i>. 3. En caso de que se introduzcan datos incorrectos se muestra un mensaje de color rojo encima del nombre del campo indicando el tipo de error. 4. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: <i>“Campo requerido”</i> encima del nombre del campo que debe ser llenado obligatorio. 5. Si se trata de introducir más de los caracteres permitidos para los campos el sistema no permite seguir introduciéndolos. 6. Interactúa con esta acción el administrador de base de datos. 	

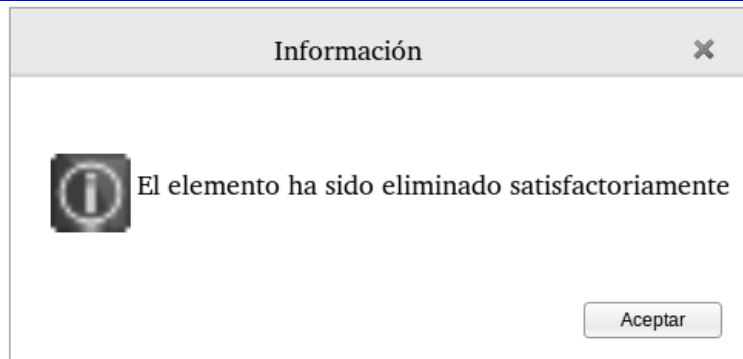
Tabla 12. Especificación de requisito. Eliminar configuración.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF4.]	Eliminar configuración.	Permite eliminar una configuración seleccionada. El usuario selecciona una configuración en el listado	Baja	Media

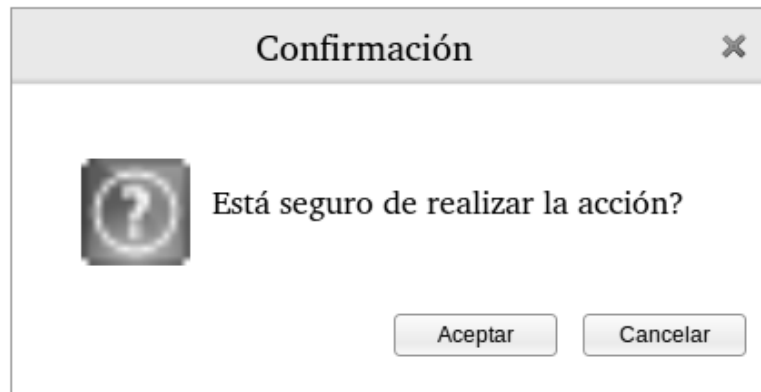


de las configuraciones existentes, seguidamente selecciona la opción Eliminar y se muestra el mensaje de advertencia ¿“Está seguro de realizar la acción”? El usuario presiona el botón Aceptar y se muestra el mensaje “El elemento ha sido eliminado satisfactoriamente”.

Prototipo



Prototipo Mensaje de confirmación.



Prototipo Mensaje de información.

Campos

Tipos de Datos

Reglas o Restricciones

Observaciones

1. Interactúa con esta acción el administrador de base de datos.



Tabla 13. Especificación de requisito. Mostrar objetos del negocio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF5.]	Mostrar objetos del negocio.	Permite mostrar un listado de los objetos del negocio correspondientes a una configuración seleccionada. Además se muestran las opciones de Crear, así como Eliminar, Modificar y Ver detalles de un objeto del negocio seleccionado.	Media	Media
Prototipo				
Prototipo Mostrar Objetos del negocio.				
	Campos	Tipos de Datos	Reglas o Restricciones	
	Observaciones	1. Interactúa con esta acción el administrador de base de datos.		



Tabla 14. Especificación de requisito. Crear objetos del negocio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF6.]	Crear objeto del negocio.	Permite crearle un objeto del negocio a una configuración seleccionada. Se muestran las tablas de la base de datos y los campos a llenar, además de la opción de Eliminar una propiedad seleccionada y Eliminar la unión de dos tablas. El usuario debe seleccionar la tabla de donde desee adicionar los campos al objeto del negocio. Si son campos de diferentes tablas se deben unir las tablas a las cuales pertenecen. Una vez creado el objeto del negocio se muestra un mensaje de información <i>“El elemento ha sido creado satisfactoriamente”</i> .	Alta	Alta
Prototipo				



Crear objeto del negocio

BD_SGU_pregrado
v_matriculados_pregrado
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
view_procesos_bloqueados
tb_dtramite_docente_ingreso
tb_nperiodo_instrucción
tb_rtipo_tamite_situacion_escolar
tb_destudiante
tb_ncausa_evaluacion
tb_nhuerfano
tb_rpersona_limitacion_salud
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
v_matriculados_pregrado
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
view_procesos_bloqueados
tb_dtramite_docente_ingreso
tb_nperiodo_instrucción
tb_rtipo_tamite_situacion_escolar
tb_ncausa_evaluacion
tb_nhuerfano
tb_rpersona_limitacion_salud
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
tb_dtramite_docente
tb_dtramite_docente_ingreso
tb_dpersona_bloqueado
tb_nsituacion_alumno_ayudante
tb_nhuerfano
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
tb_nperiodo_instrucción
tb_nsituacion_alumno_ayudante
tb_rtipo_tamite_situacion_escolar
tb_nperiodo_instrucción
v_matriculados_pregrado
tb_dtramite_docente
tb_nhuerfano
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
tb_dpersona_bloqueado

Nombre del objeto:*

Nombre visual:*

Campo:* -Seleccione-

Nomenclador

Asociar

Propiedades:

Propiedad	Nombre visual	Nomenclador	Tipo de dato	Referencia

Campo unión izquierdo: -Seleccione- = Campo unión derecho: -Seleccione-

Asociar

Uniones:

Condición

Aceptar Cancelar

Prototipo Crear objeto del negocio.



Crear objeto del negocio

BD_SGU_pregrado
v_matriculados_pregrado
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
view_procesos_bloqueados
tb_dtramite_docente_ingreso
tb_nperiodo_instrucción
tb_rtipo_tamite_situacion_escolar
tb_destudiante
tb_ncausa_evaluacion
tb_nhuerfano
tb_rpersona_limitacion_salud
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
v_matriculados_pregrado
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
view_procesos_bloqueados
tb_dtramite_docente_ingreso
tb_nperiodo_instrucción
tb_rtipo_tamite_situacion_escolar
tb_ncausa_evaluacion
tb_nhuerfano
tb_rpersona_limitacion_salud
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
tb_dtramite_docente
tb_dtramite_docente_ingreso
tb_dpersona_bloqueado
tb_nsituacion_alumno_ayudante
tb_nhuerfano
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
tb_nperiodo_instrucción
tb_nsituacion_alumno_ayudante
tb_rtipo_tamite_situacion_escolar
tb_nperiodo_instrucción
v_matriculados_pregrado
tb_dtramite_docente
tb_nhuerfano
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
tb_dpersona_bloqueado

Nombre del objeto:*
Nombre visual:*
Campo:*
-Seleccione-

Nomenclador
Campo visual: *
-Seleccione-
Campo principal: *
-Seleccione-
Asociar

Propiedades:

Propiedad	Nombre visual	Nomenclador	Tipo de dato	Referencia

Campo unión izquierdo: -Seleccione- = Campo unión derecho: -Seleccione-
Asociar

Uniones:
Condición

Aceptar Cancelar

Prototipo Crear objeto del negocio nomenclador.



Crear objeto del negocio
- X

- BD_SGU_pregrado
- ...v_matriculados_pregrado
- ...tb_dtramite_docente
- ...tb_ksituacion_alumno_ayudante
- ...tb_dsolicitud_alumno_ayudante
- ...view_procesos_bloqueados
- ...tb_dtramite_docente_ingreso
- ...tb_nperiodo_instrucción
- ...tb_rtipo_tamite_situacion_escolar
- ...tb_destudiante
- ...tb_ncausa_evaluacion
- ...tb_nhuerfano
- ...tb_rpersona_limitacion_salud
- ...tb_dpersona_bloqueado
- ...tb_ngrupo_sanguineo
- ...v_matriculados_pregrado
- ...tb_dtramite_docente
- ...tb_ksituacion_alumno_ayudante
- ...tb_dsolicitud_alumno_ayudante
- ...view_procesos_bloqueados
- ...tb_dtramite_docente_ingreso
- ...tb_nperiodo_instrucción
- ...tb_rtipo_tamite_situacion_escolar
- ...tb_ncausa_evaluacion
- ...tb_nhuerfano
- ...tb_rpersona_limitacion_salud
- ...tb_dpersona_bloqueado
- ...tb_ngrupo_sanguineo
- ...tb_dtramite_docente
- ...tb_dtramite_docente_ingreso
- ...tb_dpersona_bloqueado
- ...tb_ksituacion_alumno_ayudante
- ...tb_nhuerfano
- ...tb_dtramite_docente
- ...tb_ksituacion_alumno_ayudante
- ...tb_dsolicitud_alumno_ayudante
- ...tb_nperiodo_instrucción
- ...tb_ksituacion_alumno_ayudante
- ...tb_rtipo_tamite_situacion_escolar
- ...tb_nperiodo_instrucción
- ...v_matriculados_pregrado
- ...tb_dtramite_docente
- ...tb_nhuerfano
- ...tb_dpersona_bloqueado
- ...tb_ngrupo_sanguineo
- ...tb_dpersona_bloqueado

Nombre del objeto:*

Nombre visual:*

Campo:*
 -Seleccione-

Nomenclador

Origen de datos: *

Campo visual: *

Campo principal: *

Alias: *

Propiedades:

Propiedad	Nombre visual	Nomenclador	Tipo de datos	Referencia

Campo unión izquierdo:
 -Seleccione-

=

Campo unión derecho:
 -Seleccione-

Uniones:

Condición

Prototipo Crear objeto del negocio nomenclador_1.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre objeto	delcharacter varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.
Nombre visual	character varying	Admite un rango de 1 a 30 caracteres con valores



			alfanuméricos y guión bajo. Campo obligatorio.
Campo	character varying		Permite seleccionar los campos que pertenecerán al objeto del negocio que desea crear. Campo obligatorio.
Nomenclador	booleano		Se habilita cuando el Campo seleccionado referencia un nomenclador.
Origen de datos	character varying		Se muestra cuando no existe una referencia al nomenclador en la base de datos. Representa la tabla donde se encuentra el nomenclador o una función para obtener la información del nomenclador. Admite el nombre de una función válida o el nombre de una tabla (valores alfanuméricos y guión bajo). Campo obligatorio.
Campo visual	character varying		Permite seleccionar el campo visual del nomenclador. Si no existe una referencia al nomenclador en la base de datos se muestra un cuadro de texto para obtener información del nomenclador. Admite letras y guión bajo. Campo obligatorio.
Campo principal	character varying		Permite seleccionar el campo principal del nomenclador. Si no existe una referencia al nomenclador en la base de datos se muestra un cuadro de texto para obtener información del nomenclador. Admite letras y guión bajo. Campo obligatorio.
Alias	character varying		Permite tratar una función como una tabla de la base de datos. Admite letras y guión bajo. Se muestra cuando en el campo Origen de los datos se introduce una función. Campo obligatorio.
Campo unión izquierdo	character varying		Indica que se debe seleccionar el campo por el que se debe unir una tabla con otra en caso de que se deseen añadir campos de otras tablas.



Campo derecho	unión	character varying	Indica que se debe seleccionar el campo por el que se debe unir una tabla con otra en caso de que se deseen añadir campos de otras tablas.
<p>Observaciones</p> <ol style="list-style-type: none"> 1. En caso que el objeto del negocio ya exista se muestra un mensaje de error <i>“El elemento ya existe”</i>. 2. En caso de cancelar la acción se muestra un mensaje de advertencia <i>“¿Está seguro de realizar la acción?”</i> 3. En caso de que se introduzcan datos incorrectos se muestra un mensaje en rojo encima del nombre del campo indicando el tipo de error. 4. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: <i>“Campo requerido”</i> encima del nombre del campo que debe ser llenado obligatorio. 5. Si se trata de introducir más de los caracteres permitidos para los campos, el sistema no permite seguir introduciéndolos. 6. Interactúa con esta acción el administrador de base de datos. 			

Tabla 15. Especificación de requisito. Modificar objeto del negocio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF7.]	Modificar objeto del negocio.	Permite modificar un objeto del negocio de una configuración seleccionada. Se muestran las tablas de la base de datos y los campos a modificar, además de la opción de Eliminar una propiedad seleccionada y Eliminar la unión de dos tablas. El usuario debe seleccionar la tabla de donde desee adicionar los campos al objeto del negocio que desea modificar. Si son campos de diferentes tablas se deben unir las tablas a las cuales pertenecen. Una vez modificado el objeto del negocio se muestra un mensaje de información <i>“El elemento ha sido modificado satisfactoriamente”</i> .	Baja	Alta
Prototipo				



Modificar objeto del negocio

Nombre del objeto:* **Nombre visual:*** **Campo:*** id_estructura

Nomenclador

Propiedades:

Propiedad	Nombre visual	Nomenclador	Tipo de datos	Referencia
primer_nombre	primer_nombre	false	character varying	sq_gestion_perso
carnet_identidad	carnet_identidad	false	character varying	sq_gestion_perso
id_estado_civil	id_estado_civil	false	integer	sq_gestion_perso
id_peloton	id_peloton	false	integer	sq_gestion_perso

Campo unión izquierdo: -Seleccione- **Campo unión derecho:** -Seleccione-

Uniones:

Condición
tb_destudiante.id_persona = tb_dpersona_data_plus.id_persona

Prototipo Modificar objeto del negocio.



Modificar objeto del negocio

BD_SGU_pregrado
v_matriculados_pregrado
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
view_procesos_bloqueados
tb_dtramite_docente_ingreso
tb_nperiodo_instrucción
tb_rtipo_tamite_situacion_escolar
tb_destudiante
tb_ncausa_evaluacion
tb_nhuerfano
tb_rpersona_limitacion_salud
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
v_matriculados_pregrado
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
view_procesos_bloqueados
tb_dtramite_docente_ingreso
tb_nperiodo_instrucción
tb_rtipo_tamite_situacion_escolar
tb_ncausa_evaluacion
tb_nhuerfano
tb_rpersona_limitacion_salud
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
tb_dtramite_docente
tb_dtramite_docente_ingreso
tb_dpersona_bloqueado
tb_nsituacion_alumno_ayudante
tb_nhuerfano
tb_dtramite_docente
tb_nsituacion_alumno_ayudante
tb_dsolicitud_alumno_ayudante
tb_nperiodo_instrucción
tb_nsituacion_alumno_ayudante
tb_rtipo_tamite_situacion_escolar
tb_nperiodo_instrucción
v_matriculados_pregrado
tb_dtramite_docente
tb_nhuerfano
tb_dpersona_bloqueado
tb_ngrupo_sanguineo
tb_dpersona_bloqueado

Nombre del objeto:* est
Nombre visual:* id_estructura
Campo:* id_estructura
 Nomenclador
Campo visual:* nombre_estructura
Campo principal:* id_estructura
Asociar

Propiedades:

Propiedad	Nombre visual	Nomenclador	Tipo de datos	Referencia
primer_nombre	primer_nombre	false	character varying	sq_gestion_perso
carnet_identidad	carnet_identidad	false	character varying	sq_gestion_perso
id_estado_civil	id_estado_civil	false	integer	sq_gestion_perso
id_peloton	id_peloton	false	integer	sq_gestion_perso

Campo unión izquierdo: -Seleccione- = Campo unión derecho: -Seleccione-
Asociar

Uniones:

Condición
tb_destudiante.id_persona = tb_dpersona_data_plus.id_persona

Aceptar Cancelar

Prototipo Modificar objeto del negocio nomenclador.



Modificar objeto del negocio

BD_SGU_pregrado
 v_matriculados_pregrado
 tb_dtramite_docente
 tb_ksituacion_alumno_ayudante
 tb_dsolicitud_alumno_ayudante
 view_procesos_bloqueados
 tb_dtramite_docente_ingreso
 tb_nperiodo_instrucción
 tb_rtipo_tamite_situacion_escolar
 tb_destudiante
 tb_ncausa_evaluacion
 tb_nhuerfano
 tb_rpersona_limitacion_salud
 tb_dpersona_bloqueado
 tb_ngrupo_sanguineo
 v_matriculados_pregrado
 tb_dtramite_docente
 tb_ksituacion_alumno_ayudante
 tb_dsolicitud_alumno_ayudante
 view_procesos_bloqueados
 tb_dtramite_docente_ingreso
 tb_nperiodo_instrucción
 tb_rtipo_tamite_situacion_escolar
 tb_ncausa_evaluacion
 tb_nhuerfano
 tb_rpersona_limitacion_salud
 tb_dpersona_bloqueado
 tb_ngrupo_sanguineo
 tb_dtramite_docente
 tb_dtramite_docente_ingreso
 tb_dpersona_bloqueado
 tb_ksituacion_alumno_ayudante
 tb_nhuerfano
 tb_dtramite_docente
 tb_ksituacion_alumno_ayudante
 tb_dsolicitud_alumno_ayudante
 tb_nperiodo_instrucción
 tb_rtipo_tamite_situacion_escolar
 tb_nperiodo_instrucción
 v_matriculados_pregrado
 tb_dtramite_docente
 tb_nhuerfano
 tb_dpersona_bloqueado
 tb_ngrupo_sanguineo
 tb_dpersona_bloqueado

Nombre del objeto:* est

Nombre visual:* id_estructura

Campo:* id_estructura

Nomenclador

Origen de datos: * n.fn_obtener_estructuras()

Campo visual: * nombre_estructura

Campo principal: * id_estructura

Alias: *

Asociar

Propiedades:

Propiedad	Nombre visual	Nomenclador	Tipo de datos	Referencia
primer_nombre	primer_nombre	false	character varying	sq_gestion_perso
carnet_identidad	carnet_identidad	false	character varying	sq_gestion_perso
id_estado_civil	id_estado_civil	false	integer	sq_gestion_perso
id_peloton	id_peloton	false	integer	sq_gestion_perso

Campo unión izquierdo: -Seleccione-

Campo unión derecho: -Seleccione-

Asociar

Uniones:

Condición

tb_destudiante.id_persona = tb_dpersona_data_plus.id_persona

Aceptar Cancelar

Prototipo modificar objeto del negocio nomenclador1.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre del objeto	character varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.
Nombre visual	character varying	Admite un rango de 1 a 30 caracteres con valores alfanuméricos y guión bajo. Campo obligatorio.



Campo	character varying	Permite seleccionar los campos que pertenecerán al objeto del negocio que desea modificar. Campo obligatorio.
Nomenclador	booleano	Se habilita cuando el Campo seleccionado referencia un nomenclador.
Origen de datos	character varying	Se muestra cuando no existe una referencia al nomenclador en la base de datos. Representa la tabla donde se encuentra el nomenclador o una función para obtener la información del nomenclador. Admite el nombre de una función válida o el nombre de una tabla (valores alfanuméricos y guión bajo). Campo obligatorio.
Campo visual	character varying	Permite seleccionar el campo visual del nomenclador. Si no existe una referencia al nomenclador en la base de datos se muestra un cuadro de texto para obtener información del nomenclador. Admite letras y guión bajo. Campo obligatorio.
Campo principal	character varying	Permite seleccionar el campo principal del nomenclador. Si no existe una referencia al nomenclador en la base de datos se muestra un cuadro de texto para obtener información del nomenclador. Admite letras y guión bajo. Campo obligatorio.
Alias	character varying	Permite tratar una función como una tabla de la base de datos. Admite letras y guión bajo. Se muestra cuando en el campo Origen de los datos se introduce una función. Campo obligatorio.
Campo unión izquierdo	character varying	Indica que se debe seleccionar el campo por el que se debe unir una tabla con otra en caso de que se



		deseen añadir campos de otras tablas.
Campo unión derecho	character varying	Indica que se debe seleccionar el campo por el que se debe unir una tabla con otra en caso de que se deseen añadir campos de otras tablas.
Observaciones	<ol style="list-style-type: none"> 1. En caso que el objeto del negocio ya exista se muestra un mensaje de error “<i>El elemento ya existe</i>”. 2. En caso de cancelar la acción se muestra un mensaje de advertencia “<i>¿Está seguro de realizar la acción?</i>” 3. En caso de que se introduzcan datos incorrectos se muestra un mensaje en rojo encima del nombre del campo indicando el tipo de error. 4. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: “<i>Campo requerido</i>” encima del nombre del campo que debe ser llenado obligatorio. 5. Si se trata de introducir más de los caracteres permitidos para los campos, el sistema no permite seguir introduciéndolos. 6. Interactúa con esta acción el administrador de base de datos. 	

Tabla 16. Especificación de requisito. Ver detalles de objeto del negocio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF8.]	Ver detalles de objeto del negocio.	Permite ver detalles de un objeto del negocio seleccionado por el usuario. Se muestra un listado con las Propiedades del objeto seleccionado, Nombre visual, Nomenclador, Tipo de dato y Referencia.	Baja	Baja
Prototipo				



Ver detalles de objeto del negocio				
Propiedad	Nombre visual	Nomenclador	Tipo de dato	Referencia
lugar de nacimiento	nacimiento	false	character varying	sq_gestion_personal_tb_destudiante
discapacitado	discapacitado	false	boolean	sq_gestion_personal_tb_destudiante
id_anno_academico	id_anno_academico	false	integer	sq_gestion_personal_tb_destudiante
id_persona	id_persona	false	character varying	sq_gestion_personal_tb_destudiante
id_asignatura	id_asignatura	false	integer	sq_akademos_mig_resumen_notas

Cerrar

Prototipo Ver detalles de objeto del negocio.

Campos	Tipos de Datos	Reglas o Restricciones
Observaciones	1. Interactúa con esta acción el administrador de base de datos.	

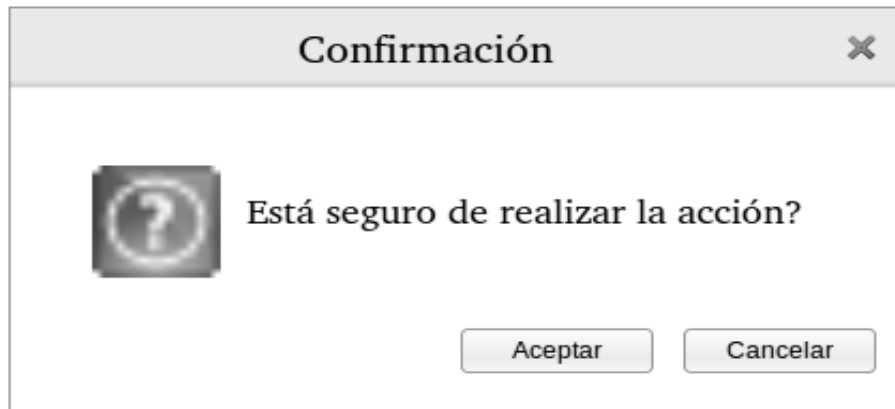
Tabla 17. Especificación de requisito. Eliminar objeto del negocio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF9.]	Eliminar objeto del negocio.	Permite eliminar un objeto del negocio. El usuario selecciona un objeto del negocio en el listado de objetos de una configuración, seguidamente selecciona la opción Eliminar objeto del negocio y se muestra el mensaje de advertencia <i>¿Está seguro de realizar la acción?</i> Seguidamente el usuario presiona el botón Aceptar y se muestra el mensaje <i>“El</i>	Media	Media

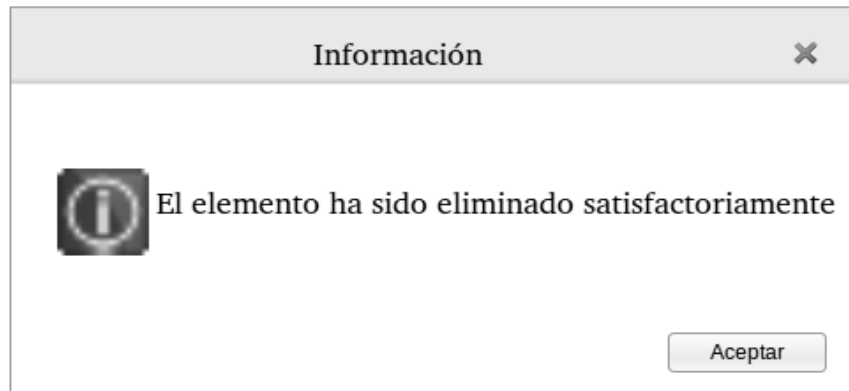


elemento ha sido eliminado satisfactoriamente".

Prototipo



Prototipo Mensaje de confirmación.



Prototipo Mensaje de información.

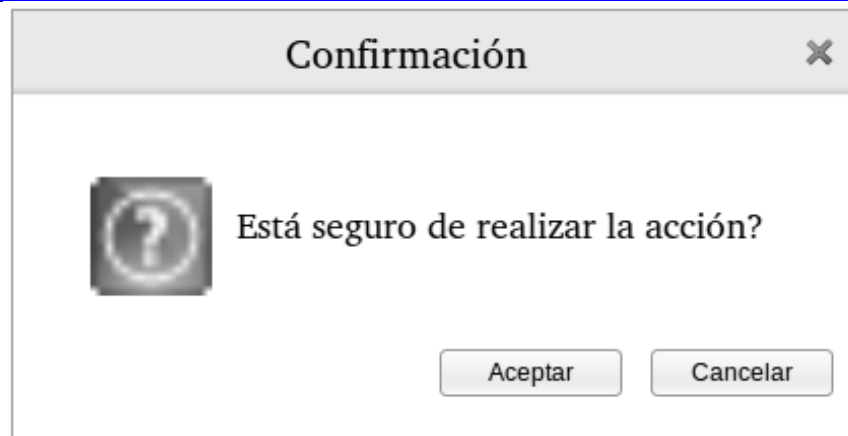


Campos	Tipos de Datos	Reglas o Restricciones
Observaciones	1. Interactúa con esta acción el administrador de base de datos.	

Tabla 18. Especificación de requisito. Eliminar propiedad.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF10.]	Eliminar propiedad.	Permite eliminar una propiedad. El usuario selecciona la propiedad que desea eliminar en el listado de propiedades de un objeto del negocio, presiona el ícono Eliminar y se muestra el mensaje <i>¿“Está seguro de realizar la acción”?</i> Seguidamente el usuario presiona el botón Aceptar y se muestra el mensaje <i>“El elemento ha sido eliminado satisfactoriamente”</i> .	Baja	Baja

Prototipo



Prototipo Mensaje de confirmación.



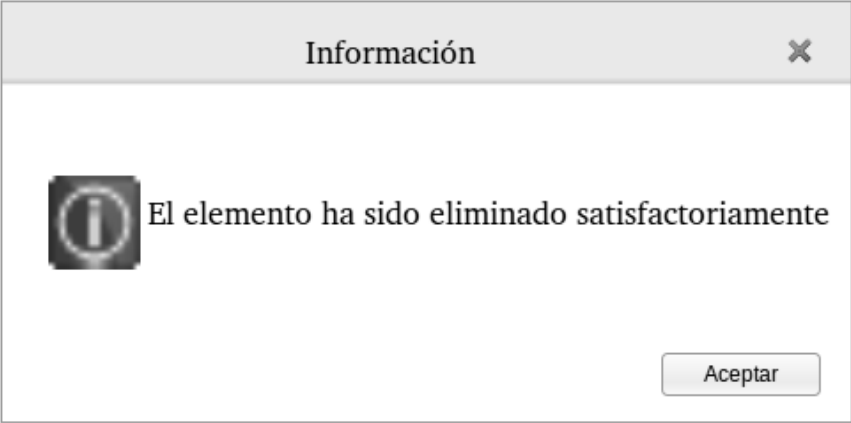
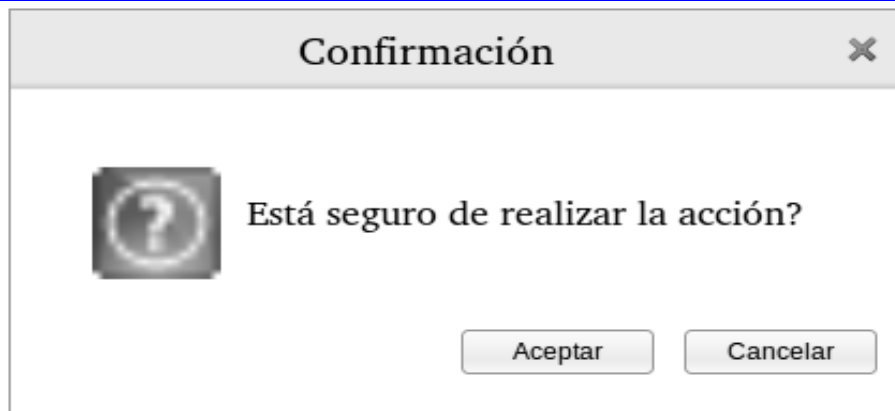
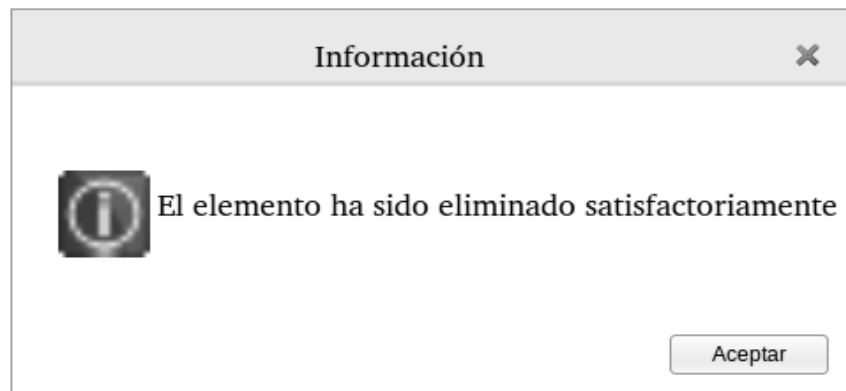
 <p style="text-align: center;">Prototipo Mensaje de información.</p>		
Campos	Tipos de Datos	Reglas o Restricciones
Observaciones		
	1. Interactúa con esta acción el administrador de base de datos.	

Tabla 19. Especificación de requisito. Eliminar unión de tablas.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF11.]	Eliminar unión de tablas.	Permite eliminar la unión de dos tablas de la base de datos. El usuario selecciona una unión en el listado de las uniones que se han realizado, seguidamente selecciona la opción Eliminar y se muestra el mensaje de advertencia <i>¿Está seguro de realizar la acción?</i> Seguidamente el usuario presiona el botón Aceptar y se muestra el mensaje <i>“El elemento ha sido eliminado satisfactoriamente”</i> .	Baja	Baja
Prototipo				



Prototipo Mensaje de confirmación.



Prototipo Mensaje de información.

Campos	Tipos de Datos	Reglas o Restricciones
Observaciones	1. Interactúa con esta acción el administrador de base de datos.	

Tabla 20. Especificación de requisito. Mostrar reglas.



Nº	Nombre	Descripción	Complejidad	Prioridad para cliente																
[RF12.]	Mostrar reglas.	Permite mostrar un listado con las reglas existentes. Se muestra la opción de Crear una regla, así como Modificar, Eliminar y Ejecutar una regla seleccionada. El usuario puede abandonar la ventana presionando el botón Salir o por la cruz que se encuentra en la parte superior derecha.	Media	Media																
Prototipo																				
<table border="1"> <thead> <tr> <th>Nombre</th> <th>Tipo de regla</th> <th>Tipo de acción</th> <th>Configuración</th> </tr> </thead> <tbody> <tr> <td>aa</td> <td>Negocio</td> <td>Selección</td> <td>pregrado</td> </tr> <tr> <td>kk</td> <td>Negocio</td> <td>Selección</td> <td>pregrado</td> </tr> <tr> <td>fdg</td> <td>Negocio</td> <td>Selección</td> <td>prueba</td> </tr> </tbody> </table>					Nombre	Tipo de regla	Tipo de acción	Configuración	aa	Negocio	Selección	pregrado	kk	Negocio	Selección	pregrado	fdg	Negocio	Selección	prueba
Nombre	Tipo de regla	Tipo de acción	Configuración																	
aa	Negocio	Selección	pregrado																	
kk	Negocio	Selección	pregrado																	
fdg	Negocio	Selección	prueba																	
Prototipo Mostrar reglas.																				
Campos	Tipos de Datos	Reglas o Restricciones																		
Observaciones	1. Interactúa con esta acción el administrador de base de datos.																			

Tabla 21. Especificación de requisito. Crear regla.



Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF13.]	Crear regla.	<p>Permite crear una regla. El usuario selecciona un tipo de regla y la asocia a un tipo de acción. Las acciones que se pueden realizar son: Seleccionar, Actualizar, Eliminar e Insertar en caso de que se seleccione el tipo de regla del Negocio. En caso que se seleccione el tipo de regla de Rectificación se puede realizar la acción de Transformar. El panel de condición se habilita cuando se selecciona cualquier tipo de acción menos Insertar, esto indica que las acciones que se realicen sobre el valor de las propiedades seleccionadas podrán estar sujetas a determinadas condiciones. Cuando se escoge la acción Seleccionar se habilitan los paneles de Agrupar y Ordenar, esto quiere decir que el usuario puede agrupar información de un objeto, así como ordenar información por una propiedad determinada, el sistema brinda la posibilidad de ordenar la información de forma ascendente o descendente. EL Having se habilita cuando se especifica que se desea agrupar información por determinada propiedad y permite especificar condiciones sobre la información agrupada. Una vez que el usuario llena todos los campos para crear una regla puede presionar el botón Aceptar para guardar la regla que creó o Cancelar para deshacer la acción de crear regla.</p>	Alta	Alta
Prototipo				



Crear regla

Nombre: *

Tipo de regla

- Negocio
- Validación
- Rectificación

Acción

- Seleccionar
- Insertar
- Actualizar
- Transformar
- Eliminar

Seleccionar

Configuración: * Objeto: *

-Seleccione- -Seleccione- + - -*

Función agregada: Propiedad: *

-Seleccione- -Seleccione- + - -*

DISTINCT TODO

Condición

Configuración: Objeto: () & / !

-Seleccione- -Seleccione-

Propiedad: Signo: Propiedad: + - -*

-Seleccione- -Seleccione- -Seleccione-

Agrupación

Objeto: + - -*

-Seleccione-

Ordenar

Objeto: + - -*

-Seleccione-

Ascendente Descendente

Having () & / !

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad:

-Seleccione- -Seleccione- -Seleccione- -Seleccione- + - -*

Aceptar Cancelar

Prototipo Crear regla_Seleccionar.



- X

Crear regla

Nombre: *

Tipo de regla

Negocio

Validación

Rectificación

Acción

Seleccionar Insertar

Actualizar Transformar

Eliminar

Actualizar

Configuración: * Objeto: *

 + - -*

Propiedad: * IGUAL A Propiedad: *

 + - -*

entrada manual

ACTUALIZAR

Condición

Configuración: Objeto: () & / !

Propiedad: Signo: Propiedad: + - -*

entrada manual

Agrupación

Objeto: + - -*

Ordenar

Objeto: + - -*

Ascendente Descendente

Having

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad: + - -*

entrada manual



Crear regla

Nombre: *

Tipo de regla

- Negocio
- Validación
- Rectificación

Acción

- Seleccionar
- Insertar
- Actualizar
- Transformar
- Eliminar

Eliminar

Configuración: * Objeto: *

-Seleccione- -Seleccione- + -

Condición

Configuración: Objeto: () & / !

-Seleccione- -Seleccione-

Propiedad: Signo: Propiedad: + - -*

-Seleccione- -Seleccione- -Seleccione-

entrada manual

Agrupación

Objeto: + - -*

-Seleccione-

Ordenar

Objeto: + - -*

-Seleccione-

Ascendente Descendente

Having

() & / !

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad: + - -*

-Seleccione- -Seleccione- -Seleccione- -Seleccione-

entrada manual

Aceptar Cancelar

Prototipo Crear regla_Eliminar.



Crear regla

Nombre: *

Tipo de regla

- Negocio
- Validación
- Rectificación

Acción

- Seleccionar
- Insertar
- Actualizar
- Transformar
- Eliminar

Insertar

Configuración: * Objeto: *

-Seleccione- -Seleccione- + - -*

Propiedad: * Igual a Propiedad: *

-Seleccione- IGUAL A -Seleccione- + - -*

entrada manual

Condición

Configuración: Objeto:

-Seleccione- -Seleccione- () & / !

Propiedad: Signo: Propiedad:

-Seleccione- -Seleccione- -Seleccione- + - -*

entrada manual

Having

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad:

-Seleccione- -Seleccione- -Seleccione- -Seleccione- + - -*

entrada manual

Aceptar Cancelar

INSERTAR



Crear regla - X

Nombre: *

Tipo de regla

Negocio
 Validación
 Rectificación

Acción

Seleccionar Insertar
 Actualizar Transformar
 Eliminar

Transformar

Configuración: * Objeto: *
-Seleccione- -Seleccione- + - -*

Propiedad: *
-Seleccione- + - -*

Condición

Configuración: Objeto: () & / !
-Seleccione- -Seleccione-

Propiedad: Signo: Propiedad: + - -*
 -Seleccione- -Seleccione- -Seleccione-
 entrada manual

Agrupación

Objeto: + - -*
-Seleccione-

Ordenar

Objeto: + - -*
-Seleccione-
 Ascendente Descendente

Having () & / !

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad: + - -*
-Seleccione- -Seleccione- -Seleccione- -Seleccione-
 entrada manual

Prototipo Crear regla_Transformar.



- X

Crear regla

Nombre: *

Tipo de regla

Negocio

Validación

Rectificación

Acción

Seleccionar Insertar

Actualizar Transformar

Eliminar

Actualizar

Actualizar

Configuración: * Objeto: *

 + - -*

Propiedad: * Igual A Propiedad: *

 IGUAL A + - -*

Condición

Configuración: Objeto:

 () & / !

Propiedad: Signo: Propiedad:

 + - -*

entrada manual

Agrupación

Objeto:

 + - -*

Ordenar

Objeto:

 + - -*

Ascendente Descendente

Having

() & / !

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad:

 + - -*

entrada manual

Prototipo Crear regla Validación.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre de regla	character varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.



Tipo de regla de negocio	booleano	Permite chequear que un conjunto de datos cumplen con determinadas restricciones del negocio.
Tipo de regla de validación	booleano	Representa determinadas funciones de base de datos que permiten darle tratamiento a las propiedades, por ejemplo eliminar espacios al inicio y al final de una propiedad.
Tipo de regla de rectificación	booleano	Permite transformar un conjunto de datos en propiedades de un objeto.
Seleccionar	booleano	Permite seleccionar las propiedades de un objeto del negocio perteneciente a una configuración seleccionada. Permite seleccionar funciones agregadas según el tipo de propiedad, seleccionar todas las propiedades, así como no mostrar información repetida.
Actualizar	booleano	Permite actualizar el valor de las propiedades de determinado objeto perteneciente a una configuración seleccionada. El valor de las propiedades se puede actualizar a partir del valor que contenga otra propiedad o asignándole un valor manualmente.
Eliminar	booleano	Permite eliminar valores asociados a un objeto del negocio perteneciente a una configuración, donde la propiedad cumpla con determinada condición.
Insertar	booleano	Permite insertar nuevos valores a determinada propiedad de un objeto del negocio perteneciente a una configuración seleccionada. Los valores de las propiedades pueden insertarse a partir del valor que contenga otra propiedad o asignándole un valor manualmente.
Transformar	booleano	Permite convertir información de un objeto en las propiedades de otro objeto perteneciente a una configuración seleccionada.
Configuración	character varying	Permite seleccionar determinada configuración. Campo



		obligatorio.
Objeto	character varying	Permite seleccionar un objeto perteneciente a una configuración previamente seleccionada. Campo obligatorio.
Función agregada	character varying	Se aplican a las propiedades para obtener valores según el tipo de propiedad. Las funciones agregadas son: AVG (promedio), SUM (Suma), MIN (Mínimo), MAX (Máximo) y COUNT (Cantidad).
Propiedad	character varying	Representa las propiedades que se pueden seleccionar de un objeto. Campo obligatorio.
Distinct	booleano	Permite que no se muestren datos repetidos en un conjunto de información.
Todo	booleano	Esta opción permite seleccionar todas las propiedades de un objeto del negocio determinado.
Entrada manual	booleano	Permite entrar datos manualmente.
Signo	character varying	Permite comparar el valor de una propiedad con otra o con un valor entrado manualmente. Los signos pueden ser: Igual, Entre, En, No en, Entre, No entre, Menor, Menor o igual, Mayor, Mayor o igual, Es, No es, Contiene, No contiene y Distinto.
Valor	character varying	Se habilita en caso de que la propiedad sea booleana, en este caso se compara la propiedad con un valor que puede ser true o false, lo mismo sucede cuando la propiedad es una fecha, en este caso el valor es una fecha. En caso de que la propiedad sea un nomenclador permite que se compare con los valores posibles para ese nomenclador en la base de datos.
Objeto/Propiedad	character varying	Permite seleccionar en un solo paso la propiedad de un objeto determinado.



Observaciones	<p>1. En caso de que se introduzcan datos incorrectos para el nombre de la regla (solo se admiten letras y guión bajo) se muestra un mensaje en rojo encima del nombre del campo indicando el tipo de error.</p> <p>2. Si se trata de introducir más de los caracteres permitidos para el nombre de la regla, el sistema no permite seguir introduciéndolos.</p> <p>3. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”.</p> <p>4. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: “Campo requerido” encima del nombre del campo que debe ser llenado obligatorio.</p> <p>5. Cuando se habilita la condición se pueden utilizar los paréntesis () para agrupar información, los operadores AND (&) y OR (/) para concatenar dos expresiones, así como el operador de negación (!).</p> <p>6. En caso que el usuario deje paréntesis no balanceados se muestra en forma de ventana el mensaje: “Paréntesis desbalanceados”.</p> <p>7. El sistema no permite construir una expresión de forma incorrecta.</p> <p>8. Interactúa con esta acción el administrador de base de datos.</p>
----------------------	---

Tabla 22. Especificación de requisito. Modificar_regla.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF14.]	Modificar regla.	Permite modificar una regla. El usuario selecciona un tipo de regla en el listado de reglas existentes y selecciona la acción de modificar. En dependencia del tipo de regla seleccionado se muestran una serie de campos a modificar. Una vez que el usuario llena todos los campos para modificar una regla puede presionar el botón Aceptar para guardar los cambios realizados o Cancelar para deshacer la acción de modificar regla.	Baja	Alta
Prototipo				



Modificar regla

Nombre: *
regla_valid

SELECCIONAR

Seleccionar

Configuración: * otra_configur
Objeto: * persona + - -*
Función agregada: -Selecione-
Propiedad: * -Selecione- + - -*
 DISTINCT TODO

Condición

Configuración: otra_configur
Objeto: persona () & / !
Propiedad: fecha_nacimiento
Signo: igual
Valor: 30-3-1990 + - -*

Agrupación

Objeto: -Selecione- + - -*

Ordenar

Objeto: -Selecione- + - -*
 Ascendente Descendente

Having

() & / !
Función agregada: -Selecione-
Objeto/Propiedad: -Selecione-
Signo: -Selecione-
Función agregada: -Selecione-
Objeto/Propiedad: + - -*

Aceptar Cancelar

Prototipo Modificar regla_Seleccionar.



Modificar regla _ X

Nombre: *

Actualizar

Configuración: * **Objeto: ***

Propiedad: * **IGUAL A** **Propiedad: ***

entrada manual

Condición

Configuración: **Objeto:**

Propiedad: **Signo:** **Propiedad:**

entrada manual

Having

Función agregada: **Objeto/Propiedad:** **Signo:** **Función agregada:** **Objeto/Propiedad:**

entrada manual

Agrupación

Objeto:

Ordenar

Objeto:

Ascendente Descendente

Prototipo Modificar regla_Actualizar.



Modificar regla - X

Nombre: *

Eliminar

Configuración: * Objeto: *

Condición

Configuración: Objeto:

Propiedad: Signo: Propiedad:

entrada manual

Agrupación

Objeto:

Ordenar

Objeto:

Ascendente Descendente

Having

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad:

entrada manual

Prototipo Modificar regla_Eliminar.



Modificar regla - X

Nombre: *

INSERTAR

Insertar

Configuración: * Objeto: * + - -*

Propiedad: * IGUAL A + - -*

entrada manual

Condición

Configuración: Objeto: () & / !

Propiedad: Signo: Propiedad: + - -*

entrada manual

Agrupación

Objeto: + - -*

Ordenar

Objeto: + - -*

Ascendente Descendente

Having

() & / !

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad:

entrada manual

Prototipo Modificar regla_Insertar.



Modificar regla _ X

Nombre: *

Transformar

Configuración: * Objeto: * + - -*

Propiedad: * + - -*

Condición

Configuración: Objeto: () & / !

Propiedad: Signo: Propiedad: + - -*

entrada manual

Having

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad: + - -*

entrada manual

TRANSFORMAR

Prototipo Modificar regla_Transformar.



Modificar regla
_ X

Nombre: *

ACTUALIZAR

EStpublic

SET

EStpublic.primer_apellido Igual Espacios_Inicio_Fin.primer_apellido

Actualizar

Configuración: * Objeto: * + - *

Propiedad: * IGUAL A + - *

Condición

Configuración: Objeto: () & / !

Propiedad: Signo: Propiedad: + - *

entrada manual

Agrupación

Objeto: + - *

Having

Función agregada: Objeto/Propiedad: Signo: Función agregada: Objeto/Propiedad: + - *

entrada manual

Ordenar

Objeto: + - *

Ascendente Descendente

Prototipo Modificar regla_Validar.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre de regla	character varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.



Configuración	character varying	Permite seleccionar determinada configuración. Campo obligatorio.
Objeto	character varying	Permite seleccionar un objeto perteneciente a una configuración previamente seleccionada. Campo obligatorio.
Función agregada	character varying	Se pueden aplicar a las propiedades para obtener valores según el tipo de propiedad. Las funciones agregadas son: AVG (promedio), SUM (Suma), MIN (Mínimo), MAX (Máximo) y COUNT (Cantidad).
Propiedad	character varying	Representa las propiedades que se pueden seleccionar de un objeto. Campo obligatorio.
Distinct	booleano	Permite que no se muestren datos repetidos en un conjunto de información.
Todo	booleano	Esta opción permite seleccionar todas las propiedades de un objeto del negocio determinado.
Entrada manual	booleano	Permite entrar datos manualmente.
Signo	character varying	Permite comparar el valor de una propiedad con otra o con un valor entrado manualmente. Los signos pueden ser: Igual, Entre, En, No en, Entre, No entre, Menor, Menor o igual, Mayor, Mayor o igual, Es, No es, Contiene, No contiene y Distinto.
Valor	character varying	Se habilita en caso de que la propiedad sea booleana, en este caso se compara la propiedad con un valor que puede ser true o false, lo mismo sucede cuando la propiedad es una fecha, en este caso el valor es una fecha. En caso de que la propiedad sea un nomenclador permite que se compare con los valores posibles para ese nomenclador en la base de datos.



Objeto/Propiedad	character varying	Permite seleccionar en un solo paso la propiedad de un objeto determinado.
Observaciones	<ol style="list-style-type: none"> 1. En caso de que se introduzcan datos incorrectos para el nombre de la regla (solo se admiten letras y guión bajo) se muestra un mensaje en rojo encima del nombre del campo indicando el tipo de error. 2. Si se trata de introducir más de los caracteres permitidos para el nombre de la regla, el sistema no permite seguir introduciéndolos. 3. En caso de cancelar la acción se muestra el mensaje de advertencia: “¿Está seguro de realizar la acción?”. 4. En caso que se deje un campo de los obligatorios vacío se muestra el mensaje de error de color rojo: “Campo requerido”, encima del nombre del campo que debe ser llenado obligatorio. 5. Cuando se habilita la condición se pueden utilizar los paréntesis () para agrupar información, los operadores AND (&) y OR (/) para concatenar dos expresiones, así como el operador de negación (!). 6. En caso que el usuario deje paréntesis no balanceados se muestra en forma de ventana el mensaje: “Paréntesis desbalanceados”. 7. El sistema no permite construir una expresión de forma incorrecta. 8. Interactúa con esta acción el administrador de base de datos. 	

Tabla 23. Especificación de requisito. Eliminar regla.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF15.]	Eliminar regla.	Permite eliminar una regla seleccionada. El usuario selecciona una regla en el listado de las reglas existentes, seguidamente selecciona la opción Eliminar y se muestra el mensaje de advertencia ¿“Está seguro de realizar la acción”? El usuario presiona el botón Aceptar y se muestra el mensaje “El elemento ha sido eliminado satisfactoriamente”.	Baja	Media
Prototipo				



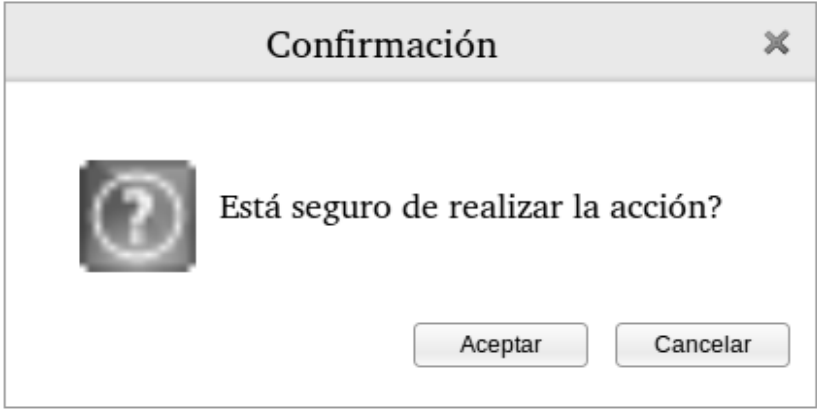
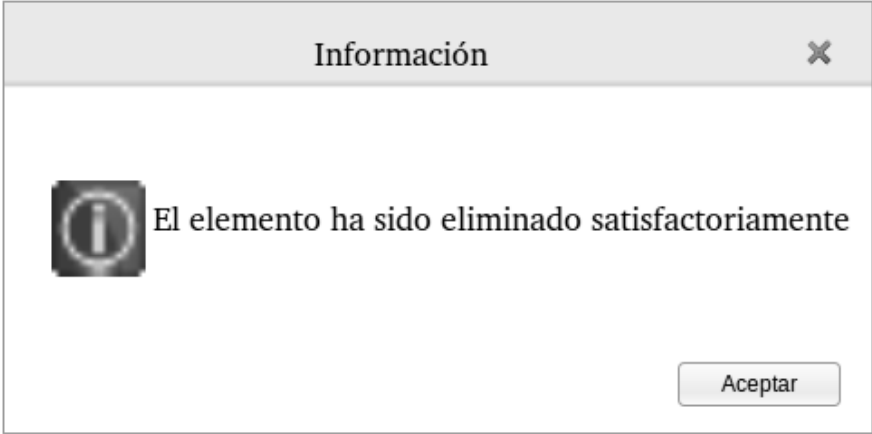
	 <p>Prototipo Mensaje de confirmación.</p>  <p>Prototipo Mensaje de información.</p>	
Campos	Tipos de Datos	Reglas o Restricciones
Observaciones	1. Interactúa con esta acción el administrador de base de datos.	

Tabla 24. Especificación de requisito. Ejecutar regla.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------



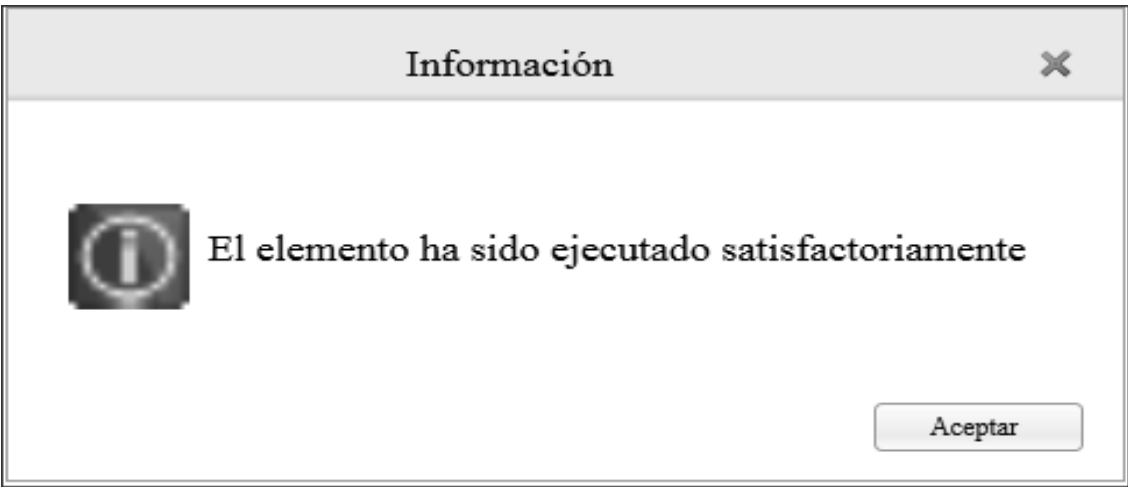
[RF16.]	Ejecutar regla.	Permite que se ejecute una regla seleccionada. El usuario selecciona una regla y presiona el ícono de Ejecutar. Una vez ejecutada la regla se muestra el mensaje de información: "El elemento ha sido ejecutado satisfactoriamente".	Alta	Alta
Prototipo				
				
Prototipo mensaje de información.				
Campos	Tipos de Datos		Reglas o Restricciones	
Observaciones	<ol style="list-style-type: none"> 1. Interactúa con esta acción el administrador de base de datos. 2. En caso que no se pueda ejecutar la regla seleccionada se muestra el mensaje de error: "No se pudo ejecutar la regla". 			

Tabla 25. Especificación de requisito. Mostrar tareas.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF17.]	Mostrar tareas.	Permite mostrar un listado de las tareas	Media	Media



	existentes. Además se muestra la opción de Crear, así como las opciones de Modificar, Eliminar y Ver detalles de una tarea seleccionada.		
Prototipo			
<p style="text-align: center;">Prototipo Mostrar tareas.</p>			
Campos	Tipos de Datos	Reglas o Restricciones	
Observaciones	1. Interactúa con esta acción el administrador de base de datos.		

Tabla 26. Especificación de requisito. Modificar tarea.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF19.]	Modificar tarea.	Permite modificar una tarea. Se muestran los campos a modificar y una vez modificada la tarea se muestra el	Baja	Alta



mensaje de información: "El elemento ha sido modificado satisfactoriamente".

Prototipo

Prototipo Modificar tarea.

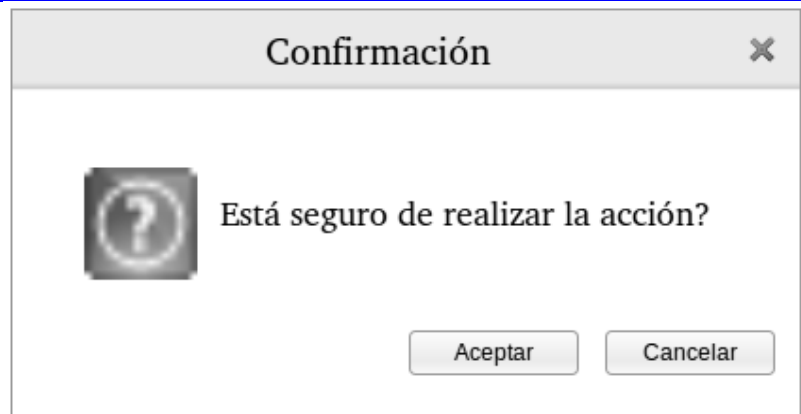
Campos	Tipos de Datos	Reglas o Restricciones
Nombre	character varying	Admite un rango de 1 a 30 caracteres con letras y guión bajo. Campo obligatorio.
Fecha ejecución	de character varying	Permite seleccionar la fecha para ejecutar la tarea. Campo obligatorio.
Hora	character varying	Permite seleccionar la hora en que se desea ejecutar la tarea. Campo obligatorio.
Minutos	character varying	Permite especificar los minutos exactos en que se va a ejecutar la tarea. Campo obligatorio.
AM	booleano	Permite seleccionar la sección del día AM, para ejecutar la tarea.



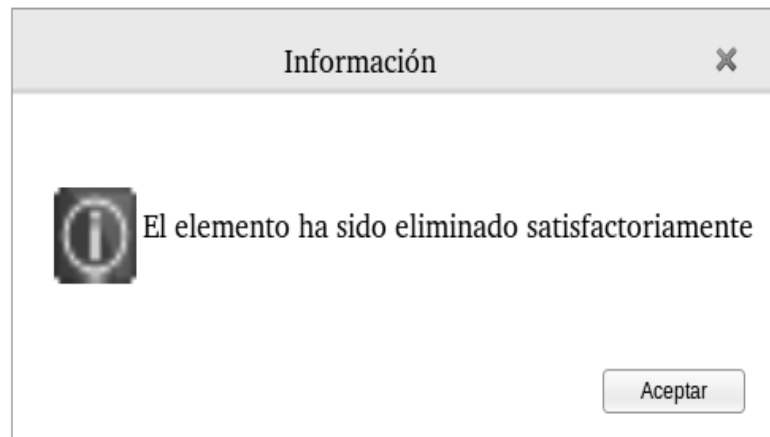
PM	booleano	Permite seleccionar la sección del día PM, para ejecutar la tarea.
Reglas	character varying	Permite asociar un grupo de reglas a la tarea que se está creando. Campo obligatorio.
Observaciones	<p>1. En caso que la tarea ya exista se muestra el mensaje de error: <i>“El elemento ya existe”</i>.</p> <p>2. En caso de cancelar la acción se muestra el mensaje de advertencia: <i>“¿Está seguro de realizar la acción?”</i>.</p> <p>3. En caso de que se introduzcan datos incorrectos (solo se admiten letras y guion bajo) se muestra un mensaje de color rojo encima del nombre del campo indicando el tipo de error.</p> <p>4. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo: <i>“Campo requerido”</i> encima del nombre del campo que debe ser llenado obligatorio.</p> <p>5. Si se trata de introducir más de los caracteres permitidos para los campos el sistema no permite seguir introduciéndolos.</p> <p>6. Interactúa con esta acción el administrador de base de datos.</p>	

Tabla 27. Especificación de requisito. Eliminar tarea.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF20.]	Eliminar tarea.	Permite eliminar una tarea. El usuario selecciona una tarea en el listado de tareas existentes, seguidamente selecciona la opción Eliminar y se muestra el mensaje de advertencia <i>“¿Está seguro de realizar la acción?”</i> . Seguidamente el usuario presiona el botón Aceptar y se muestra el mensaje <i>“El elemento ha sido eliminado satisfactoriamente”</i> .	Baja	Media
Prototipo				



Prototipo Mensaje de confirmación.



Prototipo Mensaje de información.

Campos	Tipos de Datos	Reglas o Restricciones
Observaciones	1. Interactúa con esta acción el administrador de base de datos.	

Tabla 28. Especificación de requisito. Ver detalles de tarea.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------



[RF21.]	Ver detalles de tarea.	Permite ver detalles de una tarea seleccionada por el usuario. Se muestra un listado con el Nombre de las reglas asociadas a la tarea, el Tipo de regla visual, Acción y la Base de datos donde se ejecuta la tarea.	Baja	Baja
Prototipo				
Prototipo Ver detalles de tarea.				
Campos		Tipos de Datos	Reglas o Restricciones	
Observaciones		1. Interactúa con esta acción el administrador de base de datos.		

Tabla 29. Especificación de requisito. Mostrar reportes.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
----	--------	-------------	-------------	------------------------



[RF22.]	Mostrar reportes.	Permite mostrar un listado con la cantidad de reportes generados. Se muestra la opción de Ver detalles de reporte inicial, así como Ver detalles de reporte final y Exportar reporte. Cuando el usuario pasa el cursor por encima del ícono se debe indicar la acción. El usuario puede abandonar la ventana presionando el botón Salir o por la cruz que se encuentra en la parte superior derecha.	Media	Media
Prototipo				
Prototipo Mostrar reportes.				
Campos		Tipos de Datos	Reglas o Restricciones	
Observaciones		1. Interactúa con esta acción el administrador de base de datos.		

Tabla 30. Especificación de requisito. Ver detalles de reporte inicial.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF23.]	Ver detalles de reporte inicial.	Permite ver detalles de un reporte seleccionado por el usuario. Se muestra un conjunto de datos según el reporte que se	Baja	Baja



		haya seleccionado. El usuario puede abandonar la página presionando el botón Salir o por la cruz que se encuentra en la parte superior derecha.																											
Prototipo																													
<table border="1"> <thead> <tr> <th>id_persona</th> <th>id_estructura</th> <th>lugar_nacimiento</th> <th>id_anno_académico</th> <th>id_periodo_lectivo</th> </tr> </thead> <tbody> <tr> <td>2e77f23g7716...</td> <td>347</td> <td>CIUDAD HABANA</td> <td>18</td> <td>19</td> </tr> <tr> <td>099db345r667...</td> <td>347</td> <td>Marianao</td> <td>18</td> <td>19</td> </tr> <tr> <td>afc32267899k...</td> <td>347</td> <td>Pinar del Río</td> <td>18</td> <td>19</td> </tr> <tr> <td>alada18712ca...</td> <td>347</td> <td>CIUDAD HABANA</td> <td>18</td> <td>19</td> </tr> </tbody> </table>					id_persona	id_estructura	lugar_nacimiento	id_anno_académico	id_periodo_lectivo	2e77f23g7716...	347	CIUDAD HABANA	18	19	099db345r667...	347	Marianao	18	19	afc32267899k...	347	Pinar del Río	18	19	alada18712ca...	347	CIUDAD HABANA	18	19
id_persona	id_estructura	lugar_nacimiento	id_anno_académico	id_periodo_lectivo																									
2e77f23g7716...	347	CIUDAD HABANA	18	19																									
099db345r667...	347	Marianao	18	19																									
afc32267899k...	347	Pinar del Río	18	19																									
alada18712ca...	347	CIUDAD HABANA	18	19																									
Prototipo Ver detalles de reporte inicial.																													
Campos		Tipos de Datos	Reglas o Restricciones																										
Observaciones		1. Interactúa con esta acción el administrador de base de datos.																											

Tabla 31. Especificación de requisito. Ver detalles de reporte final.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF24.]	Ver detalles de reporte final.	Permite ver detalles de un reporte seleccionado por el usuario. Se muestra un conjunto de datos según el reporte que se haya seleccionado. El usuario puede abandonar la página presionando el botón Salir o por la cruz que se encuentra en la	Baja	Baja



	parte superior derecha.																											
Prototipo																												
<div style="border: 1px solid gray; padding: 5px;"> <p>Ver detalles de reporte final - X</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>id_persona</th> <th>id_estructura</th> <th>lugar_nacimiento</th> <th>id_anno_académico</th> <th>id_periodo_lectivo</th> </tr> </thead> <tbody> <tr> <td>2e77f23g7716...</td> <td>347</td> <td>CIUDAD HABANA</td> <td>18</td> <td>19</td> </tr> <tr> <td>099db345r667...</td> <td>347</td> <td>Marianao</td> <td>18</td> <td>19</td> </tr> <tr> <td>afc32267899k...</td> <td>347</td> <td>Pinar del Río</td> <td>18</td> <td>19</td> </tr> <tr> <td>alada18712ca...</td> <td>347</td> <td>CIUDAD HABANA</td> <td>18</td> <td>19</td> </tr> </tbody> </table> <p style="text-align: right; margin-top: 10px;"><input type="button" value="Salir"/></p> </div>				id_persona	id_estructura	lugar_nacimiento	id_anno_académico	id_periodo_lectivo	2e77f23g7716...	347	CIUDAD HABANA	18	19	099db345r667...	347	Marianao	18	19	afc32267899k...	347	Pinar del Río	18	19	alada18712ca...	347	CIUDAD HABANA	18	19
id_persona	id_estructura	lugar_nacimiento	id_anno_académico	id_periodo_lectivo																								
2e77f23g7716...	347	CIUDAD HABANA	18	19																								
099db345r667...	347	Marianao	18	19																								
afc32267899k...	347	Pinar del Río	18	19																								
alada18712ca...	347	CIUDAD HABANA	18	19																								
Prototipo Detalles de reporte final.																												
Campos	Tipos de Datos	Reglas o Restricciones																										
Observaciones	<ol style="list-style-type: none"> 1. En caso que se seleccione un reporte donde el tipo de acción realizada sea de Selección la información que se muestra en el reporte final coincidirá con la información mostrada en el reporte inicial. 2. Interactúa con esta acción el administrador de base de datos. 																											

Tabla 32. Casos de prueba de caja blanca. RF1.

PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_SGU-PD-MC: RFR1
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: No tiene entrada, se espera el listado de todas las configuraciones.	
Datos de entrada:	No tiene.
Tipo de dato esperado:	Object [[]].



<p>Función de evaluación:</p> <pre> public void test0btenerConfiguraciones() throws Exception { System.out.println("ObtenerConfiguraciones"); CConfiguracion instance = CConfiguracion.getConfiguracion(); Object[][] expResult = null; Object[][] result = instance.ObtenerConfiguraciones(); expResult=result; assertEquals(expResult, result); } </pre>
<p>Evaluación del caso de prueba: Satisfactorio.</p>

Tabla 33. Casos de prueba de caja blanca. RF3.

PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_SGU-PD-MC: RFR3.
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: Los datos de entrada serán los atributos y el identificador de la configuración a modificar.	
Datos de entrada:	Objeto nulo, identificador igual a 0
Tipo de dato esperado:	Vacío.
<p>Función de evaluación:</p> <pre> public void testActualizarConfiguracion() throws Exception { System.out.println("ActualizarConfiguracion"); int id = 0; Configuracion cfg = null; CConfiguracion instance = CConfiguracion.getConfiguracion(); instance.ActualizarConfiguracion(id, cfg); } </pre>	
Evaluación del caso de prueba: Satisfactorio.	

Tabla 34. Casos de prueba de caja blanca. RF4.



PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_PRE_PUCPCB-EC: RFR4.
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: El dato de entrada será el identificador de la configuración a eliminar.	
Datos de entrada:	Identificador igual a 0
Tipo de dato esperado:	Vacío.
Función de evaluación:	
<pre> public void testEliminarConfiguracion() throws Exception { System.out.println("EliminarConfiguracion"); int id = 0; CConfiguracion instance = CConfiguracion.getConfiguracion(); instance.EliminarConfiguracion(id); } </pre>	
Evaluación del caso de prueba: Satisfactorio.	

Tabla 35. Casos de prueba de caja blanca. RF17.

PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_SGU-PD-MT: RFR17.
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: No tiene entrada, se espera el listado de todas las tareas.	
Datos de entrada:	No tiene.
Tipo de dato esperado:	Object [].
Función de evaluación:	



```
public void testObtenerTareas() throws Exception {
    System.out.println("ObtenerTareas");
    CTarea instance = new CTarea();
    Object[][] expResult = null;
    Object[][] result = instance.ObtenerTareas();
    expResult=result;

    assertEquals(expResult, result);

}
```

Evaluación del caso de prueba: Satisfactorio.

Tabla 36. Casos de prueba de caja blanca. RF18.

PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_SGU-PD-CT: RFR18.
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: Los datos de entrada serán los atributos de la tarea a crear.	
Datos de entrada:	Objeto nulo.
Tipo de dato esperado:	Vacío.
Función de evaluación: <pre>public void testAdicionarTarea() throws Exception { System.out.println("AdicionarTarea"); Tarea tarea = null; CTarea instance = new CTarea(); instance.AdicionarTarea(tarea); }</pre>	
Evaluación del caso de prueba: Satisfactorio.	

Tabla 37. Casos de prueba de caja blanca. RF19.



PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_SGU-PD-MT: RFR19.
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: Los datos de entrada serán los atributos y el identificador de la tarea a modificar.	
Datos de entrada:	Objeto nulo, identificador igual a 0
Tipo de dato esperado:	Vacío.
Función de evaluación: <pre>public void testActualizarTarea() throws Exception { System.out.println("ActualizarTarea"); int idTarea = 0; Tarea tarea = null; List<String> reglasEliminar = null; CTarea instance = new CTarea(); instance.ActualizarTarea(idTarea, tarea, reglasEliminar); }</pre>	
Evaluación del caso de prueba: Satisfactorio.	

Tabla 38. Casos de prueba de caja blanca. RF20.

PRUEBA ESTRUCTURAL O CAJA BLANCA	CÓDIGO CASO DE PRUEBA: CENIA_PRE_PUCPCB-EC: RFR20.
Probador: Dayliana Guerra Velázquez, Juan Daniel de la Vega Valenzuela.	
Procedimiento prueba automatizada	
Descripción: El dato de entrada será el identificador de la tarea a eliminar.	
Datos de entrada:	Identificador igual a 0
Tipo de dato esperado:	Vacío.
Función de evaluación:	



```
public void testEliminarTarea() throws Exception {  
    System.out.println("EliminarTarea");  
    int idTarea = 0;  
    CTarea instance = new CTarea();  
    instance.EliminarTarea(idTarea);  
  
}
```

Evaluación del caso de prueba: Satisfactorio.



Tabla 39. Diseño de casos de prueba. Mostrar configuraciones.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos de forma correcta	Mediante este escenario se mostrará al usuario el listado de configuraciones existentes. Además se muestra un conjunto de acciones a desarrollar sobre cada elemento: Crear, Modificar y Eliminar	El sistema muestra el listado de los elementos actualizados hasta la fecha.	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra el listado de las configuraciones existentes y las acciones de contexto que se pueden realizar.
EC 1.2 No existen elementos creados.	Mediante este escenario en caso de que no exista creada ninguna configuración se muestra un listado vacío.	El sistema muestra el listado vacío.	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra el listado de las configuraciones existentes y las acciones de contexto que se pueden realizar.

Tabla 40. Diseño de casos de prueba. Crear configuración.

Escenario	Descripción	Nombre	Ubicación	Nombre	Base de datos	Usuario	Contraseña	Respuesta del	Flujo central
-----------	-------------	--------	-----------	--------	---------------	---------	------------	---------------	---------------



		configuración		servidor			a	sistema	
EC 1.1 Insertar datos correctamente.	Mediante este escenario se introducen datos correctos en el sistema para crear una configuración.	V	V	V	V	V	V	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido creado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de configuraciones existente. -El usuario selecciona el ícono de Crear, llena los datos
		Configuración	/media/JD1/conf/prueba.xml	10.0.0.217	DB_12wdf	sgu_nucleo_user	sfg"fgkjbj		
		V	V	V	V	V	V		
		Configuración_nueva	/home/Janie/cron/e.xml	localhost	BD_sgu_pregrado	sgu_pregrado_user	kg""hk		
		V	V	V	V	V	V		
		Se introduce un texto que tenga como mínimo 1 caracter y como máximo 30, donde las palabras	/media/JD1/conf/prueba.xml	localhost	BD_sgu_residencia	sgu_residencia_user	dggh_!!!1fgh		



		están separadas por guión bajo.							correctamente y presiona el botón Aceptar.
EC 1.2 Insertar elementos repetidos.	Mediante este escenario se introducen datos para crear una configuración que ya existe en el sistema.	I	V	V	V	V	V	El usuario presiona el botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú.
		Configuración	/media/JD1/conf/prueba.xml	localhost	BD_sgu_nucleo	sgu_nucleo_user	asdfgh_dnf g576		-El sistema muestra las acciones de contexto que se pueden realizar y el listado de configuraciones
		V	V	V	V	V	V		
		ConfiguraciónMB	/media/DG1/config/c.xml	localhost	BD_sgu_residencia	sgu_residencia.user	789900**8 88		



									existente. -El usuario selecciona la acción de Crear e introduce datos de una configuración que ya existe en el sistema y presiona el botón Aceptar.
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para crear una configuración.	I	I	V	V	V	I	El sistema muestra encima del nombre del componente que debe ser llenado obligatorio el mensaje de color rojo: "Campo requerido".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y
		Vacío	Vacío	localhost	BD_sgu_residencia	sgu_residencia_user	Vacío		
		V	V	I	I	I	V		
		Config1122_23	/media/JD1/conf/prueba.xml	Vacío	Vacío	Vacío	****bhgjjg.		



									el listado de configuraciones existente. -El usuario selecciona la acción de Crear, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para crear una configuración.	V	I	V	V	V	V	El sistema muestra	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú.
		config_de_prueba	desktop	localhost	BD_sgu_pregrado	sgu_pregrado_user	sosas//nnnn	encima del nombre del componente el mensaje de color rojo: "El directorio no existe".	-El sistema muestra las acciones de
		I	V	V	V	V	V	El sistema no permite la entrada de más de 30 caracteres.	
		asdfghjklpoiuytrew_sdfghjlpodf_ftg	/home/denial/cron/e.xml	localhost	BD_sgu_nucleo	sgu_nucleo_user	043****7712		
		I	V	V	V	V	V	El sistema muestra	



		ad35 *****	/home/Danie l/cron/e.xml	localhost	BD_sgu_reside ncia	sgu_residen cia_user	0152""lolo	encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	contexto que se pueden realizar y el listado de configuraciones existente.
		V	V	I	I	I	V	El sistema muestra	-El usuario
		conf_nueva 66	/media/JD1/ conf/prueba. xml	dghjku vbgh	BD""Mía	dgvelazquez	***** ***	en forma de ventana el mensaje de error: "Los parámetros de conexión no son válidos".	selecciona la acción de Crear, introduce datos incorrectos y presiona el botón Aceptar.
EC	1.5	Mediante este escenario se cancela la creación de la configuración.	NA	NA	NA	NA	NA	El sistema muestra el mensaje de confirmación: "¿Está seguro que desea salir?".	-El usuario una vez en la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra las



									<p>acciones de contexto que se pueden realizar y el listado de configuraciones existente.</p> <p>-El usuario selecciona el ícono de Crear, llena o no los datos y presiona el botón Cancelar.</p>
--	--	--	--	--	--	--	--	--	---

Tabla 41. Diseño de casos de prueba. Modificar configuración.

Escenario	Descripción	Nombre configuración	Ubicación	Nombre servidor	Base de datos	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Modificar datos correctamente.	Mediante este escenario se introducen datos correctos en el sistema para modificar una	V	V	V	V	V	V	El sistema	-El usuario una vez dentro de la vista principal de la aplicación selecciona
		Configuración	/media/JD1/conf/prueba.xml	localhost	BD_sgu_nucleo	sgu_nucleo_user	NHJh"890	actualiza el listado y muestra el mensaje de información	
		V	V	V	V	V	V		
		Configuración_nueva	/home/Janie/cron/e.xml	localhost	BD_sgu_pregado	sgu_pregrado_user	IIIII::pololee		



	configuración.	V	V	V	V	V	V	elemento ha sido modificado satisfactoriamente".	Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración, selecciona la acción de Modificar, llena los datos correctamente y presiona el botón Aceptar.
		Se introduce un texto que tenga como mínimo 1 caracter y como máximo 30, donde las palabras están separadas por guión bajo.	/media/JD1/conf/prueba.xml	localhost	BD_sgu_residencia	sgu_residencia_user	1234567890		
EC 1.2	Mediante este escenario se	I	V	V	V	V	V	El usuario presiona el	-El usuario una vez dentro de



elementos repetidos.	introducen datos para modificar una configuración que ya existe en el sistema.	Configuración_ nueva	/media/JD1/conf/prueba.xml	localhost	BD_sgu_nucleo	sgu_nucleo_user	*****777777 777	botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".	la vista principal de la aplicación de selección Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración, escoge la acción de Modificar e introduce datos de una configuración
		V	V	V	V	V	V		
		Configuracióntt	/media/DG1/config/c.xml	localhost	BD_sgu_residencia	sgu_residencia.user	WWwfg666		



									que ya existe en el sistema y presiona el botón Aceptar.
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar una configuración.	I	I	V	V	V	I	El sistema muestra encima del nombre del componente que debe ser llenado obligatorio el mensaje de color rojo: "Campo requerido".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar.
		Vacío	Vacío	localhost	BD_sgu_residencia	sgu_residencia_user	Vacío		
		V	V	I	I	I	V		
		Config	/media/JD1/conf/prueba.xml	Vacío	Vacío	Vacío	amed...ccc		



									-El usuario selecciona una configuración, escoge la acción de Modificar, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
EC 1.4	Mediante este escenario se introducen datos incorrectos para modificar una configuración.	V	I	V	V	V	V	El sistema muestra encima del nombre del componente el mensaje de color rojo: "El directorio no existe".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú.
Insertar datos incorrectos .		config_de_prueba	desktop	localhost	BD_sgu_pregado	sgu_pregrado_user	Aaa!!!!		-El sistema
		I	V	V	V	V	V	El sistema no	



		asdfghjklpoiuytr ewq_dfghhjjkll ñ	/home/Daniel/cro n/e.xml	localhost	BD_sgu_nucl eo	sgu_nucleo_ user	password	permite la entrada de más de 30 caracteres.	muestra las acciones de contexto que se pueden realizar.
		I	V	V	V	V	V	El sistema muestra encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	-El usuario selecciona una configuración, escoge la acción de Modificar, introduce datos incorrectos y presiona el botón Aceptar.
		ad35 "*****"	/home/Jeanie/cro n/e.xml	localhost	BD_sgu_resid encia	sgu_resideci a_user	contraseña	El sistema muestra el mensaje de error en forma de ventana: "Los parámetros de conexión no son válidos".	
		V	V	I	I	I	V		
		conf_nueva66	/media/JD1/conf/ prueba.xml	dghjku""2	BD_mía@@	dgvelazquez bg	***** *		



<p>EC 1.5 Cancelar operación.</p>	<p>Mediante este escenario se cancela la modificación de la configuración.</p>	<p>NA</p>	<p>NA</p>	<p>NA</p>	<p>NA</p>	<p>NA</p>	<p>NA</p>	<p>El sistema muestra el mensaje de confirmación: "¿Está seguro que desea salir?".</p>	<p>-El usuario una vez en la vista principal de la aplicación selecciona Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración, escoge la acción de Modificar, llena o no los datos y presiona el</p>
---------------------------------------	--	-----------	-----------	-----------	-----------	-----------	-----------	--	---



									botón Cancelar.
--	--	--	--	--	--	--	--	--	-----------------

Tabla 42. Diseño de casos de prueba. Eliminar configuración.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar configuración.	Mediante este escenario se elimina una configuración.	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido eliminado satisfactoriamente".	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Configuración en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y el listado de configuraciones existentes.</p> <p>El usuario selecciona una configuración, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?.</p> <p>-Al dar clic en el botón Aceptar elimina la configuración y muestra el listado de configuraciones actualizado.</p>
EC 1.2 Cancelar	Mediante este escenario el usuario	El sistema cancela la operación y	-El usuario una vez dentro de la vista principal de la



<p>eliminar configuración.</p>	<p>opreme el botón Cancelar que se encuentra en la ventana de confirmación de eliminar configuración.</p>	<p>se muestra el listado de configuraciones existentes.</p>	<p>aplicación selecciona Configuración en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de configuraciones existentes. -El usuario selecciona una configuración, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -El usuario da clic en el botón Cancelar y el sistema muestra nuevamente la ventana con el listado de configuraciones existentes.</p>
--------------------------------	---	---	---

Tabla 43. Diseño de casos de prueba. Mostrar objetos del negocio.

Escenario	Descripción	Respuesta del sistema	Flujo central
<p>EC 1.1 Mostrar datos de forma correcta.</p>	<p>Mediante este escenario se mostrará al usuario el listado de los objetos del negocio existentes en el sistema que pertenecen a una configuración seleccionada. Además se muestra un conjunto de</p>	<p>El sistema muestra el listado de los elementos actualizados hasta la fecha.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El usuario selecciona una configuración y se muestra el listado de los objetos del negocio pertenecientes a esa configuración.</p>



	acciones externas a desarrollar sobre cada elemento: Crear, Modificar, Ver detalles y Eliminar.		
EC 1.2 No existen elementos creados.	Mediante este escenario en caso de que no exista creado ningún objeto del negocio para la configuración seleccionada se muestra un listado vacío.	El sistema muestra el listado vacío.	-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El usuario selecciona una configuración y se muestra el listado de los Objetos del negocio pertenecientes a esa configuración.

Tabla 44. Diseño de casos de prueba. Crear objeto del negocio. Parte uno.

Escenario	Descripción	Nombre del objeto	Nombre visual	Campos	Nomenclador	Origen de datos	Campo visual	Campo principal
EC 1.1 Crear datos correctamente.	Mediante este escenario se introducen datos correctos en el sistema para crear	V	V	V	V	V	V	V
		Objeto	nombre_visual1	nombre_esta do_civil	Habilitado	sq_disenno_carr era.tbn_anno_ac ademico	nombre_anno_ academico	id_anno_acade mico



un objeto del negocio.	V	V	V	V	V	V	V
	Objeto_bn	nombre_visual2	primer_nombre	Habilitado	sq_master_integracion_fn_obtener_color_piel ("")	nombre_color_piel	id_color_piel
	V	V	V	V	V	V	V
	Se introduce un texto con solo letras, que tenga como mínimo 1 caracter y como máximo 30, donde las palabras están separadas por guión bajo.	nombre_visual3	carnet_identidad	Habilitado	No aplica	nombresexo	idsexo
V	V	V	V	V	V	V	



		Objeto_estudiante	otro_nombre	folio	Deshabilitado	No aplica	No aplica	No aplica
EC 1.2 Insertar elementos repetidos.	Mediante este escenario se introducen datos para crear un objeto del negocio que ya existe en el sistema.	I	V	V	V	V	V	V
		Objeto_bn	nombre_visual9	primer_nombre	Deshabilitado	No aplica	No aplica	No aplica
		V	V	V	V			V
		Objeto_nuevo	nombre_visual3	carnet_identidad	Deshabilitado	No aplica	No aplica	No aplica
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para crear un objeto del negocio.	I	I	I	V	V	V	V
		Vacio	Vacio	nombre_estado_civil	Deshabilitado	No aplica	No aplica	No aplica
		V	V	V	V	V	I	I
		Mi_objeto	nombre_color_piel	primer_nombre	Habilitado	No aplica	Vacio	Vacio



		V	V	V	V	I	I	I
		Otro_objeto	nombre_visual BBB	carnet_identid ad	Habilitado	Vacio	Vacio	Vacio
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para crear un objeto del negocio.	I	V	V	V	V	I	I
		objeto1	nombre_visual	primer_nombre	Habilitado	sq_disenno_carrera.tbn_anno_academico	nombre_anno_academico25	id_anno_academico25
		V	I	V	V	V	V	V
		Otro_objeto	qwertyuiop;lkjh gfdsazxcvb098 76_fgtyuj	folio	Habilitado	No aplica	nombresexo	idsexo
		V	V	V	V	I	V	V
		Objeto66	otro_nombre_visual2	id_peloton	Habilitado	sq_master_integracion_fn_obtener_color_piel ("""").	nombre_color_piel	id_color_piel
		V	V	V	V	V	V	V
objeto_PP	Otro_nombreV 56777	id_peloton	Habilitado	sq_disenno_carrera.tbn_anno_academico56	nombre_anno_academico	id_anno_academico		



EC	1.5	Mediante este escenario se cancela la creación del objeto del negocio.	NA	NA	NA	NA	NA	NA	NA
----	-----	--	----	----	----	----	----	----	----

Tabla 45. Diseño de casos de prueba. Crear objeto del negocio. Parte dos.

Alias	Campo unión izquierdo	Campo unión derecho	Respuesta del sistema	Flujo central
V	V	V	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido creado satisfactoriamente".	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.</p> <p>-El usuario selecciona una configuración y se muestra un listado con los objetos que posee.</p> <p>-El usuario selecciona la acción de Crear, llena los datos correctamente y presiona el botón Aceptar.</p>
anno_academico	tb_destudiante.id_persona	tb_dpersona.id_persona		
V	V	V		
color_piel	Vacío	vacío		
V	V	V		
No aplica	id_provincia	id_provincia		
V	V	V		
No aplica	tb_dpersona.id_persona	tb_dpersona_data_plus.id_persona		



V	V	V	El usuario presiona el botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.</p> <p>-El usuario selecciona una configuración y se muestra un listado con los objetos que posee.</p> <p>-El usuario selecciona la acción de Crear, introduce datos de un objeto del negocio que ya está en el sistema y presiona el botón Aceptar.</p>
No aplica	estado_civil	estado_civil		
V	V	V		
No aplica	Vacío	Vacío		
V	V	V	El sistema muestra encima del nombre del componente que debe ser llenado obligatorio el mensaje de color rojo: "Campo requerido".	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una</p>
No aplica	Vacío	Vacío		
V	V	V		
No aplica	tb_destudiante.id_persona	tb_ddireccion.id_persona		



I	V	V		configuración.
Vacío	Vacío	vacío		-El usuario selecciona una configuración y se muestra un listado con los objetos que posee. -El usuario selecciona la acción de Crear, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
I	V	V	El sistema muestra encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.
25	No aplica	No aplica		-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.
V	V	V	El sistema no permite la entrada de más de 30 caracteres.	
No aplica	tb_destudiante.id_persona	tb_dpersona.id_persona		
V	V	V	El sistema muestra un mensaje en rojo: "Origen de datos no válido".	-El usuario selecciona una configuración y se muestra un listado con los objetos que posee. -El usuario selecciona la acción de Crear, introduce datos incorrectos y presiona el
color_piel	No aplica	No aplica		
V	V	V	El sistema muestra en rojo	



anno_academico	No aplica	No aplica	encima del componente: "Entre letras, números y guión bajo".	botón Aceptar.
NA	NA	NA	El sistema muestra el mensaje de confirmación ¿Está seguro que desea salir?	-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración. -El usuario selecciona una configuración y se muestra un listado con los objetos que posee. -El usuario selecciona la acción de Crear, llena o no los datos y presiona el botón Cancelar.

Tabla 46. Diseño de casos de prueba. Modificar objeto del negocio. Parte uno.

Escenario	Descripción	Nombre del objeto	Nombre visual	Campos	Nomenclador	Origen de datos	Campo visual	Campo principal
EC 1.1	Mediante este	V	V	V	V	V	V	V



Modificar datos correctamente.	escenario se introducen datos correctos en el sistema para modificar un objeto del negocio.	Objeto	nombre_visual1	nombre_estado_civil	Habilitado	sq_disenno_carrera.tbn_anno_academico	nombre_anno_academico	id_anno_academico
		V	V	V	V	V	V	V
		Objeto_bn	nombre_visual2	primer_nombre	Habilitado	sq_master_integracion_fn_obtener_color_piel ("")	nombre_color_piel	id_color_piel
		V	V	V	V	V	V	V
		Se introduce un texto con solo letras, que tenga como mínimo 1 caracter y como máximo 30, donde las palabras están separadas por guión bajo.	nombre_visual3	carnet_identidad	Habilitado	No aplica	nombresexo	idsexo
		V	V	V	V	V	V	V
		Objeto_estudiante	otro_nombre	folio	Deshabilitado	No aplica	No aplica	No aplica
EC 1.2 Insertar	Mediante este escenario se	I	V	V	V	V	V	



elementos repetidos.	introducen datos para modificar un objeto del negocio que ya existe en el sistema.	Objeto_bn	nombre_visual9	primer_nombre	Deshabilitado	No aplica	No aplica	No aplica
		V	V	V	V			V
		Objeto_nuevo	nombre_visual3	carnet_identidad	Deshabilitado	No aplica	No aplica	No aplica
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar un objeto del negocio.	I	I	I	V	V	V	V
		Vacío	Vacío	nombre_estado_civil	Deshabilitado	No aplica	No aplica	No aplica
		V	V	V	V	V	I	I
		Mi_objeto	nombre_color_piel	primer_nombre	Habilitado	No aplica	Vacío	Vacío
		V	V	V	V	I	I	I
		Otro_objeto	nombre_visual BBB	carnet_identidad	Habilitado	Vacío	Vacío	Vacío
EC 1.4 Insertar datos incorrectos	Mediante este escenario se introducen datos incorrectos para	I	V	V	V	V	I	I
		objeto1	nombre_visual	primer_nombre	Habilitado	sq_disenno_carrera.t bn_anno_academico	nombre_anno_academico25	id_anno_academico25



.	modificar un objeto del negocio.	V	I	V	V	V	V	V
		Otro_objeto	qwertyuiop;lkjh gfdsazxcvb098 76_fgtyuj	folio	Habilitado	No aplica	nombre_sexo	id_sexo
		V	V	V	V	I	V	V
		Objeto66	otro_nombre_v sual2	id_peloton	Habilitado	sq_master_integracio n_fn_obtener_color_p iel ("")""".	nombre_color_ piel	id_color_piel
		V	V	V	V	V	V	V
		objeto_PP	Otro_nombreV 56777	id_peloton	Habilitado	sq_disenno_carrera.t bn_anno_academico5 6	nombre_anno_ academico	id_anno_academ ico
EC 1.5	Mediante este escenario se cancela la modificación del objeto del negocio.	NA	NA	NA	NA	NA	NA	NA

Tabla 47. Diseño de casos de prueba. Modificar objeto del negocio. Parte dos.

Alias	Campo unión izquierdo	Campo unión derecho	Respuesta del sistema	Flujo central
V	V	V	El sistema actualiza el listado y	-El usuario una vez dentro de la vista



anno_academico	tb_destudiante.id_persona	tb_dpersona.id_persona	muestra el mensaje de información "El elemento ha sido modificado satisfactoriamente".	principal de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración. -El usuario selecciona una configuración y se muestra un listado con los objetos que posee. -El usuario selecciona un objeto del negocio y escoge la acción de Modificar, llena los datos correctamente y presiona el botón Aceptar.
V	V	V		
color_piel	Vacío	vacío		
V	V	V		
No aplica	id_provincia	id_provincia		
V	V	V		
No aplica	tb_dpersona.id_persona	tb_dpersona_data_plus.id_persona	El usuario presiona el botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.
V	V	V		
No aplica	estado_civil	estado_civil		
V	V	V		



No aplica	Vacío	Vacío		<p>-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.</p> <p>-El usuario selecciona una configuración y se muestra un listado con los objetos que posee.</p> <p>-El usuario selecciona un objeto del negocio y escoge la opción de Modificar, introduce datos de un objeto del negocio que ya está en el sistema y presiona el botón Aceptar.</p>
V	V	V	<p>El sistema muestra encima del nombre del componente que debe ser llenado obligatorio el mensaje de color rojo: "Campo requerido".</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.</p>
No aplica	Vacío	Vacío		
V	V	V		
No aplica	tb_destudiante.id_persona	tb_ddireccion.id_persona		
I	V	V		



Vacío	Vacío	vacío		-El usuario selecciona una configuración y se muestra un listado con los objetos que posee. -El usuario selecciona un objeto del negocio y escoge la opción de Crear, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
I	V	V	El sistema muestra encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.
25	No aplica	No aplica		
V	V	V	El sistema no permite la entrada de más de 30 caracteres.	-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.
No aplica	tb_destudiante.id_persona	tb_dpersona.id_persona		
V	V	V	El sistema muestra un mensaje en rojo: "Origen de datos no válido".	-El usuario selecciona una configuración y se muestra un listado con los objetos que posee.
color_piel	No aplica	No aplica		
V	V	V	El sistema muestra en rojo encima del componente: "Entre letras, números y guión bajo".	-El usuario selecciona un objeto del negocio y escoge la opción de Modificar, introduce datos incorrectos y presiona el botón Aceptar.
anno_academico	No aplica	No aplica		



NA	NA	NA	El sistema muestra el mensaje de confirmación ¿Está seguro que desea salir?	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y brinda la posibilidad de seleccionar una configuración.</p> <p>-El usuario selecciona una configuración y se muestra un listado con los objetos que posee.</p> <p>-El usuario selecciona un objeto del negocio y escoge la opción de Modificar, llena o no los datos y presiona el botón Cancelar.</p>
----	----	----	---	---

Tabla 48. Diseño de casos de prueba. Ver detalles de objeto del negocio.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Ver detalles.	Mediante este escenario se muestra toda la información del objeto del negocio seleccionado.	El sistema muestra en una ventana el listado de propiedades de un objeto del negocio con los siguientes datos: Propiedad, Nombre visual, Nomenclador,	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.</p> <p>-El sistema muestra las acciones de</p>



		Tipo de dato y Referencia.	contexto que se pueden realizar. -El usuario selecciona un objeto del negocio perteneciente a una configuración seleccionada y escoge la acción de Ver detalles.
EC 1.2 Cerrar la ventana de Ver detalles por el botón Cerrar.	Mediante este escenario se puede cerrar la ventana de Ver detalles.	El sistema cierra la ventana y muestra el listado con los objetos del negocio pertenecientes a una configuración seleccionada.	-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona un objeto del negocio perteneciente a una configuración seleccionada, escoge la acción de Ver detalles y presiona el botón Cerrar.

Tabla 49. Diseño de casos de prueba. Eliminar objeto del negocio.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar objeto	Mediante este escenario se	El sistema actualiza el listado	-El usuario una vez dentro de la vista principal



del negocio.	elimina un objeto del negocio.	y muestra el mensaje de información "El elemento ha sido eliminado satisfactoriamente".	de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración y de ella se muestra un listado de objetos del negocio. -El usuario selecciona un objeto del negocio, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -Al dar clic en Aceptar elimina el objeto del negocio y muestra el listado de objetos del negocio actualizado.
EC 1.2 Cancelar eliminar	Mediante este escenario el	El sistema cancela la	-El usuario una vez dentro de la vista principal



<p>objeto del negocio.</p>	<p>usuario oprime el botón Cancelar que se encuentra en la ventana de confirmación de eliminar objeto del negocio.</p>	<p>operación y muestra el listado de objetos del negocio existentes.</p>	<p>de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración y de ella se muestra un listado de objetos del negocio. -El usuario selecciona un objeto del negocio, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -El usuario da clic en el botón Cancelar y el sistema muestra nuevamente la ventana con el listado de objetos del negocio pertenecientes a la configuración seleccionada anteriormente.</p>
----------------------------	--	--	--

Tabla 50. Diseño de casos de prueba. Eliminar propiedad.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar	Mediante este escenario se	El sistema actualiza el listado de	-El usuario una vez dentro de la vista principal



propiedad.	elimina una propiedad de un objeto del negocio.	propiedades asociadas al objeto del negocio que se esté creando o modificado.	de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración y de ella se muestra un listado de objetos del negocio. -El usuario selecciona un objeto del negocio y presiona el ícono de Crear o Modificar. -El usuario asocia Campos al objeto del negocio y se muestra un listado de propiedades. -El usuario selecciona una propiedad y selecciona el ícono de Eliminar propiedad. -El sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -El usuario da clic en Aceptar, se elimina la propiedad y muestra el listado de propiedades actualizado.
EC 1.2 Cancelar	Mediante este escenario el	El sistema cancela la operación y	-El usuario una vez dentro de la vista principal



eliminar propiedad.	usuario oprime el botón Cancelar que se encuentra en la ventana de confirmación de eliminar propiedad.	muestra el listado de propiedades del objeto del negocio.	de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración y de ella se muestra un listado de objetos del negocio. -El usuario selecciona un objeto del negocio y presiona el ícono de Crear o Modificar. -El usuario asocia Campos al objeto del negocio y se muestra un listado de propiedades. -El usuario selecciona una propiedad y presiona el ícono de Eliminar propiedad. -El sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -El usuario da clic en el botón Cancelar y el sistema muestra nuevamente el listado de propiedades asociadas.
---------------------	--	---	--

Tabla 51. Diseño de casos de prueba. Eliminar unión de tablas.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------



EC 1.1 Eliminar unión de tablas.	Mediante este escenario se elimina una unión entre dos tablas de la base de datos.	El sistema actualiza el listado de uniones de tablas de las cuales se quieren adicionar campos al objeto del negocio que se esté creando o modificado.	<ul style="list-style-type: none">-El usuario una vez dentro de la vista principal de la aplicación selecciona Objetos del Negocio en las opciones del menú.-El sistema muestra las acciones de contexto que se pueden realizar.-El usuario selecciona una configuración y de ella se muestra un listado de objetos del negocio.-El usuario selecciona un objeto del negocio y presiona el ícono de Crear o Modificar.-El usuario asocia Campos de diferentes tablas al objeto del negocio y se muestra un listado con las uniones entre tablas realizadas.-El usuario selecciona una condición y selecciona el ícono de Eliminar.-El sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?.-El usuario da clic en Aceptar, se elimina la unión y muestra el listado actualizado.
EC 1.2 Cancelar	Mediante este escenario el	El sistema cancela la operación y	<ul style="list-style-type: none">-El usuario una vez dentro de la vista principal



<p>eliminar unión entre tablas.</p>	<p>usuario oprime el botón Cancelar que se encuentra en la ventana de confirmación de eliminar unión.</p>	<p>muestra el listado de uniones existentes.</p>	<p>de la aplicación selecciona Objetos del Negocio en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar. -El usuario selecciona una configuración y de ella se muestra un listado de objetos del negocio. -El usuario selecciona un objeto del negocio y presiona el ícono de Crear o Modificar. -El usuario asocia Campos de diferentes tablas al objeto del negocio y se muestra un listado con las uniones entre tablas realizadas. -El usuario selecciona una condición y selecciona el ícono de Eliminar. -El sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -El usuario da clic en el botón Cancelar y el sistema muestra nuevamente el listado de uniones entre tablas.</p>
-------------------------------------	---	--	--

Tabla 52. Diseño de casos de prueba. Mostrar reglas.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------



<p>EC 1.1 Mostrar datos de forma correcta</p>	<p>Mediante este escenario se mostrará al usuario el listado de reglas existentes. Además se muestra un conjunto de acciones a desarrollar sobre cada elemento: Crear, Modificar, Eliminar y Ejecutar.</p>	<p>El sistema muestra el listado de los elementos actualizados hasta la fecha.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra el listado de las reglas existentes y las acciones de contexto que se pueden realizar.</p>
<p>EC 1.2 No existen elementos creados.</p>	<p>Mediante este escenario en caso de que no exista creada ninguna regla se muestra un listado vacío.</p>	<p>El sistema muestra el listado vacío.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra el listado de las reglas existentes y las acciones de contexto que se pueden realizar.</p>

Tabla 53. Diseño de casos de prueba. Crear regla. Parte uno.

Escenario	Descripción	Nombre de regla	Tipo de regla	Acción	Configuración	Objeto	Función agregada	Propiedad	Distinct	Todo	Entrada manual	Objeto (condición)	Propiedad (condición)



EC 1.1	Mediante este	V	V	V	V	V	V	V	V	V	V		V
Insertar datos correctamente. el sistema para crear una regla.	escenario se introducen datos correctos en el sistema	Regla	Negocio	Seleccionar	Configuración_BB	estudiante	COUNT	Vacío	Deshabilitado	Habilitado	No aplica	estudiante	anno_academico
	correctos en el sistema para crear una regla.	V	V	V	V	V	V	V	V	V	V	V	V
	correctos en el sistema para crear una regla.	Regla_nueva	Validación	Actualizar	otra_configuración	persona	No aplica	primer_nombre	No aplica	No aplica	No aplica	persona	primer_nombre
	correctos en el sistema para crear una regla.	V	V	V	V	V	V	V	V	V	V	V	V
	correctos en el sistema para crear una regla.	Se introduce un texto que tenga como mínimo 1 caracter y	Rectificación	Transformar	Config	otro_objeto	No aplica	id_primer_nombre	No aplica	No aplica	No aplica	Vacío	Vacío



		como máximo 30, donde las palabras están separadas por guión bajo.											
EC 1.2	Mediante este escenario se introducen datos para crear una regla que ya existe en el sistema.	I	V	V	V	V	V	V	V	V	V	V	V
Insertar elementos repetidos.		Regla	Negocio	Eliminar	aaaaa	objeto_MMMM	No aplica	Id_periodo_lectivo	No aplica	No aplica	No aplica	objeto_MM MM	id_periodo_lectivo
		I	V	V	V	V	V	V	V	V	V	V	V
		Regla	Negocio	Insertar	confin_aaa	alumno_ayuda	No aplica	primer_nombre	No aplica	No aplica	Vacío	estudiante	fecha_registro



		a				nteBB							
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para crear una regla.	I	I	I	I	I	V	I	V	V	V	V	V
		Vacío	Vacío	Vacío	Vacío	Vacío	No aplica	Vacío	Deshabilitado	Habilitado	Deshabilitado	Vacío	Vacío
		V	V	V	V	V	V	V	V	V	V	V	V
		mi_re gla	Validación	Insertar	mi_conf g	Objeto_ bpm	No aplica	primer_n bre	Deshabilitado	Deshab ilitado	Arianna	No aplica	No aplica
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para crear una regla.	I	V	V	V	V	V	V	V	V	V		V
		asd 45567 788	Negoci o	Selecci onar	Configur ación_B B	estudia nte	COUNT	Vacío	Deshabilitado	Habilitado	No aplica	estudiante	anno_ac ademico
		I	V	V	V	V	V	V	V	V	V	V	V
		qqqq qqqq_ ttggg bb_45 ty7ui 98	Negoci o	Insertar	confin_a aa	alumno _ayuda nteBB	No aplica	primer_n bre	No aplica	No aplica	Vacío	estudiante	fecha_re gistro



EC 1.5	Mediante este escenario se cancela la creación de una regla.	NA	NA	NA	NA	NA								
Cancelar operación.														

Tabla 54. Diseño de casos de prueba. Crear regla. Parte dos.

Signo	Valor (condición)	Función agregada (Having)	Objeto/Propiedad	Signo (Having)	Objeto (Agrupación)	Objeto (Ordenar)	Ascendente	Descendente	Respuesta del sistema	Flujo central
V	V	V	V	V	V	V	V	V	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido creado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona
distinto de	primer_anno	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío		
V	V	V	V	V	V	V	V	V		
igual a	("")*	No aplica	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío		
V	V	V	V	V	V	V	V	V		
Vacío	Vacío	Vacío	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		



										el ícono de Crear, llena los datos correctamente y presiona el botón Aceptar.	
V igual a	V primer_anno	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	El usuario presiona el botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona la acción de Crear e introduce datos de una regla que ya existe en el sistema y presiona el botón Aceptar.
V igual a	V 06/04/2013	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica	V No aplica		



V	V	V	V	V	V	V	V	V	El sistema muestra encima del nombre del componente que debe ser llenado obligatorio el mensaje de color rojo: "Campo requerido".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona la acción de Crear, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
Vacío	Vacío	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		
V	V	V	V	V	V	V	V	V		
No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		
V	V	V	V	V	V	V	V	V	El sistema muestra encima del nombre del componente el mensaje de color	-El usuario una vez dentro de la vista principal de la aplicación selecciona
distinto de	primer_anno	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío		



									rojo: "Entre letras y guión bajo".	Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente.
V	V	V	V	V	V	V	V	V	El sistema no permite la entrada de más de 30 caracteres.	-El usuario selecciona la acción de Crear, introduce datos incorrectos y presiona el botón Aceptar.
igual a	06/04/2013	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		
								NA	El sistema muestra el mensaje de confirmación: "¿Está seguro que desea salir?".	-El usuario una vez en la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas



													existente. -El usuario selecciona el ícono de Crear, llena o no los datos y presiona el botón Cancelar.
--	--	--	--	--	--	--	--	--	--	--	--	--	--

Tabla 55. Diseño de casos de prueba. Modificar regla. Parte uno.

Escenario	Descripción	Nombre de regla	Tipo de regla	Acción	Configuración	Objeto	Función agregada	Propiedad	Distint	Todo	Entrada manual	Objeto (condición)	Propiedad (condición)	
EC 1.1 Insertar datos correctamente. correctamente. para modificar una regla.	Mediante este escenario se introducen datos correctos en el sistema para modificar una regla.	V	V	V	V	V	V	V	V	V	V		V	
		Regla	Negocio	Seleccionar	Configuración_BB	estudiante	COUNT	Vacío	Deshabilitado	Habilitado	No aplica	estudiante	anno_academico	
		V	V	V	V	V	V	V	V	V	V	V	V	
		Regla_nueva	Validación	Actualizar	otra_configuración	persona	No aplica	primer_nombre	No aplica	No aplica	No aplica	No aplica	persona	primer_nombre
		V	V	V	V	V	V	V	V	V	V	V	V	V
		Se introduce un texto que tenga	Rectificación	Transformar	Config	otro_objeto	No aplica	id_primer_nombre	No aplica	No aplica	No aplica	Vacío	Vacío	



		como mínimo 1 caracter y como máximo 30, donde las palabras están separadas por guión bajo.											
EC 1.2	Mediante este escenario se introducen datos para modificar una regla que ya existe en el sistema.	I	V	V	V	V	V	V	V	V	V	V	V
Insertar elementos repetidos.		Regla	Negocio	Eliminar	aaaaa	objeto_MMMM	No aplica	Id_peri odo_lectivo	No aplica	No aplica	No aplica	objeto_MM MM	id_peri odo_lectivo
		I	V	V	V	V	V	V	V	V	V	V	V
		Regla_nueva	Negocio	Insertar	confin_aaa	alumno_ayuda nteBB	No aplica	primer_nombre	No aplica	No aplica	Vacío	estudiante	fecha_registro
EC 1.3	Mediante este	I	I	I	I	I	V	I	V	V	V	V	V



Insertar datos incompletos.	escenario no se introducen todos los datos para modificar una regla.	Vacío	Vacío	Vacío	Vacío	Vacío	No aplica	Vacío	Deshabilitado	Habilitado	Deshabilitado	Vacío	Vacío
		V	V	V	V	V	V	V	V	V	V	V	V
		mi_regla	Validación	Insertar	mi_config	Objeto_bpm	No aplica	primer_nombre	Deshabilitado	Deshabilitado	Arianna	No aplica	No aplica
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para modificar una regla.	I	V	V	V	V	V	V	V	V	V		V
		asdg 45567788	Negocio	Seleccionar	Configuración_BB	estudiante	COUNT	Vacío	Deshabilitado	Habilitado	No aplica	estudiante	anno_academico
		I	V	V	V	V	V	V	V	V	V	V	V
		qqqqqqq q_ttggggb b_45ty7ui9 8	Negocio	Insertar	confin_aaa	alumno_ayudanteBB	No aplica	primer_nombre	No aplica	No aplica	Vacío	estudiante	fecha_registro
EC 1.5 Cancelar operación.	Mediante este escenario se cancela la modificación de una regla.	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA



Tabla 56. Diseño de casos de prueba. Modificar regla. Parte dos.

Signo	Valor (condición)	Función agregada (Having)	Objeto/ Propiedad	Signo (Having)	Objeto (Agrupación)	Objeto (Ordenar)	Ascendente	Descendente	Respuesta del sistema	Flujo central
V	V	V	V	V	V	V	V	V	El sistema actualiza los datos y muestra el mensaje de información "El elemento ha sido modificado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona una regla presiona el ícono de Modificar, llena los datos correctamente y presiona el botón Aceptar.
distinto de	primer_anno	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío		
V	V	V	V	V	V	V	V	V		
igual a	("")*	No aplica	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío		
V	V	V	V	V	V	V	V	V		
Vacío	Vacío	Vacío	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		
V	V	V	V	V	V	V	V	V	El usuario presiona el botón Aceptar y el sistema muestra el	-El usuario una vez dentro de la vista principal de la aplicación selecciona
igual a	primer_anno	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		



V	V	V	V	V	V	V	V	V	mensaje de error: "El elemento ya existe".	Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona una regla y escoge la acción de Modificar. Introduce datos de una regla que ya existe en el sistema y presiona el botón Aceptar.
igual a	06/04/2013	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		
V	V	V	V	V	V	V	V	V	El sistema muestra	-El usuario una vez dentro de la vista principal de la aplicación selecciona
Vacío	Vacío	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	encima del nombre del componente que debe ser modificado	Reglas en las opciones del menú.
V	V	V	V	V	V	V	V	V	obligatorio el mensaje de color rojo: "Campo requerido".	-El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona una
No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		



										regla y escoge la acción de Modificar, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
V	V	V	V	V	V	V	V	V	El sistema muestra	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú.
distinto de	primer_anno	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	Vacío	encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	
V	V	V	V	V	V	V	V	V	El sistema no permite la entrada de más de 30 caracteres.	-El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente.
igual a	06/04/2013	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica	No aplica		-El usuario selecciona una regla y escoge la acción de Modificar, introduce datos incorrectos y presiona el botón Aceptar.
NA	NA	NA	NA	NA	NA	NA	NA	NA	El sistema muestra el mensaje de confirmación: "¿Está	-El usuario una vez en la vista principal de la aplicación selecciona



									seguro que desea salir?".	Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existente. -El usuario selecciona una regla y escoge el ícono de Modificar, llena o no los datos y presiona el botón Cancelar.
--	--	--	--	--	--	--	--	--	---------------------------	--

Tabla 57. Diseño de casos de prueba. Eliminar regla.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar regla.	Mediante este escenario se elimina una regla.	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido eliminado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existentes. El usuario selecciona una regla, presiona el ícono de Eliminar y el sistema muestra una



			<p>ventana con el mensaje: ¿"Está seguro de realizar la acción"?.</p> <p>-Al dar clic en el botón Aceptar elimina la regla y muestra el listado de reglas actualizado.</p>
EC 1.2 Cancelar eliminar regla.	Mediante este escenario el usuario oprime el botón Cancelar que se encuentra en la ventana de confirmación de eliminar regla.	El sistema cancela la operación y se muestra el listado de reglas existentes.	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú.</p> <p>-El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existentes.</p> <p>-El usuario selecciona una regla, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?.</p> <p>-El usuario da clic en el botón Cancelar y el sistema muestra nuevamente la ventana con el listado de reglas existentes.</p>

Tabla 58. Diseño de casos de prueba. Ejecutar regla.



Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Ejecutar regla correctamente.	Mediante este escenario se ejecuta una regla satisfactoriamente.	El sistema ejecuta la regla y muestra el mensaje de información "El elemento ha sido ejecutado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existentes. El usuario selecciona una regla y presiona el ícono de Ejecutar.
EC 1.2 Error en la ejecución de regla.	Mediante este escenario ocurre un error en la ejecución de una regla seleccionada.	El sistema muestra el mensaje de error: "No se pudo ejecutar la regla".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reglas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reglas existentes. -El usuario selecciona una regla, presiona el ícono de ejecutar. -Ocurre un error y se muestra un mensaje: "No se pudo ejecutar la regla".

Tabla 59. Diseño de casos de prueba. Mostrar tareas.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------



<p>EC 1.1 Mostrar datos de forma correcta</p>	<p>Mediante este escenario se mostrará al usuario el listado de tareas existentes en el sistema. Además se muestra un conjunto de acciones a desarrollar sobre cada elemento: Crear, Modificar, Ver detalles y Eliminar.</p>	<p>El sistema muestra el listado de los elementos actualizados hasta la fecha.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra el listado de tareas existentes y las acciones de contexto que se pueden realizar.</p>
<p>EC 1.2 No existen elementos creados.</p>	<p>Mediante este escenario en caso de que no exista creada ninguna tarea se muestra un listado vacío.</p>	<p>El sistema muestra el listado vacío.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra el listado de tareas existentes y las acciones de contexto que se pueden realizar.</p>

Tabla 60. Diseño de casos de prueba. Modificar tarea.

Escenario	Descripción	Nombre	Fecha de ejecución	Hor a	Minutos	AM	PM	Reglas	Respuesta del sistema	Flujo central
-----------	-------------	--------	--------------------	-------	---------	----	----	--------	-----------------------	---------------



EC 1.1 Modificar datos correctamente.	Mediante este escenario se introducen datos correctos en el sistema para modificar una tarea.	V	V	V	V	V	V	V	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido creado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea y la acción de Modificar, llena los datos correctamente y presiona el botón Aceptar.
		Tarea_nueva	lunes 8 de abril de 2012	1	1	AM	Deshabilitado	regla1		
		V	V	V	V	V	V	V		
		Tarea_de_chequeo	lunes 8 de abril de 2012	1	1	AM	Deshabilitado	regla1		
		V	V	V	V	V	V	V		
		Tarea_controladora	martes 9 de abril de 2012	5	10	Deshabilitado	PM	regla1 regla2		
		V	V	V	V	V	V	V		
		Se introduce un texto que tenga como mínimo 1 caracter y como máximo 30, donde las palabras están separadas	miércoles 10 de abril de 2012	10	45	Deshabilitado	PM	regla2		



		por guión bajo.								
EC 1.2	Mediante este escenario se introducen datos para modificar una tarea que ya existe en el sistema.	I	v	V	V	V	V	V		El usuario presiona el botón Aceptar y el sistema muestra el mensaje de error: "El elemento ya existe".
Modificar elementos repetidos.		Tarea_nueva	lunes 8 de abril de 2012	1	0	AM	Deshabilitado	regla1		
		I	V	V	V	V	V	V		
		Tarea_de_chequeo	viernes 9 de octubre de 2012	7	18	AM	Deshabilitado	regla1 regla2 regla5		-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea y la acción de Modificar.



											-El usuario introduce datos de una tarea que ya está en el sistema y presiona el botón Aceptar.
EC 1.3 Insertar datos incompletos para modificar una tarea.	Mediante este escenario no se introducen todos los datos para modificar una tarea.	I	I	V	V	V	V	V	El sistema muestra encima del nombre del componente que debe ser llenado obligatorio el mensaje de color rojo: "Campo requerido".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea y la acción de	
		Vacío	Vacío	4	30	AM	Deshabilitado	regla2 regla6			
		V	V	I	I	V	V	V			
		Tarea_bb	sábado 30 de marzo de 2013	2	Vacío	Deshabilitado	PM	regla3 regla5			
		V	V	V	V	V	V	I			
Se introduce un texto que tenga como mínimo 1 caracter y como máximo 30,	sábado 21 de abril de 2013	9	20	AM	Deshabilitado	Vacío					



		donde las palabras están separadas por guión bajo.								Modificar, deja de llenar campos que son obligatorios y presiona el botón Aceptar.
EC 1.4 Modificar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para modificar una tarea.	I	V	V	V	V	V	V	El sistema muestra	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea y la acción de Modificar, introduce
		Tarea 123 #\$%	martes 13 de noviembre de 2012	6	52	AM	Deshabilitado	regla7	encima del nombre del componente el mensaje de color rojo: "Entre letras y guión bajo".	
		I	V	V	V	V	V	V	El sistema no permite	
		qwertyuiop_ gggggg_sdd rfhijugjk	martes 13 de noviembre de 2012	10	10	AM	Vacío	regla1 regla2	la entrada de más de 30 caracteres.	



											datos incorrectos y presiona el botón Aceptar.
EC 1.5	Mediante este escenario se cancela la modificación de una tarea.	NA	NA	NA	NA	NA	NA	NA	NA	El sistema muestra el mensaje de confirmación: ¿Está seguro que desea salir?	-El usuario una vez en la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea y la acción de Modificar, llena o no los datos y presiona el botón Cancelar.

Tabla 61. Diseño de casos de prueba. Eliminar tarea.



Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Eliminar tarea.	Mediante este escenario se elimina una tarea.	El sistema actualiza el listado y muestra el mensaje de información "El elemento ha sido eliminado satisfactoriamente".	-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -Al dar clic en Aceptar elimina la tarea y muestra el listado de tareas actualizado.
EC 1.2 Cancelar	Mediante este escenario el usuario	El sistema cancela la	-El usuario una vez dentro de la vista principal



<p>eliminar tarea.</p>	<p>oprime el botón Cancelar que se encuentra en la ventana de confirmación de Eliminar tarea.</p>	<p>operación y muestra el listado de tareas existentes.</p>	<p>de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea, presiona el ícono de Eliminar y el sistema muestra una ventana con el mensaje: ¿"Está seguro de realizar la acción"?. -El usuario da clic en el botón Cancelar y el sistema muestra nuevamente la ventana con el listado de tareas existentes.</p>
------------------------	---	---	--

Tabla 62. Diseño de casos de prueba. Ver detalles de tarea.

Escenario	Descripción	Respuesta del sistema	Flujo central
<p>EC 1.1 Ver detalles.</p>	<p>Mediante este escenario se muestra toda la información de la tarea seleccionada.</p>	<p>El sistema muestra en una ventana el listado de reglas asociadas a una tarea seleccionada con los siguientes datos: Nombre de regla, Tipo de regla visual,</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de</p>



		Acción y Base de datos.	contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea y escoge la acción de Ver detalles.
EC 1.2 Cerrar la ventana de Ver detalles por el botón Salir.	Mediante este escenario se puede cerrar la ventana de Ver detalles.	El sistema cierra la ventana de Ver detalles y muestra el listado con las tareas existentes.	El usuario una vez dentro de la vista principal de la aplicación selecciona Tareas en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de tareas existentes. -El usuario selecciona una tarea, escoge la acción de Ver detalles y presiona el botón Cerrar.

Tabla 63. Diseño de casos de prueba. Mostrar reportes.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------



<p>EC 1.1 Mostrar datos de forma correcta</p>	<p>Mediante este escenario se mostrará al usuario el listado de reportes existentes en el sistema. Además se muestra un conjunto de acciones a desarrollar sobre cada elemento: Ver detalles de reporte inicial, Ver detalles de reporte final y Exportar.</p>	<p>El sistema muestra el listado de los elementos actualizados hasta la fecha.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Reportes en las opciones del menú. -El sistema muestra el listado de reportes existentes y las acciones de contexto que se pueden realizar.</p>
<p>EC 1.2 No existen elementos creados.</p>	<p>Mediante este escenario en caso de que no se haya generado ningún reporte se muestra un listado vacío.</p>	<p>El sistema muestra el listado vacío.</p>	<p>-El usuario una vez dentro de la vista principal de la aplicación selecciona Reportes en las opciones del menú. -El sistema muestra el listado de reportes existentes y las acciones de contexto que se pueden realizar.</p>

Tabla 64. Diseño de casos de prueba. Ver detalles de reporte inicial.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------



EC 1.1 Ver detalles.	Mediante este escenario se muestra toda la información de un reporte seleccionado.	El sistema muestra en una ventana un listado con un conjunto de datos que varían según el reporte seleccionado.	<ul style="list-style-type: none"> -El usuario una vez dentro de la vista principal de la aplicación selecciona Reportes en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reportes generados. -El usuario selecciona un reporte y escoge la acción de Ver detalles de reporte inicial.
EC 1.2 Cerrar la ventana de Ver detalles por el botón Salir.	Mediante este escenario se puede cerrar la ventana de Ver detalles.	El sistema cierra la ventana de Ver detalles y muestra el listado con los reportes existentes.	<ul style="list-style-type: none"> -El usuario una vez dentro de la vista principal de la aplicación selecciona Reportes en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reportes generados. -El usuario selecciona un reporte, escoge la acción de Ver detalles de reporte inicial y presiona el botón Cerrar.

Tabla 65. Diseño de casos de prueba. Ver detalles de reporte final.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------



EC 1.1 Ver detalles.	Mediante este escenario se muestra toda la información de un reporte seleccionado.	El sistema muestra en una ventana un listado con un conjunto de datos que varían según el reporte seleccionado. En caso que el reporte seleccionado haya sido generado tras una acción de selección la información mostrada será la misma que la información mostrada en el reporte inicial.	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reportes en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reportes generados. -El usuario selecciona un reporte y escoge la acción de Ver detalles de reporte final.
EC 1.2 Cerrar la ventana de Ver detalles por el botón Salir.	Mediante este escenario se puede cerrar la ventana de Ver detalles.	El sistema cierra la ventana de Ver detalles y muestra el listado con los reportes existentes.	-El usuario una vez dentro de la vista principal de la aplicación selecciona Reportes en las opciones del menú. -El sistema muestra las acciones de contexto que se pueden realizar y el listado de reportes generados. -El usuario selecciona un reporte, escoge la acción de Ver detalles de reporte final y presiona el botón Cerrar.