

Universidad de las Ciencias Informáticas

Facultad 4



Universidad de las Ciencias
Informáticas

***Sistema para la gestión de catálogos y lenguajes de
patrones de diseño de Recursos Educativos***

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

Autor (s): Elianys Ferriol Medero

Ernesto Siukiong Alay Navarrete

Tutor(s): Ing. Yasirys Terry González

Ing. Reiman Alfonso Azcuy

La Habana, junio del 2018.

Frase



Lo realmente importante es luchar para vivir la vida, para sufrirla y para gozarla, perder con dignidad y atreverse de nuevo.

Charles Chaplin.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis que tiene por título: Sistema para la gestión de catálogos y lenguajes de patrones de diseño de Recursos Educativos y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____

Elianys Ferriol Medero

Ernesto Siukiong Alay Navarrete

Tutor: Ing.Yasirys Terry González

Tutor: Ing.Reiman Alfonso Azcuy

Datos de Contacto

Yasirys Terry González: Ingeniera en Ciencias Informáticas (Universidad de las Ciencias Informáticas (UCI), República de Cuba, 2007). Matriculada en el Programa Doctoral de Informática. Profesora Asistente de la Facultad 4 en la UCI, 7 años de experiencia en el tema.

Reiman Alfonso Azcuy: Ingeniero en Ciencias Informáticas (Universidad de las Ciencias Informáticas (UCI), República de Cuba, 2017). Recién graduado en adiestramiento.

AGRADECIMIENTOS ELIANYS

A mis padres. A mi mamá, mi razón de ser, mi guía en todo momento, sin duda alguna la muestra perfecta de un ser luchador y tenaz a la vez, capaz de todo por lo que quiere. A mi papá Pedro y mi papá Ulises, por las muestras de apoyo todos estos años.

A mi compañero de tesis, mi amigo, mi confidente, mi pareja, el amor de mi vida sin duda alguna, por tantos momentos compartidos... y los que nos faltan, te amo popoto.

A mi abuela Yoya, te quiero abue.

A mi familia por tanto apoyo en todo momento (tíos, primos), por toda la confianza demostrada en estos años.

A mi querida Roxy, mi amiga, mi hermana, te extraño con el alma.

A mis suegros por siempre estar al tanto de todo, y por tanto apoyo siempre, por quererme como una más de la familia.

A mis amigos de siempre, de los años, los que se quedan en las buenas y en las malas.

A los nuevos amigos, siempre estarán conmigo como una de las cosas más grande que me dio la universidad, su amistad incondicional.

A todos los compañeros de aula, a los que no están y los que sí, que me llena de orgullo saber que muy pronto serán ingenieros.

A mis tutores Yasirys y Reiman, por tanto, apoyo y comprensión en todo momento.

A todos los profesores que, en estos cinco años, de una forma u otro aportaron tanto para que hoy estuviese aquí.

AGRADECIMIENTOS ERNESTO

A mis tutores Reiman y Yasiris por el apoyo brindado a lo largo del desarrollo de este trabajo de diploma.
A los viejos amigos y a todos los que he conocido en estos 5 intensos años de mi vida, especialmente aquellos con los que he convivido, con los que he compartido momentos llenos risas, tensión, euforia, lágrimas y hermandad.

A mis suegros, por recibirme como uno más de la familia y brindarme su apoyo en todo momento.
A mi abuelo Rolando de quien he aprendido ser paciente y entender que nuestras acciones del presente son la base necesaria para las generaciones futuras.

A mi abuela Panchita, mi segunda madre, por su tolerancia, su amor, su firmeza y apoyo incondicional de por vida, a quien espero enorgullecer siendo el hombre que soy gracias a ella.

A mi abuela Popó de quien aprendí a ser constante a poner la familia por encima de todos y luchar por su unidad y felicidad.

A mi hermano David, mi compañero, de nuestras grandes diferencias aprendí a hacer lo que me gusta, a sentirme bien conmigo mismo, a mirar la vida de distintos puntos de vista y lograr mis objetivos sin perjudicar a nadie teniendo siempre presente a la familia.

A mi hermanito Siuwei, mi pequeñuelo, de quien veo mucho de mí mismo, y quien me recuerda lo necesario que es volver a ese momento de inocencia, en que nos olvidamos de los problemas y volvemos a ser niños, de quien espero y mi esfuerzo por ser su modelo a seguir.

A mi novia Elianys, mi compañera, a quien le debo gran parte de los logros obtenidos en estos últimos 5 años, la cual me ha demostrado que, si existe la pareja ideal y a quien espero poder seguir haciendo feliz de ahora en adelante.

A mi papá, mi amigo y modelo a seguir, quien me enseñó todo lo necesario para enfrentar a la vida, quien ha hecho todo lo posible por complacerme y hacer más fácil mis días, mi confidente, de quien aprendí lo que es el respeto, la tenacidad, el perdón y el amor a la familia, a quien considero la persona más inteligente y noble de este mundo.

A mi mamá, mi amiga y mi primer amor, quien me enseñó a luchar por lo que quiero, quien nunca flagea en su devoción por sus hijos, en quien encuentro el refugio necesario para salir a delante, mi cielo eterno, de quien aprendí lo que es el sacrificio, el amor, la perseverancia y la compasión, a quien considero ejemplo de entereza y la persona más importante de este mundo.

Dedicatoria de Elianys

A mi familia, en especial a mi mamá por ser mi guía y mi fortaleza, sin ella este sueño no podría hacerse realidad.

A mis abuelos.

Dedicatoria de Ernesto

A mis padres, abuelos y hermanos, a quienes espero enorgullecer con la realización este trabajo

RESUMEN

Múltiples herramientas han surgido con el objetivo de integrar las Tecnologías de la Información y las Comunicaciones a los procesos de enseñanza - aprendizaje. Esto ha propiciado un incremento de los recursos educativos que no siempre cumplen con características didácticas - pedagógicas y la calidad requerida para su reutilización. Para prevenir este problema surgen los patrones de diseño de recursos educativos, que pretenden evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos, creando un estándar en el diseño de soluciones con la formalización de un lenguaje común a todos los diseñadores. El uso de patrones ayuda a solucionar problemas con los que frecuentemente se encuentran los diseñadores y permiten ahorrar tiempo y esfuerzo. Por otro lado, dichos patrones de diseño suelen agruparse en dos colecciones catálogos y lenguajes de patrones facilitando de esta manera la utilización de estos patrones, así como el modelado visual de las relaciones establecidas entre los mismos respectivamente. Como resultado final se desarrolló una aplicación con funcionalidades tales como: adicionar, eliminar, actualizar y mostrar un patrón de diseño y sus colecciones (catálogos y lenguajes), así como la de incluir la información de los metadatos asociados a estos, y enviar a edición y revisión. Para el desarrollo del sistema se utilizó la metodología AUP-UCI enmarcada en el escenario 4 de Historias de usuario, generando los artefactos que se definen en cada una de sus fases. La aplicación de la prueba de caja negra y la técnica de ladov confirman el cumplimiento de los objetivos propuestos.

PALABRAS CLAVE

Catálogo, lenguaje, metadatos, modelado visual, patrón de diseño, recursos educativos, relaciones de patrones.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS	6
Introducción	6
1.1 Recursos Educativos	6
1.2 Patrones de diseño.....	7
1.2.1 Catálogo y lenguaje de patrones	8
1.3 Herramientas que constituyen soluciones similares	9
1.4 Modelado visual de las relaciones entre patrones	14
1.5 Metodología de desarrollo de software	15
1.6 Lenguajes y tecnologías de programación	15
1.7 HTML	17
1.8 Framework de desarrollo	17
1.9 Gestores de base de datos PostgreSQL	18
1.10 Entorno de Desarrollo Integrado Netbeans	19
1.11 Lenguaje de modelado UML.....	19
1.13 Servidor web Xampp	20
Conclusiones parciales.....	20
CAPÍTULO 2: SOLUCIÓN AL PROBLEMA PLANTEADO	21
Introducción	21
2.1 Propuesta de solución.....	21
2.2 Modelado del negocio.....	22
2.2.1 Roles definidos para el negocio.....	22
2.2.2 Modelo de dominio.....	22
2.3 Roles definidos para el sistema	23
2.4 Requisitos funcionales y no funcionales	24
2.4.1 Descripción de los Requisitos funcionales	24
2.4.2 Especificación de los requisitos funcionales.....	33
2.4.3 Requisitos no funcionales	36
2.5 Estilo arquitectónico	37
2.5.1 Patrón arquitectónico Modelo Vista Controlador	37
2.5.2 Patrones de diseño	39
2.6 Diagrama Entidad-Relación.....	40
Conclusiones parciales.....	40
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....	42
Introducción	42
3.1 Modelo de implementación.....	42
3.1.1 Diagrama de Componentes	42
3.1.2 Estándares de codificación	43
3.2 Pruebas de Software.....	44
3.2.1 Métodos de pruebas de software	45
3.2.2 Resultados de las pruebas de softwares.....	45
3.3 Validación del sistema.....	52
3.3.1 Aplicación de la técnica de ladov	52
Conclusiones parciales.....	55
CONCLUSIONES GENERALES.....	56

RECOMENDACIONES.....	57
REFERENCIAS BIBLIOGRÁFICAS	58
ANEXOS	63

ÍNDICE DE ILUSTRACIONES Y TABLAS

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Lenguaje para la creación de cuestionarios on-line	15
Ilustración 2 Modelo de dominio	23
Ilustración 3 Diagrama Entidad-Relación	40
Ilustración 4 Diagrama de componente	43
Ilustración 5 Comportamiento de las no conformidades	51
Ilustración 6 Cuadro lógico de ladov	52
Ilustración 7 Ecuación para obtener el ISG	54

ÍNDICE DE TABLAS

Tabla 1 Comparación entre herramientas similares	11
Tabla 2 Estructura de patrones y sus colecciones	12
Tabla 3 Descripción de metadatos	13
Tabla 4 Roles del negocio	22
Tabla 5 Roles del sistema	23
Tabla 6 Descripción de los requisitos funcionales	25
Tabla 7 Requisito funcional 3 Generar patrón	33
Tabla 8 Resumen de las no conformidades	51
Tabla 9 Índice de satisfacción grupal ISG	54
Tabla 10 Nivel de satisfacción	55

Introducción

Las modalidades actuales del aprendizaje, tanto las totalmente virtuales como las mixtas, han traído consigo no solo una revolución en cuanto a los recursos materiales, sino que también han supuesto un cambio en la perspectiva sobre la educación [1]. Estas tendencias actuales de la sociedad están cada vez más influenciadas por el uso de las Tecnologías de la Información y las Comunicaciones (TIC), de este modo con la difusión del Internet se ha hecho posible que los usuarios de la gran red puedan compartir información y conocimientos sin importar su localización geográfica, idioma o cultura.

La enseñanza no ha quedado exenta de estas tendencias y en este sentido, ha evolucionado para potenciar el aprendizaje a distancia y la interacción más allá del hecho de compartir un mismo espacio físico, dando paso al surgimiento de los recursos educativos [1]. Los recursos educativos tienen como propósito contribuir a la formación del estudiante a través del cumplimiento de objetivos instructivos y educativos. Estos actualmente forman parte importante en el proceso de enseñanza-aprendizaje en la Educación Superior en Cuba.

Estos recursos son muy útiles desde el punto de vista de la reutilización de los conocimientos, pues permiten reflejar en ellos multitud de temas y, lo más importante, cualquier persona puede hacer un recurso educativo y brindarle a los demás el conocimiento del que se ha apropiado. Los recursos educativos surgen con el objetivo de mostrar públicamente, de manera absolutamente libre y gratuita, todos los materiales docentes que los profesores usan en sus aulas, así como la propia organización didáctica de las asignaturas y cursos impartidos en el centro, muchas son las instituciones que en todo el mundo se han unido, manteniendo como tema de intercambio los recursos educativos.

Mientras se navegue por internet o se utilicen medios digitales integrados en la modalidad de educación presencial, se observará que hay un auténtico estallido de información. Han proliferado los trabajos (artículos, investigaciones, etc.) con el consiguiente aumento en su variedad, pero también con la consiguiente repetición de muchos tanto en la temática del contenido como en la forma.

Tanto los diseñadores de recursos educativos como personas no expertas en estos temas se encuentran constantemente con problemas asociados no solo a la búsqueda de información sino también a la elaboración del propio contenido, teniendo la necesidad de

buscar soluciones construidas por otros autores o crear sus propias salidas. Ante estas circunstancias, con la demanda y al mismo tiempo necesidad de economizar esfuerzos, son precisas nuevas estrategias que faciliten la creación de soluciones aplicables en diversos contextos.

De modo que se puede entender que existen esfuerzos en una misma dirección, pero en paralelo, porque coexisten problemas comunes que se resuelven una y otra vez, y en muchos casos de manera muy semejante. Ante este hecho surge con más fuerza la idea de la creación de patrones de diseño, para evitar así la multiplicación de trabajos paralelos que resuelven de manera similar problemas que se repiten de forma reiterada a lo largo del tiempo en diversos escenarios [1].

Un patrón “es una forma literaria de documentar las mejores prácticas y lecciones aprendidas en la resolución de un problema complejo dentro de un dominio de diseño concreto” [2]. En sentido general son guías, directrices, plantillas que se generan cuando se resuelven muchas problemáticas que tienen grandes semejanzas y, por consiguiente, se crea un modelo de respuestas que se puede reutilizar para no realizar trabajos repetidos en el análisis de una posible solución del problema que ya ha sido resuelto [1] [3] [4] [5].

Los patrones de diseño de recursos educativos le dicen al usuario qué hacer, cómo, cuándo y por qué hacerlo [6], funcionando como un mapa que ayuda a la persona a crear su propio recurso educativo. Estos patrones se pueden agrupar, fundamentalmente, en dos tipos de colecciones, catálogos y lenguajes.

Los catálogos se utilizan para clasificar un conjunto de patrones utilizando uno o más criterios. En cambio, los lenguajes de patrones se utilizan para conformar un todo cohesionado que revela las estructuras y relaciones de sus componentes para cumplir con un objetivo compartido.

Motivado por las características de los ámbitos educativos actuales, un patrón de diseño educativo puede ser creado por un diseñador de aprendizaje experimentado, un docente, un estudiante, o incluso por una persona que no esté directamente relacionada con un proceso de enseñanza y aprendizaje formal. Básicamente, cualquier persona puede diseñar un recurso educativo y, con las bondades que ofrece la tecnología, compartirlo a una amplia comunidad.

En el estudio realizado a las herramientas informáticas para la gestión de los patrones de diseño acerca del proceso de gestión de patrones y sus colecciones se han identificado las siguientes limitaciones más comunes en las herramientas:

- No se ha definido una estructura común para la gestión de los patrones de diseño, ni de sus colecciones, y por ende no se permite la creación de nuevos elementos.
- Se limita la reutilización de dichos patrones, que conlleva a los diseñadores que buscan este tipo de información a crear sus propias soluciones, existiendo otras que le resuelven su problema de manera semejante.
- No son capaces de gestionar eficientemente las colecciones de patrones; este no contempla una estructura descriptiva de los campos para la creación de las mismas.
- Por otro lado, no permiten modelar visualmente las relaciones entre patrones de un lenguaje.
- No se toman en cuenta elementos complementarios o metadatos de apoyo a la búsqueda de información, el almacenamiento y la reutilización en sentido general de estos catálogos y lenguajes.

Los problemas mencionados inciden directamente en el proceso de gestión de los patrones de diseño de recursos educativos y permiten identificar el siguiente problema social:

- Falta de intercambio de conocimientos y experiencias entre diseñadores de recursos educativos, al no utilizarse con la frecuencia requerida los patrones de diseño y sus colecciones.
- Aumento indebido de trabajos en paralelos y como consecuencia el uso innecesario de tiempo y esfuerzos.

Problema a resolver: Insuficiente tratamiento a las colecciones de patrones de diseño de recursos educativos, para evitar la realización de trabajos repetidos y el uso innecesario de tiempo u otros recursos.

Se define como **objeto de estudio:** el proceso de gestión de patrones de diseño de recursos educativos, teniendo como **campo de acción:** la gestión de los catálogos y lenguajes de patrones de diseño de recursos educativos.

Objetivo general: Desarrollar un sistema para la gestión de catálogos y lenguajes de patrones de diseño que permita el modelado visual de las relaciones entre los patrones que los conforman.

Tareas a cumplir por estudiantes:

- Análisis de los conceptos y características de los patrones de diseño de recursos educativos y de las colecciones de patrones.
- Análisis comparativo de sistemas desarrollados para la gestión de patrones de diseño mediante un estudio bibliográfico.
- Fundamentación de las características del sistema a desarrollar.
- Selección de las herramientas y tecnologías a utilizar en el desarrollo del sistema para la gestión de catálogos y lenguajes de patrones de diseño de recursos educativos.
- Análisis y diseño del sistema para la gestión de catálogos y lenguajes de patrones de diseño de recursos educativos.
- Implementación de las funcionalidades del sistema para la gestión de catálogos y lenguajes de patrones de diseño de recursos educativos.
- Validación del sistema implementado mediante la aplicación de pruebas de software.

Para dar cumplimiento a los objetivos propuestos, se hizo necesario durante la confección del presente trabajo, el empleo de métodos teóricos, tales como:

- **Análisis-síntesis**, posibilitó el estudio de fuentes bibliográficas referentes al diseño y la gestión de recursos educativos y a las estrategias y tecnologías para la gestión de estos recursos, identificando los elementos necesarios para dar solución al problema planteado.
- **Histórico-lógico**, permitió el estudio relacionado con la evolución de las tecnologías asociadas al proceso de gestión de patrones en ambientes educativos.
- **Análisis documental**, potenció la extracción de información necesaria para definir los elementos a incluir en la propuesta de solución, al utilizarlo en la consulta de literatura especializada en temáticas relacionadas con el objeto de estudio.

El presente trabajo se estructura en introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas, glosario y anexos.

Estructura capitular:

Capítulo 1: Marco teórico referencial sobre los patrones de diseño de recursos educativos

Describe el marco teórico de la investigación, definiendo los principales conceptos y definiciones referentes a las colecciones de patrones de diseño de recursos educativos, y se seleccionan las herramientas necesarias para el desarrollo de un sistema que gestione los catálogos y lenguajes de patrones. También se fundamenta el uso de una metodología de desarrollo para la concepción de dicho sistema.

Capítulo 2: Solución al problema planteado

Describe la propuesta de solución, donde se definen y documentan los requisitos funcionales, no funcionales, especificaciones y se elabora el modelo del dominio donde quedan reflejados los principales conceptos que conforman el negocio. Además, se identifican los roles que interactuarán con el sistema, los patrones GRASP se emplean como buenas prácticas para lograr un producto de calidad y el patrón arquitectónico que caracterizará la aplicación.

Capítulo 3: Implementación y Prueba

Describe la implementación de los principales requisitos de la aplicación, las pruebas que se le hacen al software para ver si cumple con los requerimientos establecidos por el cliente, y la técnica de ladov para medir el nivel de aceptación por parte de los usuarios con respecto al sistema.

Capítulo 1: Marco teórico referencial sobre los patrones de diseño de Recursos educativos

Introducción

El auge que ha adquirido el desarrollo del aprendizaje por medio de los recursos educativos en los últimos años ha impulsado el desarrollo de numerosas herramientas que fortalecen los procesos de enseñanza-aprendizaje, propiciando que las instituciones educativas cuenten con recursos que permiten elevar los conocimientos, no solo por parte de los estudiantes sino también a favor de los propios educandos. Es por eso que el presente trabajo se describe el marco teórico de la investigación, definiendo los principales conceptos y definiciones referentes a las colecciones de patrones de diseño de recursos educativos, haciéndose necesario realizar un estudio de soluciones similares existentes y de las tecnologías idóneas para la implementación de la propuesta de solución, fundamentando de este modo las bases de la investigación.

1.1 Recursos Educativos

Un recurso educativo es cualquier tipo de recurso que se haya diseñado para su uso en los procesos de enseñanza-aprendizaje [6]. Estos recursos son muy útiles desde el punto de vista de reutilización de los conocimientos, pues permiten reflejar en ellos multitud de temas y, lo más importante, cualquier persona puede hacer un recurso educativo y brindar a los demás el conocimiento del que se ha apropiado.

El término REA (Recurso Educativo Abierto) se utilizó por primera vez en una conferencia organizada por la UNESCO en el 2002, donde fue definido como: “materiales digitalizados ofrecidos de forma libre y abierta a educadores, estudiantes y autodidactas para utilizar y reutilizar en la enseñanza, la educación o en la investigación [7]. Dicha definición ha sido utilizada en lo adelante en las características que se aplican a todo tipo de recursos educativos.

Los recursos educativos son medios a través de los cuales se transmite un conocimiento determinado, que puede ser consultado y compartido por cualquier persona [8], dicha definición de recursos educativos se asume en la presente investigación.

Los recursos educativos hacen referencia a los materiales educativos que son publicados gratuitamente para favorecer la educación de las personas. Contienen audios, multimedias, videos, animaciones, actividades interactivas para la evaluación del conocimiento plasmado en él, entre otros componentes. Pueden encontrarse en internet o

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

en repositorios. Su uso por múltiples personas alrededor del mundo los convierte en excelentes apoyos o complementos del proceso de enseñanza aprendizaje. En muchas ocasiones, se han desarrollado soluciones similares que resuelven el mismo problema en los diseños de recursos educativos, generando trabajos en paralelos y por otro lado empleo innecesario de tiempo u otros recursos. Para resolver esta situación se utilizan los patrones de diseño. Pretendiendo así evitar la multiplicación de trabajos que resuelven de manera semejante problemas similares [1].

1.2 Patrones de diseño

La definición inicial de este tema es la dada por Christopher Alexander en 1977, donde propuso la idea del patrón como aquel que: “describe un problema que ocurre una y otra vez y, a continuación, describe el núcleo de la solución de ese problema, de tal manera que el usuario puede utilizar esta solución un millón de veces más, sin tener que hacerlo de la misma manera dos veces” [9].

En un principio Alexander lo aplicó a su profesión de arquitecto, pero dejó la puerta abierta para trasladar su idea a otras disciplinas. Destacan dos características fundamentales en los patrones en aprendizaje: reusabilidad y flexibilidad [1], la primera se refiere a la cantidad de veces que puede ser usado un patrón para resolver un problema, y la segunda a que un patrón puede sufrir pequeñas adaptaciones para darle solución a un mismo problema.

Miguel Zapata aporta otro concepto más específico donde pone de manifiesto algunos contextos donde se puede utilizar un patrón. Él plantea que los patrones son “estructuras de información que permiten resumir y comunicar la experiencia acumulada y la resolución de problemas, tanto en la práctica como en el diseño, en programas de enseñanza y aprendizaje a través de redes” [9]. Los patrones en el ámbito del aprendizaje pretenden evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos, creando un estándar en el diseño de soluciones con la formalización de un lenguaje común a todos los diseñadores. Por ende, se facilitará el aprendizaje a las nuevas generaciones que ven condensado el conocimiento existente y disponen de un método ya conocido para la creación de nuevos patrones. No pretendiendo en ningún caso coartar la libertad creativa de cada persona que puede aplicar el patrón de la manera que más le convenga o guste, así como puede ofrecer nuevas alternativas [9].

Los patrones de diseño pueden agruparse mediante catálogos y lenguajes, el primero permite agruparlos respondiendo a un problema en específico, por otro lado, el lenguaje

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

permite mostrar las relaciones entre patrones teniendo en cuenta el modelado visual. El modelado visual de las relaciones entre patrones en un lenguaje facilita el entendimiento y el razonamiento dinámico de los usuarios, sin embargo, son escasas las herramientas que lo desarrollan, debido al alto nivel de complejidad que conlleva su realización. Actualmente, las relaciones establecidas entre patrones son insuficientes, pues la tendencia es a tratarlos como una entidad independiente y no como una colección de elementos interconectados (lenguaje de patrones). Aquellas soluciones que tienen en cuenta las relaciones entre patrones se limitan a la agrupación según una temática y no explican las relaciones entre ellos [2].

1.2.1 Catálogo y lenguaje de patrones

Un catálogo de patrones permite agrupar a los patrones teniendo en cuenta cualidades en común que tengan éstos. Suelen agruparse generalmente por el tipo de problema que resuelven, y pueden proporcionar, de conjunto, una solución.

Por lo tanto, se puede decir que un catálogo de patrones es: “un grupo de patrones clasificados por uno o más criterios y relacionados entre sí, los cuales pueden ser utilizados de forma conjunta o independiente” [2]. Se toma en este trabajo como definición de catálogos, un conjunto de patrones generalmente extenso con características que responden a problemas semejantes.

Por otro lado, la expresión “lenguajes de patrón”, al igual que el término “patrón” son constructos acuñados por el arquitecto Christopher Alexander. Los defensores de este enfoque para los métodos de diseño suponen que esta forma de proceder ayuda a los no expertos a resolver con éxito problemas de diseño complejo en bastantes ocasiones. Al igual que todos los idiomas, un lenguaje de patrones tiene un vocabulario, una sintaxis y una gramática [9].

Los lenguajes de patrones se refieren a colecciones de patrones que se agrupan con habilidad para formar un todo cohesionado que revela las estructuras y relaciones de sus componentes para cumplir con un objetivo compartido. Las reglas y pautas para la combinación de los patrones, definidas en un lenguaje establecen la forma de construir estructuras a todos los niveles de escala y en todos los niveles de diversidad [10].

Se considera entonces un lenguaje de patrones como una colección de patrones conectados entre sí de manera coherente y están orientados a dar respuesta a un problema específico de gran escala. Estos lenguajes tienen como propósito guiar e

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

informar al diseñador según atraviesa las relaciones de uso desde los patrones más generales a los más específicos.

Un lenguaje de patrones se compone de un léxico, vocabulario con términos afines, una gramática que establece cómo unirlos para formar estructuras sintácticas [9]. La red de relaciones que conforma un lenguaje de patrones permite al diseñador empezar desde cualquier parte del problema que ya conozca y escoger un camino para lograr su objetivo. Así, no es preciso que el diseñador comprenda de forma exhaustiva las razones que llevan a solucionar un problema para aceptar en un primer momento la estructura del lenguaje [9]. Para dotar a los lenguajes de patrones de mejor entendimiento y usabilidad pueden enriquecerse con herramientas como mapas, guías de uso y ejemplos.

1.3 Herramientas que constituyen soluciones similares

La gestión de patrones de diseño de recursos educativos es un campo novedoso en el cual no se identificó alguna herramienta que realizase el tratamiento adecuado a los catálogos y lenguajes de patrones, en correspondencia con las necesidades de los desarrolladores.

En consecuencia, se estudian aplicaciones informáticas que se utilizan para gestionar patrones de diseño:

TELL:

Es un proyecto que proporciona patrones para el aprendizaje colaborativo en red, financiado por la Unión Europea dentro de su programa e-learning (DG/EAC e-learning Initiatives). Se enfoca en la aproximación pedagógica y didáctica del aprendizaje colaborativo apoyado por ordenador. Incluye dentro de sus objetivos el de apoyar la comprensión de los procesos de aprendizaje que suceden en entornos e-learning mediante patrones de diseño y potenciar el diseño de nuevas herramientas tecnológicas efectivas para aprendizaje colaborativo.

Pedagogical Patterns Project (PPP)¹

Fue una de las iniciativas pioneras en la producción de colecciones de patrones relacionados con diversos aspectos de la enseñanza y el aprendizaje, por ejemplo, las pautas de seminarios, aprendizaje activo, el aprendizaje experimental, retroalimentación, etc.

¹ www.pedagogicalpatterns.org

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

El proyecto se inició en 1996 con el objetivo de recoger y difundir experiencias sobre objetos tecnológicos para la enseñanza y aprendizaje. La captura de las experiencias se hace en un formato homogéneo, de fácil acceso, para lo que consideraron los patrones como un método eficaz. En varios trabajos se ha señalado su relevancia y resultados, estos últimos enmarcados desde sus inicios.

Actualmente han comenzado a darse cuenta de que los patrones en aislamiento solo proporcionan mejoras incrementales a los sistemas de software, las organizaciones y los procesos. Los lenguajes de patrones, por otro lado, prometen mejoras fundamentales y duraderas.

E-LEN²

Lanzado en el año 2008, proporciona patrones para recursos de aprendizaje, LMS, aprendizaje colaborativo, aprendizaje adaptativo, etc. E-LEN es una red europea de instituciones con experiencia en e-learning. La red fue establecida para compartir y desarrollar información y diseñar patrones con respecto al e-learning. Una actividad importante de la red es la difusión del diseño patrones a las partes interesadas. Cuenta con un repositorio de patrones de diseño de e-learning. Estos patrones se han desarrollado conjuntamente dentro de los proyectos E-LEN. Este repositorio es una entidad activa que se amplía constantemente con la adición de nuevos patrones de diseño.

Dentro de esta comunidad también se ha comenzado a explorar los posibles usos de los lenguajes de patrones en el campo de la pedagogía. Varios patrones relacionados conceptualmente pueden juntos formar un lenguaje de patrones, que puede ayudar a resolver un grupo de problemas interrelacionados.

CRODA

Es una herramienta de autor de recursos educativos abiertos, específicamente de objetos de aprendizaje, que se ha estado desarrollando en la Universidad de las Ciencias Informáticas. Dicha herramienta actualmente se encuentra en la versión 3.0, en la cual incorpora un módulo que permite la creación, administración y adaptación de los patrones de diseño de recursos educativos.

² www2.tsip.no/E-LEN

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

Tabla 1 Comparación entre herramientas similares

	TELL	PPP³	E-LEN	CRODA
Creación de Patrones	No	No	No	Sí (Versión 3.0)
Estructuración de catálogos	No	No	No	No
Estructuración de lenguajes	No	No	No	No
Repositorio de Patrones	Sí	Sí	Sí	Sí
Relaciones entre patrones	No	No	Sí	No
Modelado visual	No	No	No	No
Descripción de metadatos	No	No	No	Sí

En base a los sistemas caracterizados anteriormente se establece una comparación mostrada en la tabla 2. En función de lo observado se puede afirmar que las herramientas analizadas anteriormente permiten la creación de patrones de diseño, la inserción de imágenes, multimedias y facilitan la exportación de su contenido en diferentes formatos.

Para establecer la comparación se tuvieron en cuenta las funcionalidades que permiten la creación de los patrones y por consiguiente de sus colecciones. Dichas funcionalidades constituyen la base del sistema a desarrollar, ya que trata el tema de la gestión de catálogos y lenguaje de patrones, características que no se encontraron en las herramientas estudiadas.

Resumiendo, los problemas encontrados en el diagnóstico realizado a dichas herramientas son los siguientes:

- No permiten la creación de nuevos catálogos y lenguajes. Solo en la herramienta de CRODA se lleva a cabo esta funcionalidad. Incidiendo así en la poca utilización de estos elementos.
- No incluyen en sus soluciones el modelado visual de las relaciones entre patrones de un lenguaje, influyendo en la falta de comprensión y utilización de los mismos.

³ Pedagogical Patterns Project

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

- No contienen elementos complementarios o metadatos de apoyo a la búsqueda de información, al almacenamiento y la reutilización en sentido general de estos catálogos y lenguajes. Aunque, en el módulo de CRODA si se utiliza la descripción de los metadatos.
- No presentan una estructura descriptiva de los campos para la creación de colecciones.

La comparación realizada también permitió identificar elementos positivos que sirven de punto de partida para la creación de un sistema que gestione los patrones de diseño y sus colecciones como se muestra a continuación:

Estructura de patrones:

Del módulo para la gestión de patrones de diseño de CRODA, se decidió utilizar la siguiente estructura descriptiva para los patrones, común además para sus colecciones. Aunque no es la única manera de estructurarlos, ya que existen otros autores que le han incluido o suprimido algunos de sus elementos:

Tabla 2 Estructura de patrones y sus colecciones

	Patrón	Catálogo	Lenguaje
Nombre	X	X	X
Descripción del problema	X	X	X
Contexto	X		
Contexto Resultante	X		
Clasificación	X	X	X
Catálogo al que pertenece	X		
Fundamento	X	X	X
Solución	X	X	X

Metadatos para la descripción de patrones:

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

Además de la estructura ya definida anteriormente para el patrón y de los propios elementos por los cuales se conformarán los catálogos y lenguajes, se establece su descripción mediante metadatos, para aprovechar los beneficios que estos proporcionan en cuanto a la reutilización. Se propone el uso del estándar IEEE LOM para la descripción de metadatos [11]. Específicamente, se incluyen como parte de la solución los metadatos definidos en la estructura de CRODA.

El uso de estos metadatos también garantiza el incremento en la velocidad de las búsquedas de estos recursos, un mejor entendimiento de ellos y un mayor respeto a los derechos de propiedad intelectual de sus autores.

Los siguientes metadatos propuestos (Tabla 1) fueron seleccionados a partir de la consideración de que aportan información relevante que no está dentro de la estructura del patrón [11].

Tabla 3 Descripción de metadatos

No.	Atributo/Categoría	Valores
1	Estado/Ciclo de vida	borrador, final, revisado, no disponible
2	Rol_Contribución/Ciclo de Vida	autor, publicador, desconocido, iniciador, terminador, validador, editor, diseñador gráfico, implementador técnico, proveedor de contenido, validador técnico, validador educacional, escritor de script, diseñador de aprendizaje, especialista en contenido, especialista en información científico-técnica
3	Versión/ Ciclo de Vida	1,2,3
4	Densidad semántica/Educacional	muy bajo, bajo, medio, alto, muy alto
5	Dificultad/Educacional	muy fácil, fácil, medio, difícil, muy difícil
6	Rol_Contribución/MetaMeta	creador, validador
7	Formato/Técnica	jpg, npg, avi, mp4, pdf

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

8	Costo/Derechos	sí, no
9	Localización/ Técnica	url, descripción

1.4 Modelado visual de las relaciones entre patrones

El Modelado Visual es el modelado de la información usando notaciones gráficas. Es el uso de notaciones de diseño textuales, gráficas, semánticamente elaboradas para capturar diseños de software. El modelado visual provee una plantilla, que al visualizarla ayuda a entender lo que se quiere crear y cómo hacerlo, permite entender la estructura o comportamiento del mismo, y documentar las decisiones que se toman [14] [15].

A continuación, se muestra un ejemplo de lenguaje y otro en el que se utiliza el modelado visual de las relaciones que componen dicho lenguaje, mostrando así la ventaja de facilita el modelado visual en cuanto

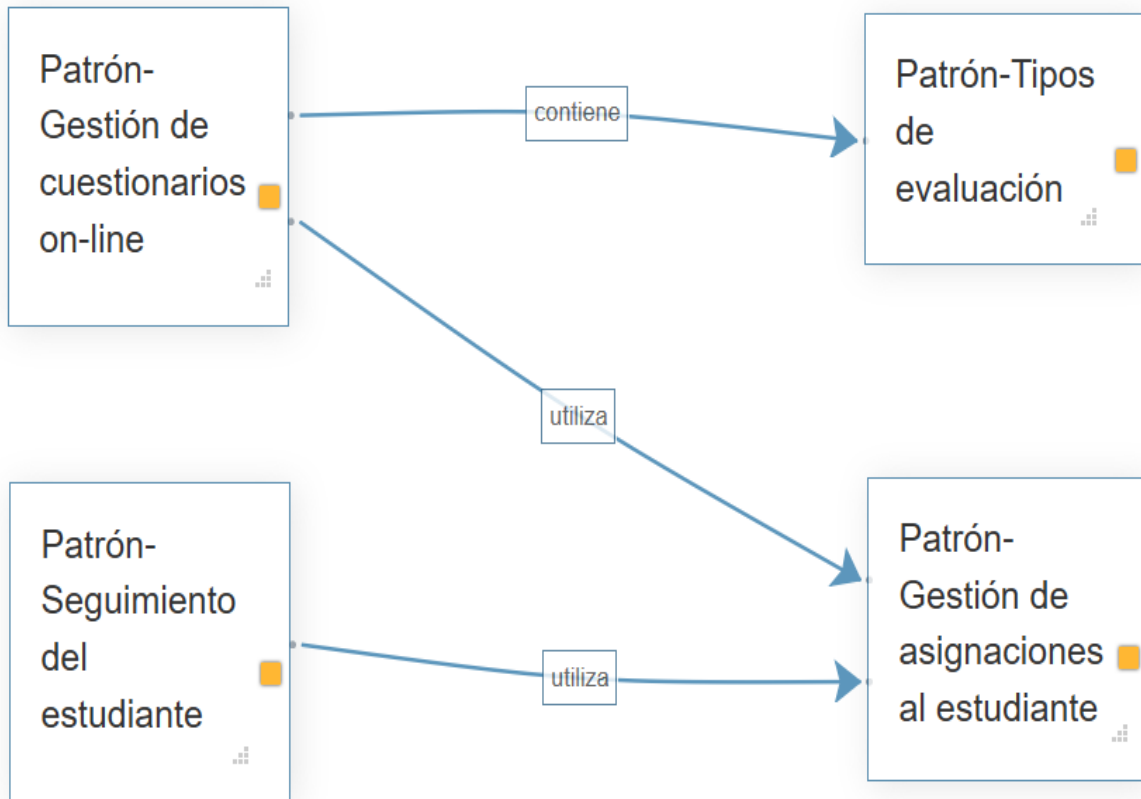
Ejemplo de lenguaje sin modelado visual:

Lenguaje para crear cuestionarios de evaluación on-line

Para gestionar los cuestionarios on-line se utilizan los siguientes patrones de diseño, primeramente, se parte del Patrón Gestión de cuestionarios on-line que explica cómo crear, entregar y calificar cuestionarios de evaluación de manera on-line. El mismo contiene el Patrón Tipos de evaluación, donde se muestra la evaluación del conocimiento de los estudiantes como una de las principales actividades del proceso de enseñanza-aprendizaje. Dicha evaluación suele ser a menudo implementada a través de ciertas preguntas o test asignados a unidades de aprendizaje, de tal forma que el estudiante pueda comprobar si verdaderamente ha asimilado los conceptos presentados en dicha unidad. Para crear estos cuestionarios se utiliza la gestión de asignaciones a cada alumno del grupo o año académico (Patrón Gestión de asignaciones al estudiante), facilitando el intercambio entre estos y sus profesores, donde la automatización de dicho proceso supone una ventaja para el estudiante al permitir llevar a cabo la evaluación de su conocimiento en cualquier momento o lugar. Para conocer los resultados de los mismos es necesario utilizar el Patrón Seguimiento del estudiante, que facilita la entrega y calificación de los resultados, sus avances y deficiencias y poder así mejorar o reconocer sus debilidades o logros respectivamente.

Ejemplo de lenguaje utilizando el modelado visual:

Ilustración 1 Lenguaje para la creación de cuestionarios on-line



1.5 Metodología de desarrollo de software

Enmarcándose en las necesidades de la UCI de converger hacia un único marco de desarrollo de software, se decide escoger como metodología a la variante AUP-UCI. Esta metodología se caracteriza por hacer un enfoque hacia los requisitos evitando la excesiva documentación que genera RUP, define un solo marco de trabajo para el desarrollo de software en la universidad y permite modelar una arquitectura que logre soportar y refinar los requisitos funcionales y no funcionales.

1.6 Lenguajes y tecnologías de programación

Para el desarrollo del sistema de gestión de patrones y colecciones de patrones se decidió utilizar como lenguaje de programación PHP, así como algunas de las tecnologías de la programación que se muestran a continuación:

PHP

El PHP (Preprocesador de Hipertextos), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor [16]. Es gratuito y de código

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

abierto. Es un lenguaje multiplataforma, funciona en casi cualquier versión de Windows y GNU/Linux, interpretado y orientado a objetos.

Al ser de código abierto, cuenta con una gran cantidad de desarrolladores que contribuyen a solucionar problemas que se presentan, como fallos de funcionamiento o brechas de seguridad [17].

Ofrece soporte a los gestores de base de datos más importantes, tales como Oracle, MySQL, PostgreSQL y SQL Server [18]. Permite crear páginas web dinámicas, con una gran seguridad de los formularios, manipula las cookies y procesa la información de foros.

Entre sus ventajas se encuentran la creación de videos flash, archivos en formato PDF y la elaboración de imágenes [19]. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutarlo y enviar el resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable [19].

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define por estar orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico [24].

Características de JavaScript

- Es simple, no hace falta tener conocimientos avanzados de programación para poder hacer un programa en JavaScript [22].
- Maneja objetos dentro del sistema y sobre ese objeto se definen diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades y modificación de archivos [24].
- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto se puede cambiar totalmente el aspecto del sistema al gusto del usuario, evitando tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario [22].

Hojas de Estilo en Cascada

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

Las hojas de estilo en cascadas, por sus siglas en inglés Cascading Style Sheets (CSS) es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML.

Las hojas de estilos en cascada (CSS) le otorgan el control sobre la presentación de las páginas web. Mediante CSS, puede colocar y definir con precisión la apariencia de los elementos de una página web.

Una CSS puede ser externa, interna o en línea, relativa a una página web, y una página web puede usar uno o varios de estos tipos de CSS a la vez. En general, los estilos que se definen en una CSS en línea tienen prioridad sobre los de una CSS interna o externa, y los estilos de una CSS interna tienen prioridad sobre los de una CSS externa [22].

1.7 HTML

HTML es una colección de estándares para el diseño y desarrollo de páginas web. Ésta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella [20]. Se trata de un sistema para formatear los campos de las páginas, así como hacer algunos ajustes a su aspecto. Con HTML, los navegadores como Firefox, Chrome, Explorer, Safari, entre otros, pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto [21]. Permite una mayor interacción entre las páginas web y contenido media (video, audio, entre otros) así como una mayor facilidad a la hora de codificar el diseño básico [21].

El HTML (*Hyper Text Markup Language*) es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento [22].

Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, vídeo, etc., por lo que el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores (programas que permiten visualizar las páginas web) [23].

1.8 Framework de desarrollo

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. A continuación, se muestra los utilizados en el sistema a desarrollar:

Bootstrap

Bootstrap es un framework de CSS que permite maquetar el sitio web en minutos, sin tocar una sola línea de CSS [25]. Simplifica el proceso de creación de diseños web combinando CSS y JavaScript [26].

Algunas de sus características más importantes son [26] [27]:

- Fácil e intuitivo: En vez de estar modificando grandes cantidades de código CSS se maqueta directamente.
- Optimizado para dispositivos móviles: Provee las reglas CSS necesarias para que la aplicación se adapte a cualquier resolución de pantalla.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo, JQuery.
- Ofrece un diseño sólido usando estándares como CSS3/HTML5.

JQuery

JQuery es una framework para el lenguaje de JavaScript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acordes a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos [28].

Como es una biblioteca de JavaScript, permite la validación de formularios y despliega contenido dinámico. Se ejecuta desde la parte del cliente. Entre las ventajas más sobresalientes se encuentran [28]:

- Compatible con todos los navegadores modernos
- Utiliza sintaxis parecidas a CSS
- Funciona con series de elementos

1.9 Gestores de base de datos PostgreSQL

Es un gestor de base de datos que se caracteriza por su robustez en la concepción de base de datos. Funciona muy bien con grandes cantidades de información, y permite una alta concurrencia de usuarios accediendo al sistema a la misma vez [29].

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

PostgreSQL trata una gran diversidad de datos que van desde números extremadamente grandes hasta direcciones IP, cadenas de caracteres, fechas, horas, entre otros. Utiliza como encriptado el SSL, es fácil de administrar y es multiplataforma [29].

Es un gestor de código extensible, que permite a los usuarios acceder a él, mejorarlo y adaptarlo a sus propias necesidades. Es compatible con lenguajes de programación tales como PHP y Java. Maneja de forma correcta las vistas, los disparadores, otorga privilegios a diferentes usuarios y tiene soporte completo para claves extranjeras y uniones. Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales. Otra de sus bondades es que maneja multiprocesos para garantizar la estabilidad del sistema [29].

1.10 Entorno de Desarrollo Integrado Netbeans

La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones [30].

Netbeans permite, mediante los paquetes que lo integran, trabajar con PHP5 en el desarrollo de aplicaciones web. Es uno de los Entornos de Desarrollo Integrado (IDE) por excelencia para trabajar con múltiples lenguajes de programación de código abierto. Puede ser usado para desarrollar aplicaciones para diversos dominios y con diferentes características, es multiplataforma, de código abierto, admite la importación de librerías y paquetes, y puede administrar el almacenamiento, guardando y cargando cualquier tipo de datos [30].

Tiene soporte para crear interfaces gráficas de forma visual, crear aplicaciones para móviles y desarrollar aplicaciones web [31].

1.11 Lenguaje de modelado UML

El UML, por sus siglas en inglés, Unified Modeling Language: es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software [32]. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL SOBRE LOS PATRONES DE DISEÑO DE RECURSOS EDUCATIVOS

lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables [33].

Por la importancia del mismo se utiliza como herramienta CASE para el modelado, el Visual paradigm, utilizado en este caso para representar los artefactos y diagramas correspondientes a la investigación mediante el UML [35].

1.13 Servidor web Xampp

XAMPP es una distribución de Apache completamente gratuita, software libre. Permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Es un software que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, un servidor de FTP FileZilla [36].

Conclusiones parciales

Una vez terminado el estudio del arte de los procesos involucrados en la gestión de los patrones de diseño de recursos educativos quedo demostrado la importancia que resalta la agrupación de los patrones de diseño en la creación de los recursos educativos, sin embargo, las herramientas estudiadas presentan limitaciones para gestionar estas colecciones, afectando así al intercambio de experiencias entre los diseñadores. Por estas razones se hace evidente la necesidad de desarrollar un sistema que permita la gestión de las colecciones de patrones de diseño. La investigación permitió constatar que el modelado visual de los lenguajes de patrones brindaría una mejor comprensión y utilización sobre los mismos, en cuanto a la relación establecida entre los patrones de diseño.

Para ello se utilizará como metodología que guíe el proceso de desarrollo a la variante AUP-UCI, como entorno de desarrollo integrado el Netbeans y como gestor de base de datos PostgreSQL, como lenguaje de programación se utilizará PHP, como servidor web el Xampp, así como la herramienta Visual Paradigm utilizando la notación UML para representar algunos artefactos que genera la metodología seleccionada.

Capítulo 2: Solución al problema planteado

Introducción

El desarrollo de un software de cualquier índole ha demostrado ser una actividad compleja, dado que consta de muchas partes que deben funcionar de forma correcta entre ellas para brindarle al cliente un producto de calidad. Dicho proceso de desarrollo ha ido perfeccionándose con el tiempo, y han surgido una serie de mecanismos que garantizan un producto libre de errores y que cuente con la satisfacción del cliente. Entre esos mecanismos están: definición de roles, artefactos, asignación de responsabilidades, modelado del negocio y sus partes, por último, identificación y documentación de los requerimientos del sistema. Algunos de estos mecanismos se emplearán para darle solución al problema científico planteado en el presente documento.

Por lo tanto, en el presente capítulo se describe la propuesta de solución al problema planteado, se utiliza el modelo de dominio para describir los principales conceptos asociados al dominio del problema y sus relaciones, así como la asignación de responsabilidades a cada rol del sistema. Se realiza la captura de los requisitos funcionales y no funcionales del producto, las historias de usuario para la descripción de los mismos, y el diagrama de Entidad-Relación.

2.1 Propuesta de solución

Para dar solución a la problemática planteada en la presente investigación se propone la implementación de SGCL (Sistema para la Gestión de Catálogos y Lenguajes de patrones de diseño), un sistema que servirá de ayuda a los diseñadores de patrones de diseño, facilitando su labor en la creación de RE. El sistema es capaz de brindar un catálogo de elementos reutilizables y de esta manera evitar la búsqueda de soluciones a problemas solucionados anteriormente. Estandarizar el modo en que se realiza el diseño, facilitar el aprendizaje de las nuevas generaciones de diseñadores y estipular un vocabulario común entre estos.

Concretamente SGCL, permitirá gestionar los patrones de diseño de RE, la estructuración de los catálogos y lenguajes de patrones, la creación de catálogos de patrones para soluciones a problemas similares y la realización de un lenguaje de patrones que permitirá relacionar los patrones de diseño, dicho lenguaje podrá estar descrito solo con palabras o representado mediante el modelado visual. Dicho modelado visual de las relaciones entre patrones en un lenguaje facilita el entendimiento y el razonamiento dinámico de los

usuarios. Adicionalmente, el usuario podrá enviar patrones, catálogos y lenguajes a revisión, eliminar alguno que esté en revisión y mostrar los que hayan sido publicados.

2.2 Modelado del negocio

El modelado del negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito [38.]

Para comprender mejor la lógica del negocio, es necesario elaborar un modelo del dominio que abarque los principales conceptos a tratar, así como definir los roles principales que interactuarán con el sistema que se quiere implementar.

2.2.1 Roles definidos para el negocio

Tabla 4 Roles del negocio

Rol	Descripción
Diseñador de patrones	Es la persona encargada de crear los patrones y sus colecciones
Revisor	Es quien realiza una revisión a los patrones y sus colecciones, quien avale su calidad para poder publicarlos.

2.2.2 Modelo de dominio

El modelo del dominio captura, comprende y describe los objetos más importantes dentro del contexto de un sistema. Se describe mediante diagramas de UML, especialmente mediante diagramas de clases. Los objetos del dominio o clases pueden obtenerse a partir de una especificación de requisitos o mediante la entrevista con los expertos del tema [39].

Descripción de los conceptos del modelo de dominio

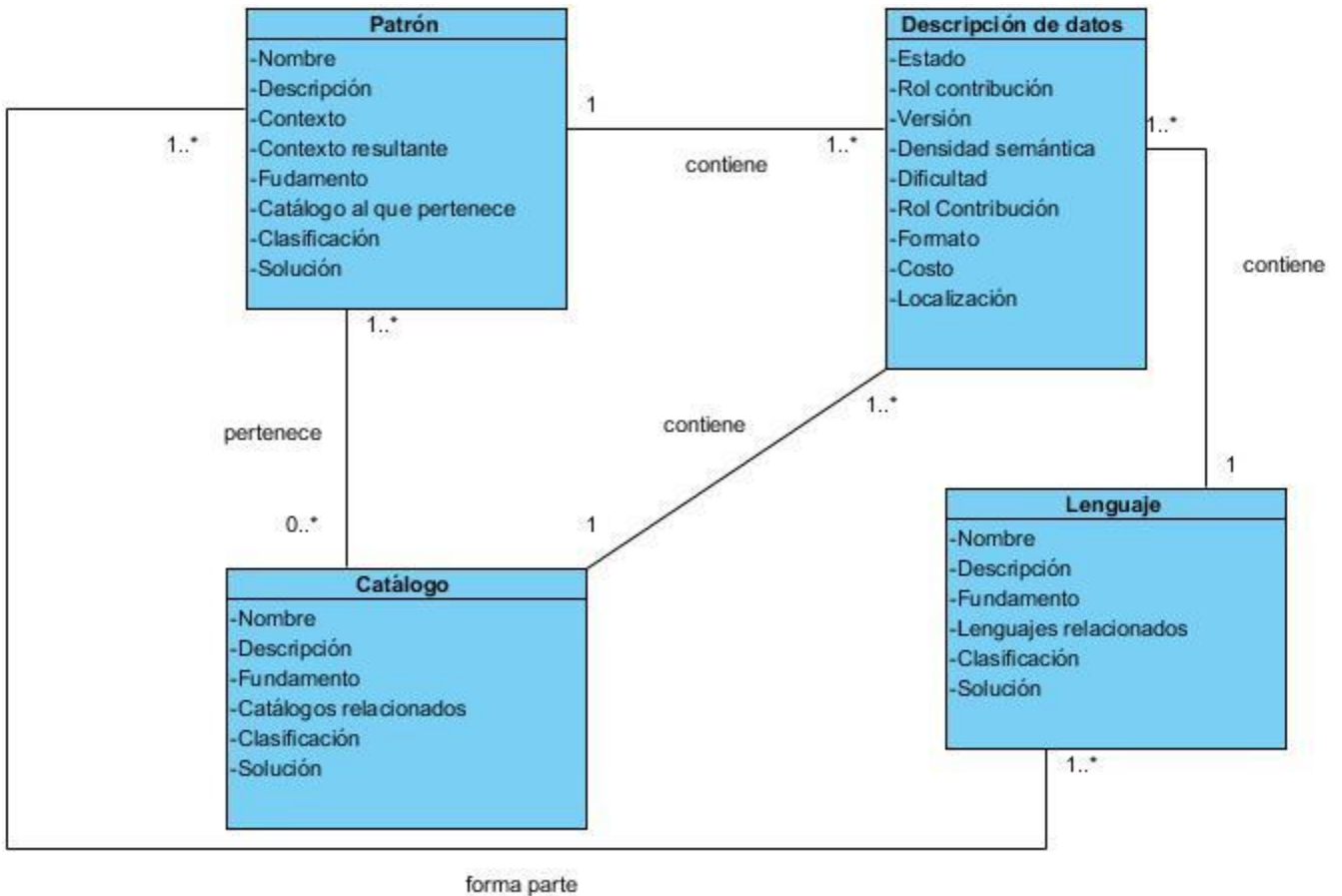
A continuación, se esclarecen los conceptos asociados al dominio, definiendo los términos más importantes que serán de utilidad para la posterior confección del modelo de dominio.

- Patrón: Es una guía, plantilla o secuencia de pasos que constituyen una solución factible a un problema determinado.
- Metadatos: Son datos acerca de datos, de cualquier tipo y medio. Son datos que describen otros datos.

- Catálogo de patrón: Agrupa a los patrones que dan solución a un mismo problema.
- Lenguaje de patrón: Refleja las relaciones que existen entre patrones. Un patrón puede estar relacionado con una lista de patrones que contribuyen a encontrar una solución, y a la vez él estar relacionado con otro patrón.

El modelo de dominio que a continuación se describe constituye una representación gráfica de los principales conceptos que forman parte de SGCL.

Ilustración 2 Modelo de dominio



2.3 Roles definidos para el sistema

Tabla 5 Roles del sistema

Rol	Descripción
Usuario no registrado	Es quien no ha sido autenticado en el sistema, solo puede consultar el material

	publicado y exportarlo si así lo desea.
Diseñador de patrones	Es la persona que posee una cuenta en el sistema y accede a través de esta para gestionar los patrones y sus colecciones.
Administrador	Persona que posee todos los permisos requeridos para trabajar con el sistema, es quien se encarga de su configuración y control.
Revisor	Es quien posee una cuenta de usuario en el sistema y accede a través de ella para revisar los patrones de diseño de RE y sus clasificaciones, analizar aquellos que se han creado y están en espera de una revisión que avale su calidad y puedan ser publicados.

2.4 Requisitos funcionales y no funcionales

El proceso de desarrollo de software comprende en sus etapas tempranas la definición de tareas orientadas a captar las necesidades o características para satisfacer el sistema que se vaya a crear o modificar [14].

2.4.1 Descripción de los Requisitos funcionales

Los requisitos funcionales representan las funcionalidades del sistema y se modelan mediante Historias de Usuarios (HU). La descripción de requisitos se encarga de mostrar para cada requisito su descripción, donde se explica qué se espera que el sistema pueda hacer una vez que un usuario requiera darle inicio al requisito. Además, contiene la prioridad que tiene para el cliente la realización del requisito, siendo “Alta” la más importante, siguiendo en el orden de prioridad “Media” y por último “Baja”. Refleja la complejidad del requisito, que se refiere a la dificultad que puede tener un programador para implementarlo.

A continuación, se muestra la descripción de los requisitos funcionales definidos para la propuesta de solución:

Tabla 6 Descripción de los requisitos funcionales

No.	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF1	Registrar usuario	Permite al usuario crearse una cuenta en el sistema	Alta	Media
RF2	Autenticar usuario	Permite al usuario autenticarse al sistema	Alta	Media
RF3	Generar patrón	El sistema debe permitir crear un patrón teniendo en cuenta la siguiente estructura: <ul style="list-style-type: none"> • Nombre • Descripción del problema • Contexto • Contexto resultante • Fundamento • Catálogo al que pertenece • Clasificación • Solución 	Alta	Alta
RF4	Editar patron	El sistema debe permitir modificar los datos de un patrón existente	Alta	Alta
RF5	Eliminar patrón	El sistema debe permitir eliminar un patrón existente	Media	Media
RF6	Revisar patrón en listado de revisión	El sistema debe permitir, a partir de un listado de patrones en revisión, seleccionar uno para ser revisado.	Alta	Alta

CAPÍTULO 2: SOLUCIÓN AL PROBLEMA PLANTEADO

RF7	Publicar patrón	El sistema debe permitir, una vez se haya revisado el patrón, publicarlo.	Alta	Media
RF8	Enviar patrón a revisión	El sistema debe permitir enviar el patrón a revisión una vez que se ha modificado	Alta	Media
RF9	Guardar patrón sin terminar	El sistema debe permitir guardar un patrón que no se haya terminado para ser concluido posteriormente	Alta	Alta
RF10	Mostrar patrón en revisión	El sistema debe mostrar los patrones que están en revisión y permitir que el revisor los revise.	Alta	Alta
RF11	Mostrar patrón publicado	El sistema debe mostrar los datos de un patrón publicado.	Alta	Media
RF12	Buscar patrón	El sistema debe permitir buscar un patrón en el listado de patrones publicados a partir de un criterio determinado.	Media	Media
RF13	Listar patrón almacenado en el sistema	El sistema brindará la opción de listar los patrones que están almacenados en el sistema	Alta	Alta
RF14	Mostrar patrón en edición	Muestra un listado de patrones que se encuentran en proceso de edición	Alta	Alta
RF15	Insertar comentario al patrón	Permite insertarle un comentario al patrón una vez revisado.	Media	Media

CAPÍTULO 2: SOLUCIÓN AL PROBLEMA PLANTEADO

RF16	Enviar patrón a revisión una vez editado	El sistema permite enviar el patrón a revisión una vez que se ha editado.	Alta	Baja
RF17	Reutilizar patrón	El sistema debe permitir, a partir de un listado de patrones publicados, seleccionar uno para ser reutilizado	Media	Alta
RF18	Enviar a revisión un patrón luego de rehusado	El sistema permite enviar a revisión un patrón luego de ser rehusado.	Media	Media
RF19	Editar patrón luego de ser rehusado	El sistema permite modificar un patrón luego de ser rehusado.	Media	Media
RF20	Generar catálogo	El sistema debe permitir crear un catálogo teniendo en cuenta la siguiente estructura: <ul style="list-style-type: none"> • Nombre • Contexto • Descripción • Fundamento • Contexto resultante • Catálogos relacionados • Índice de patrones 	Alta	Alta
RF21	Editar catálogo	El sistema debe permitir modificar los datos de un catálogo existente.	Alta	Alta
RF22	Eliminar catálogo	El sistema debe permitir eliminar un catálogo existente.	Alta	Media

CAPÍTULO 2: SOLUCIÓN AL PROBLEMA PLANTEADO

RF23	Revisar catálogo en listado de revisión	El sistema debe permitir, a partir de un listado de catálogos en revisión, seleccionar uno para ser revisado.	Alta	Alta
RF24	Publicar catálogo	El sistema debe permitir, una vez se haya revisado el catálogo, publicarlo	Alta	Alta
RF25	Enviar catálogo a revisión	El sistema debe permitir enviar el catálogo a revisión una vez que se ha modificado.	Media	Media
RF26	Guardar catálogo sin terminar	El sistema debe permitir guardar un catálogo que no se haya terminado para ser concluido posteriormente	Baja	Baja
RF27	Mostrar catálogo en revisión	El sistema debe mostrar los catálogos que están en revisión y permitir que el revisor los revise.	Alta	Media
RF28	Mostrar catálogo publicado	El sistema debe mostrar los datos de un catálogo publicado.	Alta	Media
RF29	Buscar catálogo	El sistema debe permitir buscar un catálogo en el listado de catálogos publicados a partir de un criterio determinado.	Baja	Media
RF30	Listar catálogo almacenado en el sistema	El sistema brindará la opción de listar los catálogos que están almacenados en él.	Alta	Alta

CAPÍTULO 2: SOLUCIÓN AL PROBLEMA PLANTEADO

RF31	Mostrar catálogo en edición	Muestra un listado de catálogos que se encuentran en proceso de edición	Alta	Media
RF32	Insertar comentario al catálogo	Permite insertarle un comentario al catálogo una vez revisado.	Alta	Baja
RF33	Enviar catálogo a revisión una vez editado	El sistema permite enviar el catálogo a revisión una vez que se ha editado	Media	Media
RF34	Reutilizar catálogo	El sistema debe permitir, a partir de un listado de catálogos publicados, seleccionar uno para ser reutilizado.	Alta	Media
RF35	Enviar a revisión un catálogo luego de rehusado	El sistema permite enviar a revisión un catálogo luego de ser rehusado.	Media	Media
RF36	Editar catálogo luego de ser rehusado	El sistema permite modificar un catálogo luego de ser rehusado.	Media	Media
RF37	Crear lenguaje	El sistema debe permitir crear un lenguaje teniendo en cuenta la siguiente estructura: <ul style="list-style-type: none"> • Nombre • Contexto • Tipo de relación • Fundamento • Contexto resultante • Lenguajes relacionados 	Alta	Alta

		<ul style="list-style-type: none"> • Relaciones entre patrones 		
RF38	Editar lenguaje	El sistema debe permitir modificar los datos de un lenguaje existente	Alta	Alta
RF39	Eliminar lenguaje	El sistema debe permitir eliminar un lenguaje existente	Alta	Alta
RF40	Revisar lenguaje en listado de revisión	El sistema debe permitir, a partir de un listado de lenguajes en revisión, seleccionar uno para ser revisado	Alta	Alta
RF41	Publicar lenguaje	El sistema debe permitir, una vez se haya revisado el lenguaje, publicarlo.	Alta	Alta
RF42	Enviar lenguaje a revisión	El sistema debe permitir enviar el lenguaje a revisión una vez que se ha modificado	Alta	Alta
RF43	Guardar lenguaje sin terminar	El sistema debe permitir guardar un lenguaje que no se haya terminado para ser concluido posteriormente	Alta	Alta
RF44	Mostrar lenguaje en revisión	El sistema debe mostrar los lenguajes que están en revisión y permitir que el revisor los revise.	Alta	Alta
RF45	Mostrar lenguaje publicado	El sistema debe mostrar los datos de un lenguaje publicado.	Alta	Media
RF46	Buscar lenguaje	El sistema debe permitir buscar un lenguaje en el	Alta	Media

		listado de lenguajes publicados a partir de un criterio determinado.		
RF47	Listar lenguaje almacenado en el sistema	El sistema brindará la opción de listar los lenguajes que están almacenados en él.	Alta	Alta
RF48	Mostrar lenguaje en edición	Muestra un listado de lenguaje que se encuentran en proceso de edición.	Alta	Baja
RF49	Insertar comentario	Permite insertarle un comentario al lenguaje una vez revisado.	Alta	Baja
RF50	Enviar lenguaje a revisión una vez editado	El sistema permite enviar el lenguaje a revisión una vez que se ha editado	Media	Media
RF51	Reutilizar lenguaje	El sistema debe permitir, a partir de un listado de lenguajes publicados, seleccionar uno para ser reutilizado.	Alta	Media
RF52	Enviar a revisión un lenguaje luego de rehusado	El sistema permite enviar a revisión un lenguaje luego de ser rehusado.	Alta	Alta
RF53	Editar lenguaje luego de ser rehusado	El sistema permite modificar un lenguaje luego de ser rehusado.	Alta	Alta

RF54	Añadir metadatos al patrón	El sistema debe permitir añadir metadatos teniendo en cuenta la siguiente estructura: <ul style="list-style-type: none"> • Estado • Rol contribución • Versión • Densidad semántica • Dificultad • Rol contribución • Formato • Costo • Localización 	Alta	Alta
RF55	Añadir metadatos al catálogo	El sistema debe permitir añadir metadatos teniendo en cuenta la siguiente estructura: <ul style="list-style-type: none"> • Estado • Rol contribución • Versión • Densidad semántica • Dificultad • Rol contribución • Formato • Costo • Localización 	Alta	Alta
RF56	Añadir metadatos al lenguaje	El sistema debe permitir añadir metadatos teniendo en cuenta la siguiente estructura: <ul style="list-style-type: none"> • Estado • Rol contribución • Versión • Densidad semántica • Dificultad • Rol contribución 	Alta	Alta

		<ul style="list-style-type: none"> • Formato • Costo • Localización 		
RF57	Exportar a PDF	El sistema debe permitir exportar a pdf el elemento que desee el usuario	Alta	Alta
RF58	Minimizar barra	El sistema debe permitir minimizar la barra del menú lateral	Baja	Baja
RF59	Maximizar barra	El sistema debe permitir maximizar la barra del menú lateral	Baja	Baja
RF60	Asignar rol	El sistema debe permitir asignar un rol a cada usuario autenticado.	Alta	Alta
RF61	Asignar revisor	El sistema debe permitir asignar el rol revisor al elemento correspondiente	Alta	Alta

2.4.2 Especificación de los requisitos funcionales

Las historias de usuarios son la forma en que se especifican en las metodologías ágiles los requisitos del sistema, las mismas no deben ser descritas en más de tres líneas e idealmente es el cliente quien las redacta y prioriza. Por tanto, serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica.

En el cuarto escenario de AUP-UCI, los requisitos se especifican utilizando historias de usuario, por lo que para la solución propuesta se describieron un total de (62) historias de usuario (Ver ANEXOS), de las cuales se presenta la correspondiente al RF3:

Tabla 7 Requisito funcional 3 Generar patrón

--

Número: 3	Nombre del requisito: Generar patrón
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir generar un nuevo patrón.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para generar un patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>Los campos nombre y descripción son obligatorios.</p> <p>Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres</p> <p>Descripción: campo de texto que permite cualquier caracter</p> <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema debe permitir incluir y/o seleccionar los datos para incluir un nuevo patrón desplegando la opción Patrón. - Se selecciona la opción Generar patrón - Se muestra un formulario con los siguientes campos a rellenar: <ul style="list-style-type: none"> - Nombre - Descripción del problema - Contexto - Contexto resultante - Fundamento 	

- Catálogo al que pertenece
- Clasificación
- Solución
- Se define la opción confirmar
- El sistema valida los datos insertados
- Si los datos son correctos el sistema genera el patrón
- Concluye el requisito

Flujo alternativo. Datos incorrectos o incompletos

-El sistema señala los datos incorrectos o incompletos y permite corregirlos.

-Se corrigen los datos.

Flujo alternativo. El usuario cancela la acción

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

The image shows a software window titled 'SGCL' with a 'Patrón' section. Below this is a 'Generar patrón' section containing a tabbed interface with three tabs: 'Información', 'Solución', and 'Meta-Datos'. The 'Información' tab is active and contains several text input fields: 'Nombre', 'Descripción', 'Contexto', 'Contexto resultante', and 'Fundamento'. Below these is a 'Catálogos relacionados' section with a 'Listar catálogos' button. At the bottom of the form is a 'Clasificación' field. At the very bottom of the window are two buttons: 'Confirmar' and 'Cancelar'.

2.4.3 Requisitos no funcionales

Los requerimientos no funcionales son requisitos que imponen restricciones en el diseño o implementación. Son propiedades o cualidades que el producto debe tener [14]. Seguidamente, se muestran los requisitos no funcionales para la solución propuesta. El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por ocho características (portabilidad, usabilidad, seguridad, entre otras). A partir de lo que plantea este modelo se determinaron los requisitos no funcionales para el sistema SGCL.

Requisitos no funcionales de Portabilidad (Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro):

RNF1: Sistema Operativo Windows 7 o superior. Cualquier Sistema Operativo de la familia GNU/Linux.

RNF2: Navegador web Mozilla Firefox 29.0, o superior, Ópera o Google Chrome.

RNF3: Capacidad mínima de memoria RAM de PC: 512mb.

Requisitos no funcionales de Seguridad (Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos):

RNF4: Asegurar la correcta asignación de roles a los usuarios de modo que cada quien acceda exclusivamente a lo que está autorizado. Mediante la asignación de roles por parte del Administrador.

Requisitos no funcionales de usabilidad (Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario):

RNF5: Verificar que los campos obligatorios de los formularios se encuentren llenos, en caso contrario mostrar un aviso que indique que deben llenarse.

RNF6: Cuando se crea, actualiza, elimina, exporta o publica un elemento, se muestra un mensaje con el resultado de la acción.

2.5 Estilo arquitectónico

Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema, cumpliendo el objetivo de establecer una estructura para todos los componentes del mismo. Presentan un alcance global dentro de los sistemas ya que se encuentra enfocado en toda la arquitectura del software [14].

Arquitectura de llamada y retorno:

Este estilo fue aplicado por la necesidad existente de separar las funciones de la aplicación, principalmente las que se utilizan entre las interfaces y el modelo de datos, para lograr optimizar las peticiones que se realicen por parte de los usuarios. Además, permite que una interfaz de usuario pueda mostrar múltiples vistas de los mismos datos simultáneamente [14].

2.5.1 Patrón arquitectónico Modelo Vista Controlador

En la construcción de software existen diversos componentes que proporcionan una filosofía de trabajo para los equipos de desarrollo, creando una visión que unifica la forma en que los miembros del equipo ven el sistema y además impone una transformación del mismo. Entre los mencionados componentes se encuentran los estilos arquitectónicos,

patrones arquitectónicos y patrones de diseño, los cuales se encuentran estrechamente relacionados, conformando la base de la arquitectura de un software.

Para asegurar la calidad en el software aparecen los llamados patrones arquitectónicos. Estos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos [40].

Los patrones arquitectónicos heredan mucha de la terminología y conceptos de patrones de diseño, pero se centran en proporcionar modelos y métodos re-utilizables específicamente para la arquitectura general de los sistemas de información.

Los ingenieros de software se dedican a estudiar de qué manera se pueden mejorar los procesos de creación de software y una de las soluciones a las que han llegado es la arquitectura basada en capas que separan el código en función de sus responsabilidades o conceptos. Un ejemplo de este tipo de arquitectura es el patrón Modelo-Vista-Controlador (MVC), que permite a los desarrolladores de software crear aplicaciones con mayor calidad. Surge de la necesidad de crear softwares más robustos con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: el Modelo, la Vista y el Controlador [40]. Es un patrón de arquitectura de las aplicaciones de software, separa la lógica de negocio de la interfaz de usuario, facilita la evolución por separado de ambos aspectos e incrementa la reutilización y la flexibilidad.

El **Modelo** representa la información con la que trabaja la aplicación web, representando un intermediario entre la base de datos y el resto del sistema. Es el responsable del acceso directo a los datos, a través de funciones y restricciones que especifican la lógica del negocio.

La **Vista** es la representación visual de los datos que provee el sistema, permitiendo que el usuario interactúe con el mismo de forma dinámica.

El **Controlador** es quien ejecuta las acciones dentro de la lógica del sistema. Funge como un intermediario entre las peticiones del usuario y lo que el sistema debe mostrar. Esta capa basa su funcionamiento en las solicitudes del usuario, obteniendo datos del modelo y visualizándolos nuevamente al usuario en una plantilla perteneciente a la vista.

2.5.2 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares [41] [42] [43].

Patrones GRASP

Son patrones que resaltan la importancia de captar principios para la programación, a la hora de diseñar eficazmente un software orientado a objetos. Los patrones GRASP describen entre otros elementos la asignación de responsabilidades a objetos, expresados en forma de patrones. Entre los patrones GRASP utilizados se encuentran [44] [45]:

Controlador: es un patrón que sirve como intermediario entre una interfaz y el algoritmo que la implementa, de tal forma es la que recibe los datos del usuario y los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Esto se evidencia en la clase Controladora en la cual se encuentran los métodos que se encargan de la lógica del negocio y esta a su vez pide información a la clase Modelo y envía datos a la Vista.

Alta Cohesión: cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Esto se evidencia en la clase Modelo la cual solo se encarga del acceso a los datos, Vista se encarga de crear y validar el formulario de usuario y Controladora la cual se encarga de la lógica del problema.

Experto: permite asignar una responsabilidad al experto en información, a la clase que cuenta con la información necesaria para cumplir la responsabilidad. Las clases generadas contienen toda la información necesaria de la entidad que representan y sus funcionalidades, además estas generalmente se encuentran bajo el nombre `entity_nameRepository`.

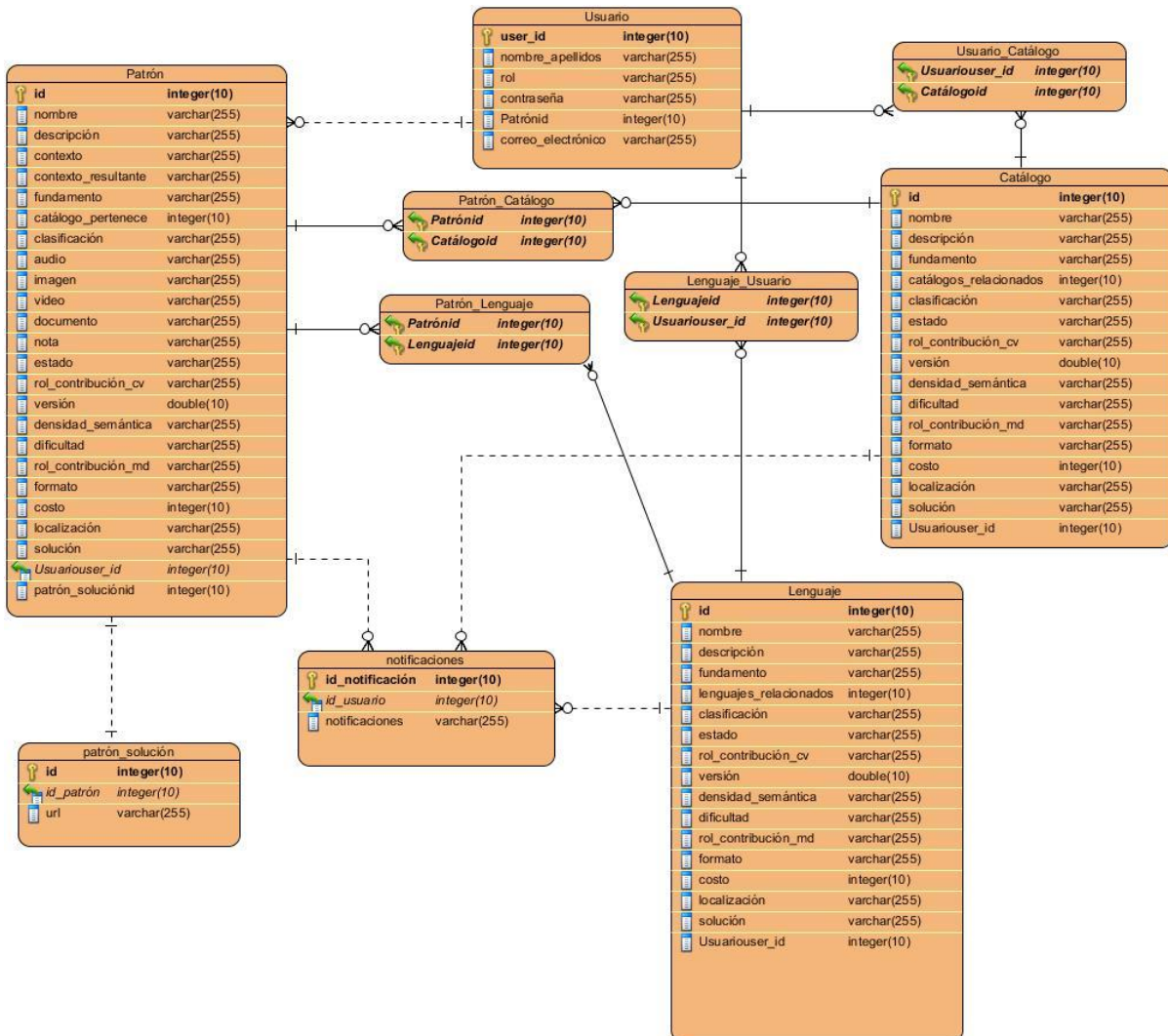
Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos [46]. Este patrón se usa con el objetivo de que la clase que maneja la información de los nuevos objetos a crear sea la responsable de crearlos.

2.6 Diagrama Entidad-Relación

El modelo de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma de gráfica a través del diagrama Entidad-Relación [47].

En la siguiente imagen se muestra el diagrama de Entidad-Relación propuesto para SGCL:

Ilustración 3 Diagrama Entidad-Relación



Conclusiones parciales

A lo largo del capítulo se analizó la propuesta de solución para sentar las bases del entendimiento del mismo y de su posterior implementación. Se definieron los requisitos

CAPÍTULO 2: SOLUCIÓN AL PROBLEMA PLANTEADO

funcionales y no funcionales del sistema atendiendo a los requerimientos identificados en el diagnóstico preliminar, donde se describieron y especificaron los requisitos funcionales mediante las historias de usuario y los prototipos de interfaz. Se identificó el patrón arquitectónico, así como los patrones de diseño, utilizado para expresar una estructura de organización base o esquema para el sistema. También, se generó el diagrama de Entidad-Relación para definir la estructura de la base de datos. Toda esa información generada contribuye a efectuar una implementación con calidad y teniendo en cuenta las exigencias del cliente.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Introducción

Una vez concluidas las disciplinas anteriormente descritas se procede al desarrollo de la implementación y las pruebas del software. El proceso de implementación documenta todos los elementos identificados durante la etapa de diseño. Por otro lado, todos los sistemas al ser desarrollados por humanos pueden presentar errores, pero es en la fase de pruebas donde los posibles errores cometidos se descubren y se corrigen.

Los objetivos del actual capítulo son: establecer las funcionalidades principales de la aplicación que serán implementadas, definir las pruebas que se realizarán una vez concluida la implementación y presentar los resultados que estas arrojan y, por último, validar la solución mediante la técnica de ladov para medir el nivel de satisfacción de los usuarios potenciales del sistema.

3.1 Modelo de implementación

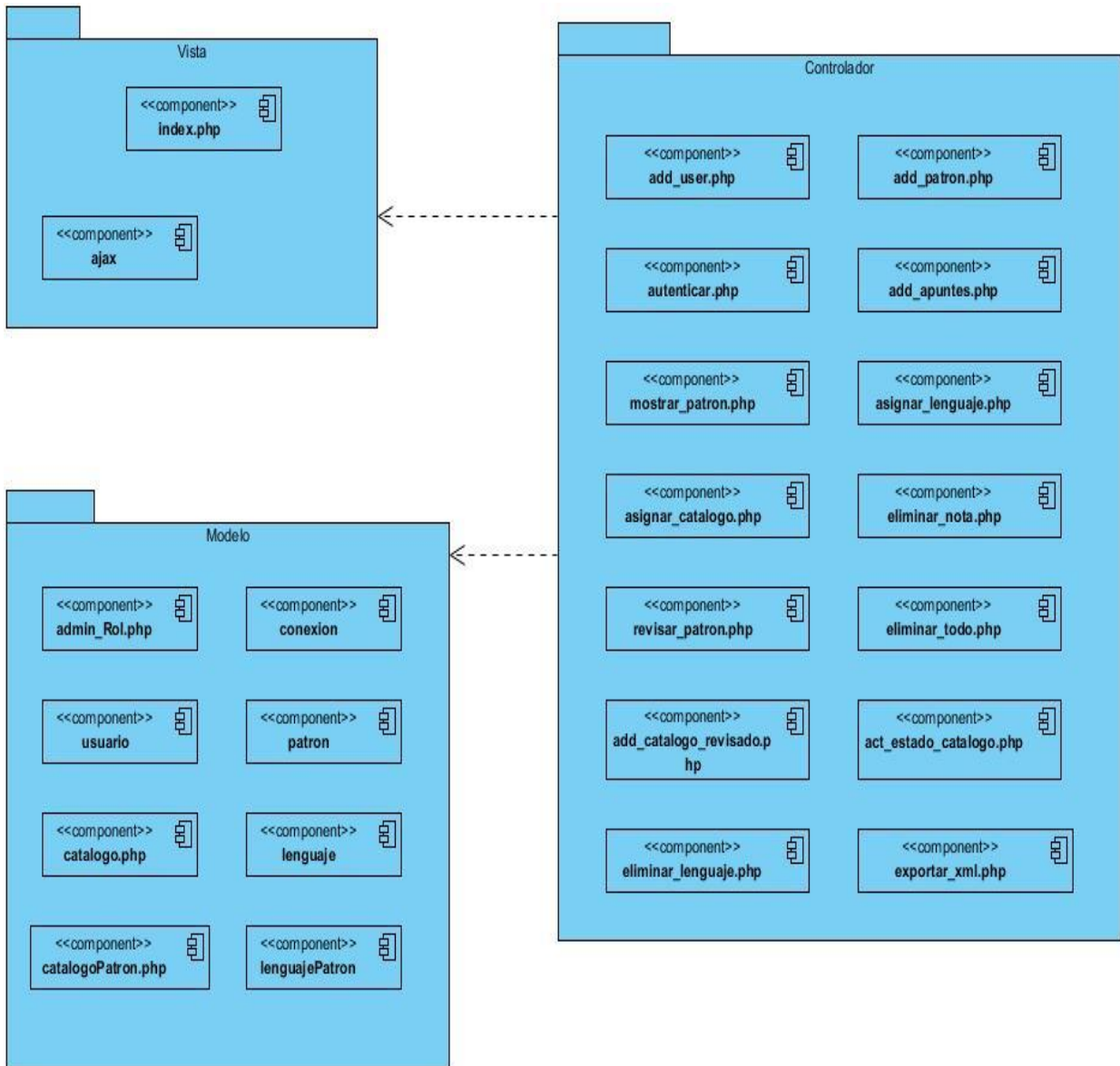
El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema [48]. Describe también la organización de los componentes según los mecanismos de estructuración y modularización disponibles en el entorno de desarrollo, el lenguaje de programación utilizado, y la dependencia entre componentes [49]. A continuación, se muestra el diagrama de componentes como parte del modelo de implementación.

3.1.1 Diagrama de Componentes

El diagrama de componentes muestra la estructura de los componentes, incluyendo clasificadores que especifican componentes, y artefactos que los implementan. También se muestra la estructura de alto nivel del modelo de implementación en términos de subsistemas de implementación, y las relaciones entre elementos de implementación [50].

El diagrama de componentes representa la estructura física de la implementación. Este diagrama permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces [51]. En resumen, muestra los elementos del diseño de un sistema de software y se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes: de software, de código binario y los ejecutables.

Ilustración 4 Diagrama de componente



3.1.2 Estándares de codificación

La utilización de los estándares de codificación en un proyecto permite que todos los participantes lo puedan entender en menos tiempo. Estos permiten tener una programación homogénea y que regula la realización de ciertos procesos, de esta forma el sistema resulta fácil de entender y mantener, independientemente de su autor. En la implementación de SGCL se utilizó el estándar PSR-1 para favorecer la legibilidad y organización al código, simplificando esfuerzos a la hora de darle mantenimiento y seguimiento a la aplicación.

Todo el código para este sistema se desarrollará en lenguaje php

El código debe seguir el estándar PSR-1

- El código PHP debe utilizar las etiquetas largas `<? php?>` o las etiquetas cortas para imprimir salida de información `<? = ?>`;
- El código debe usar 4 espacios como indentación, no tabuladores.
- Las llaves de apertura de las clases o métodos deben ir en la línea siguiente, y las llaves de cierre deben ir en la línea siguiente al cuerpo de la clase o método.
- La visibilidad debe estar declarada en todas las propiedades y métodos; `abstract` y `final` deben estar declaradas antes de la visibilidad; `static` debe estar declarada después de la visibilidad.
- Las palabras claves de las estructuras de control deben tener un espacio después de ellas, las llamadas a los métodos y las funciones no deben tenerlo.
- Las llaves de apertura de las estructuras de control deben estar en la misma línea, y las de cierre deben ir en la línea siguiente al cuerpo.
- Las constantes de PHP `true`, `false` y `null` deben estar en minúsculas.
- Los nombres de métodos no deben estar declarados con un espacio después del nombre del método. La llave de apertura debe situarse en su propia línea, y la llave de cierre debe ir en la línea siguiente al cuerpo del método. No debe haber ningún espacio después del paréntesis de apertura, y no debe haber ningún espacio antes del paréntesis de cierre.
- Las estructuras de control utilizan siempre llaves de apertura y cierre `{ }`, incluso en situaciones en las que técnicamente son opcionales. Con esto se aumenta la legibilidad del código y se disminuye la probabilidad de errores lógicos. Incluyen *if*, *for*, *foreach*, *while* y *switch*.

3.2 Pruebas de Software

La prueba es una actividad fundamental en muchos procesos de desarrollo, incluyendo el del software. Estas permiten detectar la presencia de errores que pudieran generar las entradas o salidas de datos y comportamientos inapropiados durante su ejecución. Un concepto más específico emitido por Pressman en su edición de 1998, plantea lo siguiente:

“La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación”. [14].

La fase de pruebas es una de las más costosas del ciclo de vida del software, en sentido estricto deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y el código fuente y los productos que forman parte de la aplicación de esta forma se aplican diferentes técnicas de prueba a cada tipo de producto software.

3.2.1. Métodos de pruebas de software

Pruebas de **caja blanca o estructural**: se basan en un minucioso análisis de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Su uso posibilita la obtención de casos de prueba que garantizan, al menos una vez, que sean ejecutados todos los caminos independientes de cada módulo. Posibilita ejercitar todas las decisiones lógicas en sus vertientes verdaderas y falsas. Permite, además, la ejecución de cada bucle con sus límites operacionales [52].

Por otro lado, el método de **caja negra o funcional**, se aplica la técnica de partición de equivalencia, que permite examinar los valores válidos o inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genético [53], [54].

La prueba de caja negra no es una opción frente a las técnicas de caja blanca, sino un enfoque complementario que tiene probabilidades de descubrir una clase diferente de errores de los que descubrirían con los métodos de caja blanca. Este método de prueba es realizado a nivel de sistema, es decir no se tiene ninguna relación con el código del producto [54].

3.2.2 Resultados de las pruebas de softwares

Partiendo de las pruebas de software expuestas anteriormente, se lograron corregir mediante su aplicación problemas presentes en las funcionalidades del sistema. A continuación, se muestran los resultados de dichas pruebas.

3.2.2.1 Resultados del método de caja blanca

Las pruebas unitarias permitieron localizar errores en las diferentes funcionalidades del sistema a medida que avanzaba el desarrollo del mismo. Para llevar a cabo este tipo de

pruebas se utilizó el método de caja blanca mediante el framework PHPUnit. Dichas pruebas no fueron planificadas ya que se emplearon durante todo el proceso de desarrollo del sistema.

3.2.3.2 Resultados del método de caja negra

Para verificar el cumplimiento de los requisitos funcionales establecidos en la investigación se hace uso de las pruebas de Caja Negra, teniendo en cuenta la técnica de partición por equivalencia. Dicha técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software, así como los valores de salida y descubrir de forma inmediata los errores presentes en el sistema

Para llevar este método a cabo se utilizaron los casos de pruebas con el fin de detectar la mayor cantidad de no conformidades posibles en las funcionalidades del sistema realizándose tres iteraciones de la prueba.

A continuación, se muestran los casos de prueba SC1, SC2 y SC3 en correspondencia con los requisitos RF3, RF29 y RF41 respectivamente:

Descripción General:

Permitir generar un nuevo patrón.

Condiciones de ejecución:

Estar autenticado en el sistema con el rol de diseñador de patrones.

SC1. Generar patrón

Escenario	Descripción	Nombre*	Descripción del problema*	Contexto*	Contexto resultante*	Fundamento*	Catálogo al que pertenece	Clasificación*	Solución*	Respuesta del sistema	Flujo central
EC 1.1 Opción Generar patrón.	Selección a la opción de generar un nuevo patrón.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Brinda la posibilidad de introducir o seleccionar de manera obligatoria los siguientes datos del patrón: - Nombre – Descripción del problema – Contexto – Contexto resultante – Fundamento – Clasificac	Patrón/Generar patrón/

										ión – Solución Y de forma opcional: - - Catálogo al que pertenece Permite: - Guardar los datos. - Cancelar la operación en cualquier momento –Enviar datos a revisión	
EC 1.2 Opción de Guardar los datos.	Introduce y/o seleccion a los datos del patrón y seleccion a la opción guardar	V	V	V	V	V	V	V	V	Valida los datos. Crea un nuevo patrón Muestra un mensaje de informaci	Patrón/Gen erar patrón/Guar dar

	los datos.									ón.	
EC 1.3 Opción de Cancelar	Selección a la opción de cancelar.									<i>Elimina los datos creados muestra un mensaje de información.</i>	Patrón/Generar patrón/Eliminar
EC 1.4 Opción de Enviar datos a revisión	Selección a la opción de enviar datos a revisión.									Muestra un mensaje de información.	Patrón/Generar patrón/Enviar a revisión
EC 1.5 Datos incompletos	Existen datos incompletos.	I	V	V	V	V	V	V	V	Muestra un mensaje de información. Muestra un indicador sobre los campos vacíos. <u>Regresa</u>	Patrón/Generar patrón/Guardar
		V	I	V	V	V	V	V	V		
		V	V	I	V	V	V	V	V		
		V	V	V	I	V	V	V	V		
		V	V	V	V	I	V	V	V		
		V	V	V	V	V	I	V	V		
		V	V	V	V	V	V	I	V		

		V	V	V	V	V	V	V	I	<u>al EC 1.1.</u>	
EC 1.6 Datos incorrectos	Existen datos incorrectos.	I	V	V	V	V	V	V	V	Muestra un mensaje de información. Muestra un indicador sobre los campos incorrectos. <u>Regresa al EC 1.1.</u>	Patrón/Generar patrón/Guardar
		V	I	V	V	V	V	V	V		
		V	V	I	V	V	V	V	V		
		V	V	V	I	V	V	V	V		
		V	V	V	V	I	V	V	V		
		V	V	V	V	V	I	V	V		
		V	V	V	V	V	V	I	V		
		V	V	V	V	V	V	V	I		

Una no conformidad es un fallo en el sistema de gestión de la calidad que puede producirse por varias razones: no alcanzar el nivel de aceptación establecido en un determinado indicador y errores en la documentación del sistema. Las mismas se clasifican de acuerdo al nivel de importancia en [54]:

- 1- **Significativas:** Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.
- 2- **No significativas:** Son aquellas que resultan menos visibles, que no atentan contra el cumplimiento de algún requisito.

3- **Recomendaciones:** Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

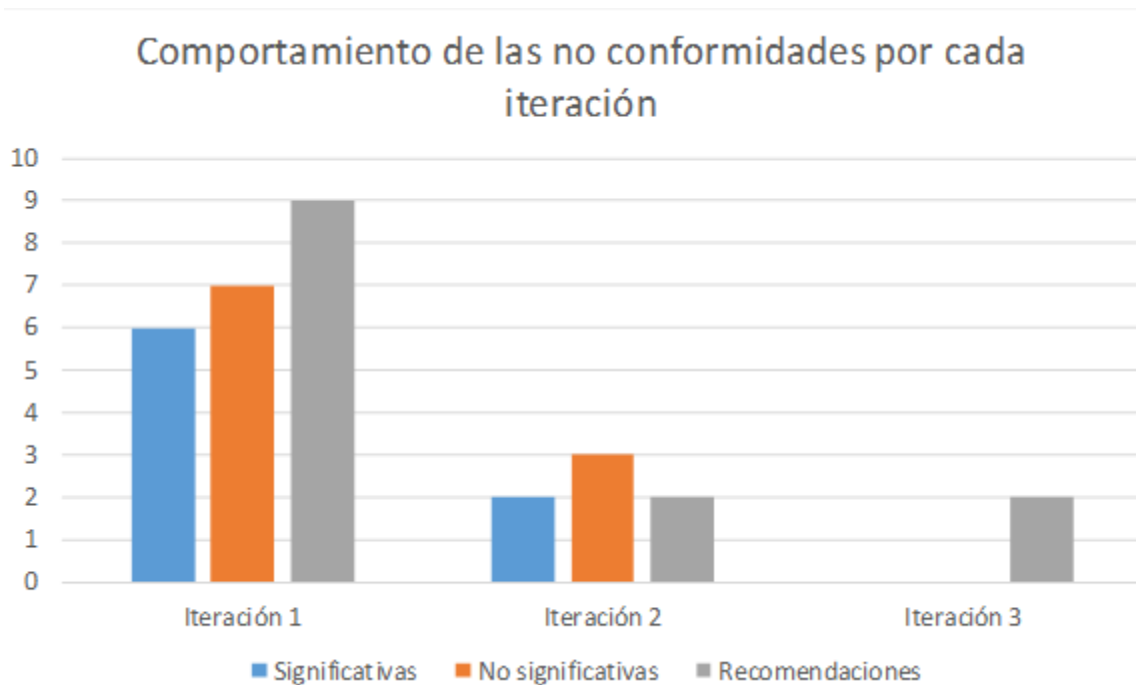
En la siguiente tabla se muestran la cantidad de no conformidades encontradas en cada iteración clasificadas de acuerdo al su nivel de significación:

Tabla 8 Resumen de las no conformidades

Significación de la no conformidad	Iteración 1	Iteración 2	Iteración 3
Significativas	6	2	0
No significativas	7	3	0
Recomendaciones	9	2	2

La información se muestra de forma más detallada en el siguiente gráfico.

Ilustración 5 Comportamiento de las no conformidades



Las no conformidades encontradas en la última iteración no constituyen deficiencias mayores para el sistema, puesto que fueron corregidas posteriormente, de ahí que no fue requerido documentar una tercera iteración.

3.3 Validación del sistema

La validación es una de las etapas más importantes en el análisis del comportamiento de un sistema. El término validación se refiere a un análisis de la calidad del sistema inteligente en su entorno real de trabajo, lo que permite determinar si el producto desarrollado satisface convenientemente las expectativas inicialmente depositadas.

Con el objetivo de realizar la validación de SGCL, se hace uso de la técnica de ladov para medir el nivel de satisfacción que podrían tener los diseñadores del aprendizaje con el uso de SGCL.

3.3.1 Aplicación de la técnica de ladov

La Técnica de ladov constituye una vía para el estudio del grado de satisfacción de implicados en el proceso de gestión de patrones de diseño de recursos educativos. Esta técnica ha sido aplicada para valorar la satisfacción en investigaciones desarrolladas en múltiples campos, varias de ellas con perfiles técnicos [56].

Dicha técnica está conformada por siete preguntas: cuatro cerradas y tres abiertas y constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" [56].

A continuación, se muestra el cuadro lógico de ladov con las tres preguntas del cuestionario referentes a la satisfacción con la propuesta de solución:

Ilustración 6 Cuadro lógico de ladov

1. ¿Utilizaría el sistema SGCL para gestionar patrones de diseño de recursos educativos y sus colecciones?		
Sí	No sé	No
2. ¿Considera usted que la implementación del modelado v de las relaciones entre patrones facilita el entendimiento y el razonamiento dinámico de los usuarios?		

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3. ¿Qué considera de los componentes diseñados para el desarrollo de SGCL y su integración?	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisfacen mucho	1	2	6	2	2	6	6	6	6
Más satisfecho que Insatisfecho	2	2	3	2	3	3	6	3	6
Me es indiferente	3	3	3	3	3	3	3	3	3
Más insatisfecho que Satisfecho	6	3	6	3	4	4	3	4	4
No me satisfacen	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada encuestado en la escala de satisfacción siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para el desarrollo de la técnica se escogió una muestra de 10 encuestados con conocimientos básicos sobre el trabajo con los recursos educativos y patrones de diseño.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla 9 Índice de satisfacción grupal ISG

Escala	Nivel de Satisfacción
+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Estos valores se emplean para obtener el Índice de Satisfacción Grupal (ISG), como puede observarse en la siguiente ecuación.

Ilustración 7 Ecuación para obtener el ISG

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

Donde A, B, C, D y E representan la cantidad de sujetos en cada una de las categorías.

Tabla 10 Nivel de satisfacción

Variables	Nivel de satisfacción	Valores
A	Máximo de satisfacción	6
B	Más satisfecho que insatisfecho	2
C	No definido y contradictorio	1
D	Más insatisfecho que satisfecho	1
E	Máxima insatisfacción	0

El cálculo del ISG arrojó un valor de 0.65, lo que indica satisfacción de usuarios potenciales con respecto al sistema propuesto.

Conclusiones parciales

Al concluir este capítulo se generaron los artefactos correspondientes a la disciplina de implementación incluyendo dentro de ellos el diagrama de componentes. Las pruebas aplicadas identificaron una serie de no conformidades con respecto al sistema que fueron documentadas y corregidas. Con la realización de la validación a SGCL se obtuvo un nivel de satisfacción adecuado de acuerdo a la técnica de Iadov, la cual se llevó a cabo mediante la encuesta a especialistas en el trabajo con RE de la Universidad de las Ciencias Informáticas.

CONCLUSIONES GENERALES

Luego de culminar el proceso de análisis, diseño, implementación y pruebas desarrollados al sistema para la creación de patrones de diseño y sus colecciones, se arribó a las siguientes conclusiones:

- En el desarrollo del marco teórico de la investigación se analizaron las herramientas que constituyen soluciones similares, permitiendo identificar características comunes entre ellas, así como las limitaciones y/o deficiencias que estas podrían tener para la creación de los patrones de diseño y de sus colecciones.
- En la fase de análisis y diseño quedó documentado cada uno de los artefactos definidos por la metodología AUP-UCI, los cuales facilitaron el desarrollo del SGCL (Sistema para la Gestión de Catálogos y Lenguajes de patrones de diseño).
- El sistema SGCL brinda la facilidad de gestionar patrones de diseño, su publicación, creación, almacenamiento y adaptación, así como la gestión de los catálogos y lenguajes de patrones, incluyendo a este último el modelado visual de las relaciones establecidas ente patrones.
- Las correctas ejecuciones de las pruebas aplicadas al sistema permitieron detectar y corregir las 31 no conformidades presentes a lo largo de 3 iteraciones hasta obtener resultados satisfactorios, que demuestran que el sistema cuenta con la calidad necesaria.

RECOMENDACIONES

Para la continuidad de la investigación se recomienda:

- Incluir el intercambio electrónico de datos mediante el uso del lenguaje XML, para permitir así la interoperabilidad técnica del sistema con otras aplicaciones.
- Incluir para el modelado visual de las relaciones entre patrones, la notación UML, que permite establecer una estructura basada en un diseño gráfico ampliamente extendido y que apoyaría la comprensión de los lenguajes de patrones.

REFERENCIAS BIBLIOGRÁFICAS

- [1] González, A. "Patrones en aprendizaje: concepto, aplicación y diseño de un patrón." Educación a Distancia, 2012, No.31.
- [2] Montero, S; Zarraonadía, T; Díaz, P y Aedo, I. «PATRONES DE DISEÑO DE ODE». [En línea]. Educación. 2011. [Accedido: 07-nov-2017] Disponible en: <http://ares.cnice.mec.es/informes/21/ contenidos/4.htm>.
- [3] Seoane, A. García, J. Patrones pedagógicos y docencia en red. Segundo Congreso Iberoamericano de Aprendizaje Mediado por Tecnología. Facultad de Estudios Superiores Zaragoza. Universidad Nacional Autónoma de México, 2014, 1ra edición: 30-47.
- [4] Calvo, P. Una metodología para el desarrollo de Patrones de Diseño de Objetos de Aprendizaje Constructivista Colaborativo. Maestría en tecnología informática aplicada en educación. Universidad Nacional de La Plata. Argentina, 2015
- [5] Hernández, Y. and Aranguren, G. Patrón tecnopedagógico: ruta de aprendizaje basado en actividades comprensivas. Revista Vínculos, 2016 Vol. 13, No. 2: 30-39.
- [6] Muñoz, H; Padilla, A. Clasificación de Recursos Educativos a través de un Sistema Basado en el Estilo de Aprendizaje y las Competencias de los Alumnos de Educación Media Superior. Anais da X Conferência Latino-Americana de Objetos e Tecnologias de Aprendizagem (LACLO 2015). Brasil.
- [7] UNESCO, «Recursos educativos abiertos | Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura», Unesco. [En línea]. Comunicación e Información, 2012. [Accedido: 09-oct-2017] Disponible en: [http://www.unesco.org/new/es/communication-and-information/access-to-knowledge/open-educational-resources/..](http://www.unesco.org/new/es/communication-and-information/access-to-knowledge/open-educational-resources/)
- [8] Fundación Universia, «Recursos educativos abiertos: estrategia para apertura y desarrollo social de la Educación Superior», universia, 24-sep-2007. [En línea]. [Accedido: 10-oct-2017] Disponible en: <http://noticias.universia.net.co/vidauniversitaria/noticia/2007/09/ 24 /246858 /recursos - educativos-abiertos-estrategia-apertura- desarrollosocialeduccion-superior.html>.
- [9] Alexander, C; Ishikawa, S and M. Silverstein. Pattern Language: Towns, Buildings, Construction. 1977

- [9] Zapata-Ros, M. "Patrones en elearning. Elementos y referencias para la formación." RED, Revista de Educación a Distancia, 2011, No. 27
- [10] Burgos, J; Galve, J. «Organización del conocimiento mediante patrones de diseño», en Actas del Congreso ISKO, España, 2001, p. 10.
- [11] Terry, Y. y Colomé, D. Utilización de metadatos para identificar problemas en diseños de recursos educativos. *Enl@ce Revista Venezolana de Información, Tecnología y Conocimiento*, 2016, 13 (2), 28-39
- [12] Monje, A «¿Qué es eXeLearning? | Tutorial manual de eXeLearning.net. El nuevo eXeLearning», exelearning. [En línea]. [Accedido: 07-nov-2017]. Disponible en: http://exelearning.net/html_manual/exe_es/qu_es_exelearning.html.
- [13] González, A «CourseLab | Entornos de aprendizaje en Hipermedia», Universidad Nacional de La Plata. [En línea]. [Accedido: 07-nov-2017]. Disponible en: [http://blogs.unlp.edu.ar/entornos/2012/11/08/courselab/..](http://blogs.unlp.edu.ar/entornos/2012/11/08/courselab/)
- [14] Pressman, R. Ingeniería de software. Un enfoque práctico, (2010), (5 ed.).
- [15] Caeiro, M., et al. E-Learning patterns: an approach to facilitate the design of E-Learnings materials. VII Congreso Iberoamericano de Informática Educativa. Mar del Plata, Argentina, 2007.
- [16] C. V. D. H. S, «¿Qué es el PHP?», Maestros del Web, 23-may-2001. [En línea]. [Accedido: 09-ene-2018]. Disponible en: <http://www.maestrosdelweb.com/phpintro/>. [17] T. R. Sánchez, «Metodología de desarrollo para la actividad productiva de la UCI». UCI.
- [18] Definición de PHP - ¿qué es PHP?», Jojooa - tecnología, marketing y crm. [En línea]. [Accedido: 09-ene-2018]. Disponible en: <https://sites.google.com/site/jojooa/informatica-tecnologia/definicion-de-php-que-es-php>.
- [19] Martínez, K «TODO SOBRE PHP: VENTAJAS Y DESVENTAJAS», Klarimartinezbenjumea, 04-abr-2011.
- [20] Diego, «HTML5: ¿Qué es? Y ¿Cómo usarlo?», TrazosWeb, 01-feb-2010. [En línea]. [Accedido: 01-dic-2018]. Disponible en: <http://www.trazos-web.com/2010/02/01/html5-que-es-y-como-usarlo/>.

- [21] barbarapvn, «Qué es HTML5», Hipertextual, 28-may-2013. [En línea]. [Accedido: 10-ene-2018]. Disponible en: <http://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>.
- [22] Gauchat, J. El gran libro de HTML5, CSS3 y Javascript, 2015.
- [23] Eguíluz, J. Introducción a XHTML, 2015.
- [24] Eguíluz, J. Introducción a JavaScript.,2015
- [25] Sheila, «Que es Bootstrap para tu web», Vullzthemediia.
- [26] T. Rodríguez, «Bootstrap, framework de twitter», Genbetadev.
- [27] A. Chávez, «Si eres desarrollador web, tienes que usar Bootstrap.», Alanchavez, 13-sep-2013.
- [28] Sergio Manzur, «¿Qué es jQuery?», Mexired, 14-oct-2013. [En línea]. [Accedido: 11-ene-2018]. Disponible en: http://www.mexired.com/blog/que-es-jquery/#.VU5__JN2L90.
- [29] Escuela Politécnica Nacional de Ecuador, «PostgreSQL: Características, limitaciones y ventajas», Postgresql, 18nov-2012.
- [30] Giménez, M «NETBEANS accesible: Información Básica sobre netbeans», Netbeansaccesible.
- [31] Balboa, M «Tutorial Java con NetBeans IDE», Tutorial Java con NetBeans IDE, 13-may-2010.
- [32] J. E. G. Cornejo, «El Lenguaje de Modelado Unificado (UML)», Docirs, ene-2008. [En línea]. [Accedido: 15-mar-2018]. Disponible en: <http://www.docirs.com/uml.htm>.
- [33] Osmosis Latina, «Importancia de UML», Osmosis Latina. [En línea]. [Accedido: 18-mar-2018] Disponible en: <http://www.osmosislatina.com/lenguajes/uml/basico.htm..>
- [34] Venete, A «Visual Paradigm para UML a II27 - Ingeniería del Software», Universidad Jaime I. [En línea]. [Accedido: 4-may-2018] Disponible en: <http://mahara.uji.es/view/artefact.php?artefact=54800&view=4648>.
- [35] Sierra, D «Visual Paradigm For Uml», Instituto Politécnico Nacional, 15-nov-2007.
- [36] Maldonado, D«Apache, el servidor Web más reconocido», *Empresa y Economía*, 20-abr-2008. [En línea]. [Accedido: 23-may-2018]. Disponible en: <http://www.empresaeconomia.es/aplicaciones-para-empresas/apache-el-servidor-web-mas-reconocido.html>.

- [37]Exes, «XML ¿Qué es? | Manual de XML», *Mundo Linux*. [En línea]. [Accedido: 10-may-2018]. Disponible en: <http://www.mundolinux.info/que-es-xml.htm>.
- [38]T.Sánchez, R «Metodología de desarrollo para la Actividad productiva de la UCI.» 21-nov-2014.
- [39] Jacobson, I; Booch, G., & Rumbaugh, J. *The Unified Software Development*, 1999.
- [40] Mestras, P «Estructura de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC)», Universidad Complutense de Madrid.
- [41]Reutilización del Software. Patrones de Diseño. Disponible en: [En línea] [Accedido: 23-may-2018] Disponible en: <http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>.
- [42]¿Qué es un Patrón de Diseño? [En línea] [Accedido: 23-mar-2018] Disponible en: <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.
- [43] Design Patterns. [En línea] [Accedido: 25-abr-2018] Disponible en: https://sourcemaking.com/design_patterns.
- [44] Using Design Patterns with GRASP. [En línea] [Accedido: 25-abr-2018] Disponible en: <https://web.cs.dal.ca/~jin/3132/lectures/dp-13.pdf>.
- [45]Diseño dirigido por responsabilidades con los patrones GRASP. [En línea] [Accedido: 25-abr-2018] Disponible en: <http://lsi.ugr.es/~mvega/isoo/algp Patrones.pdf>.
- [46]Hernández, P «Uso de patrones de arquitectura», UPM.
- [47] Storti, G; Ríos, G y Campodónico, G «Base de datos. Modelo Entidad Relación». Colegio Manuel Belgrano, 2007.
- [48]Murga, D, Vázquez, A «Versión offline y multiplataforma del Repositorio de Objetos de Aprendizaje RHODA», Universidad de las Ciencias Informáticas.
- [49]Sanz, M. (2009). *Proceso Unificado: Implementación*.
- [50]Gafas, Y y Chacón, J. «DESARROLLO DE UN PROCESO INTEGRAL DE REVISIÓN EN EL REPOSITORIO DE OBJETOS DE APRENDIZAJE RHODA.», Tesis, Universidad de las Ciencias Informáticas.
- [51]Microsft. (s.f.). Diagrama de Componentes UML. Recuperado el 11 de abril de 2017, [En línea] [Accedido: 25-abr-2018] Disponible en: <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.

REFERENCIAS BIBLIOGRÁFICAS

[52] Marick, B. New Models for test development. [En línea] [Accedido: 25-may-2018] Disponible en: <http://www.exampler.com/testing-com/writings/new-models.pdf>.

[53] Tipos de pruebas de software. [En línea] [Accedido: 16-may-2018] Disponible en: <http://es.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software>.

[54] Testing Overview and Black-Box Testing Techniques. [En línea] [Accedido: 26-abr-2018] Disponible en: <http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf>.

[55] Rodríguez, A y Maura, V. La técnica de Iadov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física . [En línea] [Accedido: 20-may-2018] Disponible en: <http://www.efdeportes.com/efd47/iadov.htm>.

Anexo 1. Historias de Usuario

RF1 Registrar usuario

Número: 1	Nombre del requisito: Registrar usuario
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir que el usuario se registre.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para registrar un usuario hay que:</p> <ul style="list-style-type: none"> - Registrarse con una cuenta nueva, no con una ya existente en el sistema. <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos de nombre y apellidos, usuario, contraseña, correo electrónico son obligatorios.</p> <ul style="list-style-type: none"> -Nombre y apellidos: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres -Usuario: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres -Contraseña: campo de texto que permite cualquier carácter -Correo electrónico: campo de texto que permite cualquier carácter <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El usuario debe rellenar los siguientes campos <ul style="list-style-type: none"> - Nombre y apellidos - Usuario - Contraseña - Correo electrónico - Si desea que se guarde su usuario y contraseña se selecciona el botón Recordarme - Se selecciona el botón Acceder - El sistema valida los datos insertados 	

- Si los datos son correctos el usuario se autentica
- Concluye el requisito

Flujo alternativo. Datos incorrectos

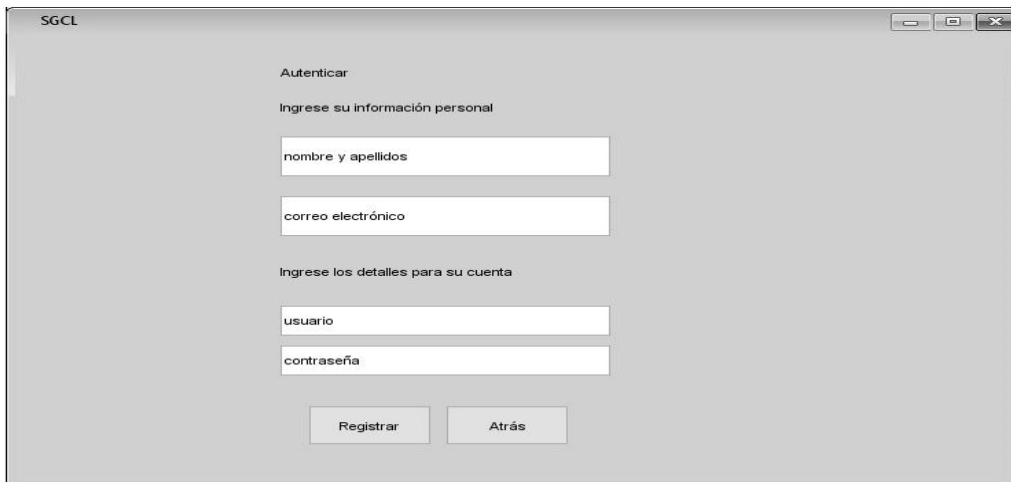
- El sistema señala los datos incorrectos y permite corregirlos.
- Se corrigen los datos.

Flujo alternativo. Campos en blanco

- El sistema señala los campos en blanco y permite rellenarlos.
- Se validan los datos.

Observaciones:

Prototipo de interfaz:



RF2 Autenticar usuario

Número: 2	Nombre del requisito: Autenticar usuario
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
Descripción:	
1- Objetivo:	
Permitir autenticar usuario.	
2- Acciones para lograr el objetivo (precondiciones y datos):	

Para autenticar usuario hay que:

- Estar registrado en el sistema.

3- Comportamientos válidos y no válidos (flujo central y alternos):

Los campos usuario y contraseña son obligatorios.

-Usuario: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres

-Contraseña: campo de texto que permite cualquier carácter

4- Flujo de la acción a realizar:

- El usuario debe rellenar los siguientes campos

- Usuario
- Contraseña

- Si desea que se guarde su usuario y contraseña se selecciona el botón Recordarme

- Se selecciona el botón Acceder

- El sistema valida los datos insertados

- Si los datos son correctos el usuario se autentica

- Concluye el requisito

Flujo alternativo. Datos incorrectos

- El sistema señala los datos incorrectos y permite corregirlos.

- Se corrigen los datos.

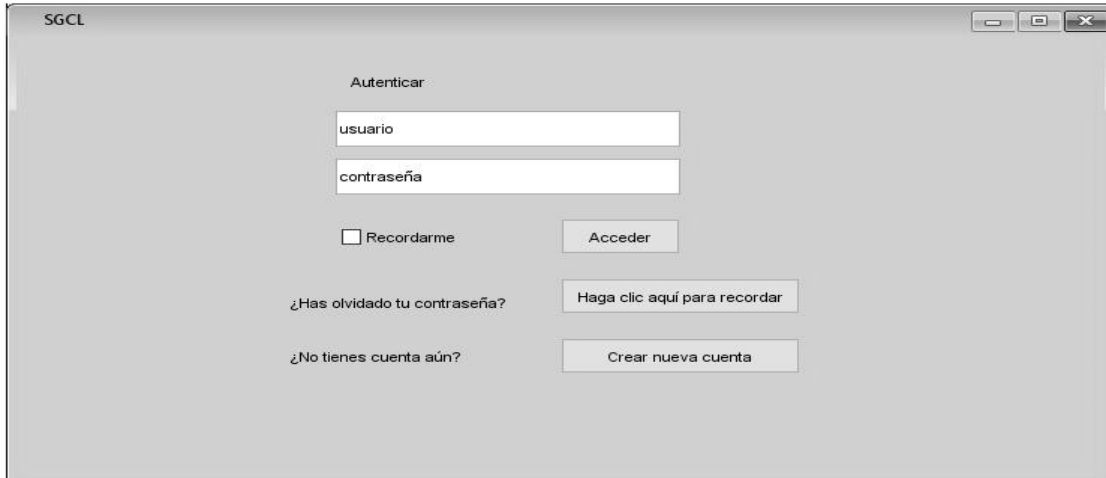
Flujo alternativo. Campos en blanco

- El sistema señala los campos en blanco y permite rellenarlos.

- Se validan los datos.

Observaciones:

Prototipo de interfaz:



RF3 Generar patrón

RF4 Editar patrón

Número: 4	Nombre del requisito: Editar patrón
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir modificar los datos de un patrón existente</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar los datos de un patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol diseñador. - Debe existir en el sistema al menos un patrón. <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios.</p> <ul style="list-style-type: none"> -Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres -Descripción: campo de texto que permite cualquier carácter <p>4- Flujo de la acción a realizar: -El sistema debe permitir editar un patrón, esta acción puede realizarse seleccionando la opción editar desplegando la opción Patrón y luego seleccionar Listar Patrón.</p>	

-Cuando el usuario edita de forma correcta los datos necesarios y selecciona la opción Actualizar, se muestra un mensaje de información de que el patrón fue editado de forma correcta.

Flujo alternativo. Datos incorrectos o incompletos

-El sistema señala los datos incorrectos o incompletos y permite corregirlos.

-Se corrigen los datos.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

SGCL

Patrón

Editar patrón

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

Clasificación

RF5 Eliminar patrón

Número: 5

Nombre del requisito: Eliminar patrón

Programador: Ernesto Alay Navarrete

Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir eliminar un patrón existente</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para eliminar un patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un patrón. <p>3- Flujo de la acción a realizar: Se despliega la opción Patrón y luego Listar Patrón, donde se seleccionan los patrones a eliminar</p> <ul style="list-style-type: none"> - El sistema permite eliminar uno o varios patrones: <ul style="list-style-type: none"> -Para eliminar varios patrones el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción Aplicar. -En caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento. - Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando. - Si selecciona la opción Aceptar se actualiza el listado y el sistema muestra un mensaje de información. <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	



RF6 Revisar patrón en listado de revisión

Número: 6		Nombre del requisito: Revisar patrón en listado de revisión	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir revisar los patrones en el listado de revisión			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para revisar los patrones en el listado de revisión hay que:			
- Estar autenticado en el sistema con el rol de Revisor.			
- Debe existir en el sistema al menos un patrón en el listado de revisión.			
3- Flujo de la acción a realizar:			
- Para revisar un patrón primeramente el revisor despliega la opción Patrón y en el Listar Patrón selecciona el patrón que desea revisar, verifica si los datos del patrón están correctos.			
Flujo alternativo. Datos incorrectos			
-Si el revisor detecta datos incorrectos o incompletos, le deja un comentario para que luego el diseñador de patrones lo pueda corregir, seleccionando la opción enviar apuntes.			

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

SGCL

Patrón

Revisar patrón

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados Listar catálogo

Clasificación

Cancelar

Comentario

Introduzca comentario

Guardar comentario Publicar Corregir

RF7 Publicar patrón

Número: 7	Nombre del requisito: Publicar patrón
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir publicar un patrón</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para publicar un patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol revisor. - El patrón debe estar revisado. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El revisor despliega en la opción Patrón, Listar Patrón, y selecciona el patrón que desea publicar, que haya sido revisado anteriormente. Para luego publicarlo mediante la opción Publicar, cuando seleccione el botón aceptar, se mostrará un mensaje para informar que el patrón se ha publicado correctamente <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

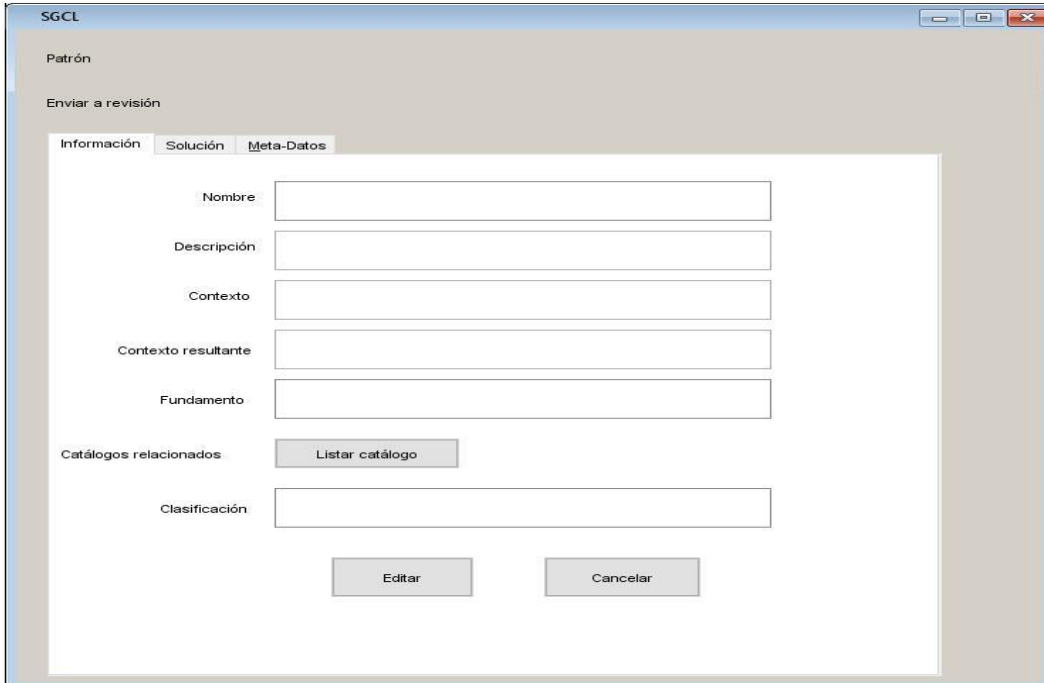
RF8 Enviar patrón a revisión

Número: 8	Nombre del requisito: Enviar patrón a revisión
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar patrón a revisión</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para enviar patrón a revisión hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol diseñador. - Debe existir en el sistema al menos un patrón. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El usuario selecciona el patrón que desea enviar a revisión desplegando Patrón, Listar Patrón respectivamente. El patrón es enviado a lista de patrones en revisión. <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	

-Concluye el requisito

Observaciones:

Prototipo de interfaz:



RF9 Guardar patrón sin terminar

Número: 9		Nombre del requisito: Guardar patrón sin terminar	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir guardar un patrón sin terminar.			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para guardar un patrón sin terminar hay que:			
- Estar autenticado en el sistema con el rol diseñador.			
- Debe existir en el sistema al menos un patrón lo mismo que en creación, modificación o corrección.			
3- Flujo de la acción a realizar:			
- El usuario selecciona en la vista de creación de patrones si desea guardar sin			

terminar de crearlo, y selecciona el botón Guardar, luego se mostrará un mensaje para informar que el patrón se ha guardado correctamente, el sistema guarda los datos introducidos por el usuario

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

The screenshot shows a web application window titled "SGCLc". Inside, there's a section titled "Patrón" with a sub-section "Generar patrón". Below this, there are three tabs: "Información", "Solución", and "Meta-Datos". The "Información" tab is active, showing several text input fields: "Nombre", "Descripción", "Contexto", "Contexto resultante", "Fundamento", and "Clasificación". There is also a button labeled "Listar Catálogos" next to the "Catálogos relacionados" label. At the bottom of the form, there are two buttons: "Guardar" and "Cancelar".

RF10 Mostrar patrón en revisión

Número: 10	Nombre del requisito: Mostrar patrón en revisión
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir mostrar los patrones en revisión en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los patrones en revisión hay que:

- Estar autenticado en el sistema con el rol X.
- El patrón debe estar en lista de revisión.

3- Flujo de la acción a realizar:

Se despliega la opción Patrón y se selecciona el listado de patrones y se muestra un listado con los patrones en revisión, y se selecciona la opción mostrar patrón.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones:

Prototipo de interfaz:



RF11 Mostrar patrón publicado

Número: 11		Nombre del requisito: Mostrar patrón en publicado	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	

Descripción:

1- Objetivo:

Permitir mostrar los patrones publicados en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los patrones publicados hay que:

- Estar autenticado en el sistema con el rol X.
- El patrón debe estar en lista de publicados.

3- Flujo de la acción a realizar:

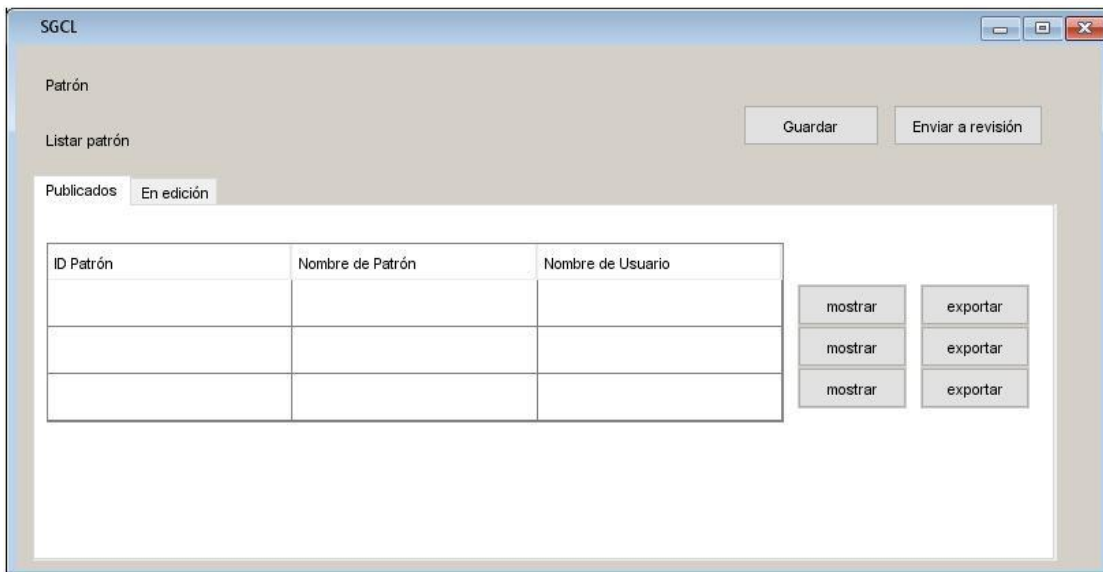
Se despliega la opción Patrón y se selecciona el listado de patrones y se muestra un listado con los patrones publicados, y se selecciona la opción mostrar patrón.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF12 Buscar patrón

Número: 12		Nombre del requisito: Buscar patrón	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			

1- Objetivo:

Permitir buscar los patrones en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para buscar los patrones en el sistema hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir al menos un patrón creado.

3- Flujo de la acción a realizar:

Se despliega la opción Patrón y se muestra un campo de texto para buscar el patrón. El usuario introduce un criterio de búsqueda y el sistema verifica la existencia del patrón. Se muestra la información del patrón.

Flujo alternativo. El patrón no existe

-Regresa a la vista previa

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones:**Prototipo de interfaz:**

RF13 Listar patrón almacenado en el sistema

Número: 13		Nombre del requisito: Listar patrón almacenado en el sistema	
Programador: Ernesto Alay Navarrete		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir listar los patrones en el sistema			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para listar los patrones en el sistema hay que:			

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos un patrón.

3- Flujo de la acción a realizar:

Cuando el usuario despliega Patrón, selecciona la opción Listar Patrón aparecen todos los patrones registrados en el sistema. Además, el usuario tiene la posibilidad de ver detalles de un patrón, editarlo y eliminarlo.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa
- Concluye el requisito

Observaciones:

Prototipo de interfaz:



RF14 Mostrar patrón en edición

Número: 14	Nombre del requisito: Mostrar patrón en edición
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir mostrar los patrones en edición en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los patrones en edición hay que:

- Estar autenticado en el sistema con el rol X.
- El patrón debe estar en lista de edición.

3- Flujo de la acción a realizar:

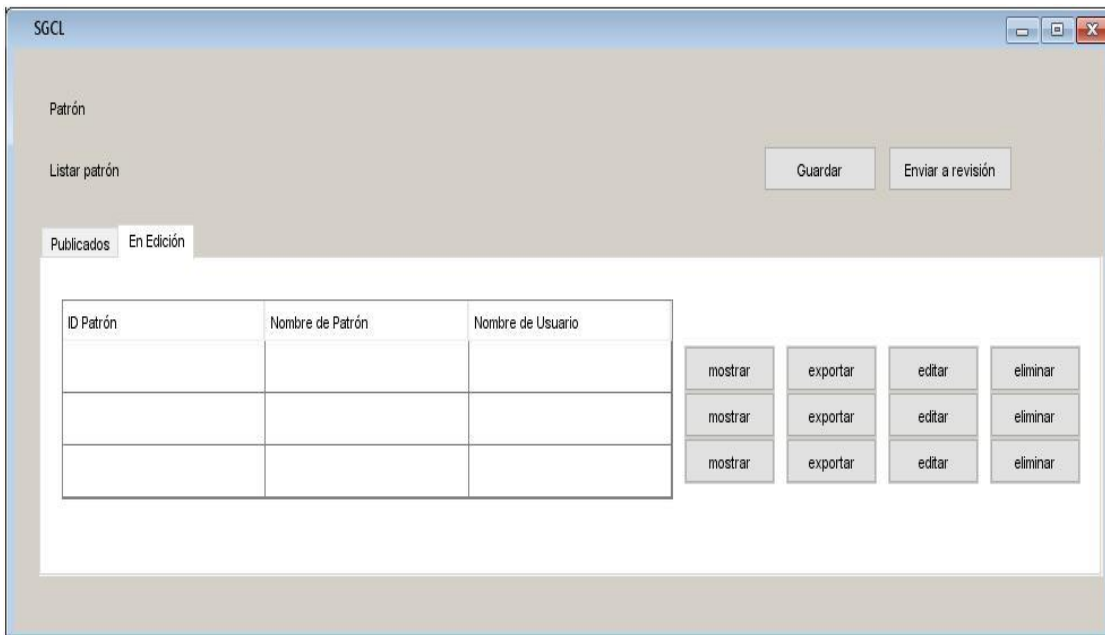
Se despliega la opción Patrón y se selecciona el listado de patrones y se muestra un listado con los patrones en edición, y se selecciona la opción mostrar patrón.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF15 Insertar comentario al patrón

Número: 15	Nombre del requisito: Insertar comentario al patrón

Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir insertar un comentario en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los patrones en edición hay que:

- Estar autenticado en el sistema con el rol Revisor.
- El patrón debe estar en el proceso de edición.

3- Flujo de la acción a realizar:

Se muestran los campos a revisar, el usuario identifica los campos erróneos y despliega la pestaña comentario. Escribe los comentarios correspondientes y selecciona la opción de guardar comentario. El sistema valida que el comentario tenga la extensión requerida.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF16 Enviar patrón a revisión una vez editado

Número: 16	Nombre del requisito: Enviar patrón a revisión una vez editado

Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar un patrón a revisión una vez editado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar un patrón a revisión una vez editado hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El patrón debe estar en procesos de modificación. <p>3- Flujo de la acción a realizar: Se despliega la opción Patrón y se selecciona en el listado de patrones, el patrón modificado que se quiere enviar a revisión. Se selecciona la opción enviar a edición. Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones	
Prototipo de interfaz:	

SGCL

Patrón

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

Clasificación

RF17 Reutilizar patrón

Número: 17	Nombre del requisito: Reutilizar patrón
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir reutilizar un patrón en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para reutilizar un patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El patrón debe estar previamente publicado. <p>3- Flujo de la acción a realizar: Se despliega la opción Patrón y se selecciona en el listado de patrones, el patrón publicado que se desea reutilizar, seleccionando la opción Reutilizar. Se muestra el patrón con todos sus campos, se modifican los campos necesarios, el sistema valida los datos y se reutiliza el patrón si los datos son correctos.</p> <p>Flujo alternativo. Datos incorrectos</p> <ul style="list-style-type: none"> -El sistema señala los datos incorrectos y permite corregirlos -Se corrigen los datos -Regresa a la vista previa <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	

SGCL

Patrón

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

Clasificación

RF18 Enviar patrón a revisión luego de rehusado

Número: 18	Nombre del requisito: Enviar patrón a revisión luego de rehusado
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días

Riesgo en Desarrollo: N/A

Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir enviar patrón a revisión luego de rehusado.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para enviar patrón a revisión luego de rehusado hay que:

- Estar autenticado en el sistema con el rol X.
- El patrón debe estar previamente publicado y rehusado.

3- Flujo de la acción a realizar:

Se despliega la opción Patrón y se selecciona en el listado de patrones, el patrón publicado que se quiere enviar a revisión. Se selecciona la opción enviar a edición.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa

Observaciones

Prototipo de interfaz:

SGCL

Patrón

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

Listar catálogo

Clasificación

Editar Cancelar

RF19 Editar patrón luego de ser rehusado

Número: 19	Nombre del requisito: Editar patrón luego de ser rehusado
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir editar patrón luego de ser rehusado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar patrón luego de ser rehusado hay que: - Estar autenticado en el sistema con el rol X. - El patrón debe estar rehusado.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar: Se despliega la opción Patrón y se selecciona en el listado de patrones, el patrón rehusado que se quiere editar. Se modifican los campos que el usuario desee y se selecciona la opción modificar. El sistema valida los datos insertados, y se actualiza.</p> <p>Flujo alternativo. Datos incorrectos -El sistema señala los datos incorrectos y permite corregirlos -Se corrigen los datos -Regresa a la vista previa</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones	
Prototipo de interfaz:	

SGCL

Patrón

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

Clasificación

RF20 Generar catálogo

Número: 20	Nombre del requisito: Generar catálogo

Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir generar un nuevo catálogo.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para generar un catálogo hay que: - Estar autenticado en el sistema.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema debe permitir incluir y/o seleccionar los datos para incluir un nuevo catálogo desplegando la opción Catálogo. - Se selecciona la opción Generar catálogo - Se muestra un formulario con los siguientes campos a rellenar <ul style="list-style-type: none"> - Nombre - Descripción del problema - Contexto - Contexto resultante - Fundamento - Catálogos relacionados - Índice de patrones - Se define la opción confirmar - El sistema valida los datos insertados - Si los datos son correctos el sistema genera el catálogo - Concluye el requisito 	

Flujo alternativo. Datos incorrectos o incompletos

-El sistema señala los datos incorrectos o incompletos y permite corregirlos.

-Se corrigen los datos.

Flujo alternativo. El usuario cancela la acción

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

El prototipo de interfaz de usuario muestra una ventana de aplicación con el título "SGCLc". Dentro de la ventana, se encuentra un formulario con el título "Catálogo" y un botón "Generar catálogo". El formulario tiene tres pestañas: "Información", "Solución" y "Meta-Datos", con "Meta-Datos" seleccionada. El formulario contiene los siguientes campos y botones:

- Campo de texto "Nombre".
- Campo de texto "Descripción".
- Campo de texto "Contexto".
- Campo de texto "Contexto resultante".
- Campo de texto "Fundamento".
- Botón "Listar Catálogos" situado debajo del campo "Fundamento".
- Botón "Confirmar" situado en la parte inferior del formulario.
- Botón "Cancelar" situado a la derecha del botón "Confirmar".

RF21 Editar catálogo

Número: 21	Nombre del requisito: Editar catálogo
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir modificar los datos de un catálogo existente</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar los datos de un catálogo hay que: - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un catálogo.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar: - El sistema debe permitir editar un catálogo, esta acción puede realizar desplegando la opción catálogo y luego seleccionar Listar catálogo, seleccionando la opción editar. - Cuando el usuario edita de forma correcta los datos necesarios y selecciona la opción Actualizar, se muestra un mensaje de información de que el catálogo fue editado de forma correcta.</p> <p>Flujo alternativo. Datos incorrectos o incompletos -El sistema señala los datos incorrectos o incompletos y permite corregirlos. -Se corrigen los datos.</p> <p>Flujo alternativo. El usuario cancela la acción</p>	

-Regresa a la vista previa

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

The image shows a web application window titled "SGCL". The main content area is titled "Catálogo" and contains a sub-section "Revisar catálogo". There are three tabs: "Información" (selected), "Solución", and "Meta-Datos". Below the tabs is a form with the following fields:

- Nombre:
- Descripción:
- Contexto:
- Contexto resultante:
- Fundamento:

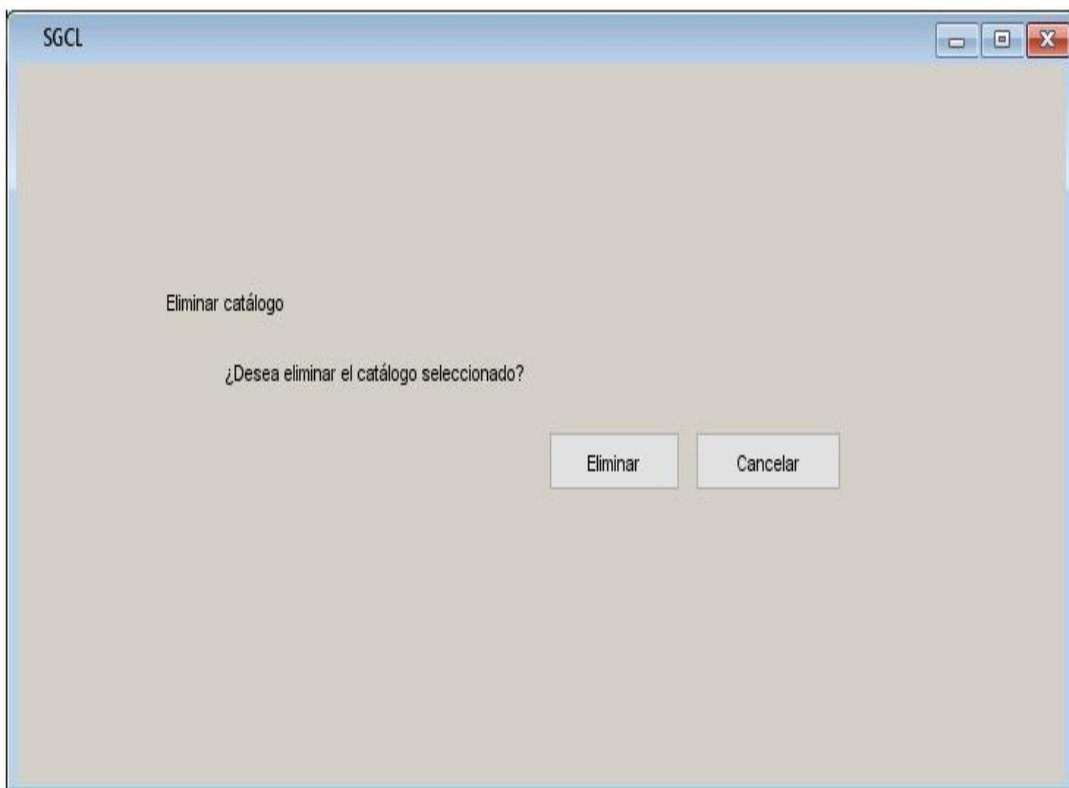
Below the form, there is a section "Catálogos relacionados" with a button "Listar catálogo". A "Cancelar" button is centered below the form.

Below a horizontal line, there is a "Comentario" section with a text area labeled "Introduzca comentario". At the bottom of the window, there are three buttons: "Guardar comentario", "Publicar", and "Corregir".

RF22 Eliminar catálogo

Número: 22	Nombre del requisito: Eliminar catálogo
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir eliminar un catálogo existente</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para eliminar un catálogo hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un catálogo. <p>3- Flujo de la acción a realizar: Se despliega la opción Catálogo y luego Listar Catálogo, donde se seleccionan los catálogos a eliminar</p> <ul style="list-style-type: none"> - El sistema permite eliminar uno o varios catálogos para eliminar varios catálogos el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción Aplicar. <p>en caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento.</p> <ul style="list-style-type: none"> - Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando. - Si selecciona la opción Aceptar se actualiza el listado y el sistema muestra un mensaje de información. <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	

Prototipo de interfaz:



RF23 Revisar catálogo en listado de revisión

Número: 23		Nombre del requisito: Revisar catálogo en listado de revisión	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir revisar los catálogos en el listado de revisión			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para revisar los catálogos en el listado de revisión hay que:			

- Estar autenticado en el sistema con el rol de Revisor.
- Debe existir en el sistema al menos un catálogo en el listado de revisión.

3- Flujo de la acción a realizar:

- Para revisar un catálogo primeramente el revisor despliega la opción Catálogo y en el Listar Catálogo selecciona el catálogo que desea revisar, verifica si los datos del catálogo están correctos.

Flujo alternativo. Datos incorrectos

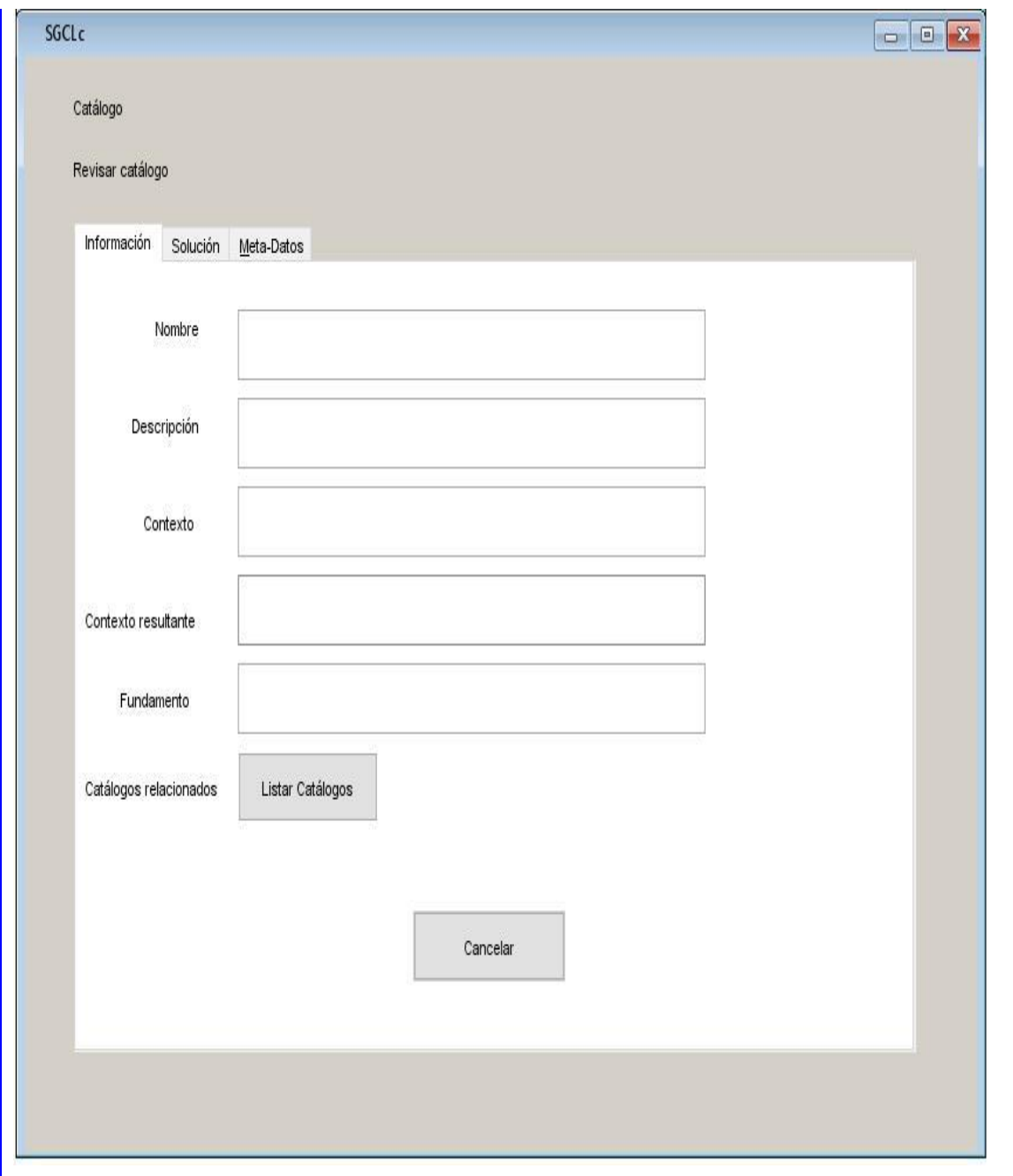
- Si el revisor detecta datos incorrectos o incompletos, le deja un comentario para que luego el diseñador de catálogos lo pueda corregir.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa
- Concluye el requisito

Observaciones:

Prototipo de interfaz:



RF24 Publicar catálogo

Número: 24	Nombre del requisito: Publicar catálogo
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir publicar un catálogo</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para publicar un catálogo hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol revisor. - El catálogo debe estar revisado. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El revisor despliega en la opción Catálogo, Listar Catálogo, y selecciona el catálogo que desea publicar, que haya sido revisado anteriormente. Para luego publicarlo mediante la opción Publicar, cuando seleccione el botón aceptar, se mostrará un mensaje para informar que el catálogo se ha publicado correctamente <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

SGCL

Catálogo

Revisar catálogo

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

Comentario

Introduzca comentario

RF25 Enviar catálogo a revisión

Número: 25	Nombre del requisito: Enviar catálogo a revisión
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar catálogo a revisión</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar catálogo a revisión hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un catálogo. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El usuario selecciona el catálogo que desea enviar a revisión desplegando Catálogo, Listar Catálogo respectivamente. El catálogo es enviado a lista de catálogos en revisión. <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

SGCLc

Catálogo

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

RF26 Guardar catálogo sin terminar

Número: 26	Nombre del requisito: Guardar catálogo sin terminar

Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir guardar un catálogo sin terminar.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para guardar un catálogo sin terminar hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un catálogo lo mismo que en creación, modificación o corrección. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El usuario selecciona en la vista de creación de catálogos si desea guardar sin terminar de crearlo, y selecciona el botón Guardar, luego se mostrará un mensaje para informar que el catálogo se ha guardado correctamente, el sistema guarda los datos introducidos por el usuario <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

SGCLc

Catálogo

Generar catálogo

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados Listar Catálogos

Guardar Cancelar

RF27 Mostrar catálogo en revisión

Número: 27	Nombre del requisito: Mostrar catálogo en revisión
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir mostrar los catálogos en revisión en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los catálogos en revisión hay que:

- Estar autenticado en el sistema con el rol X.
- El catálogo debe estar en lista de revisión.

3- Flujo de la acción a realizar:

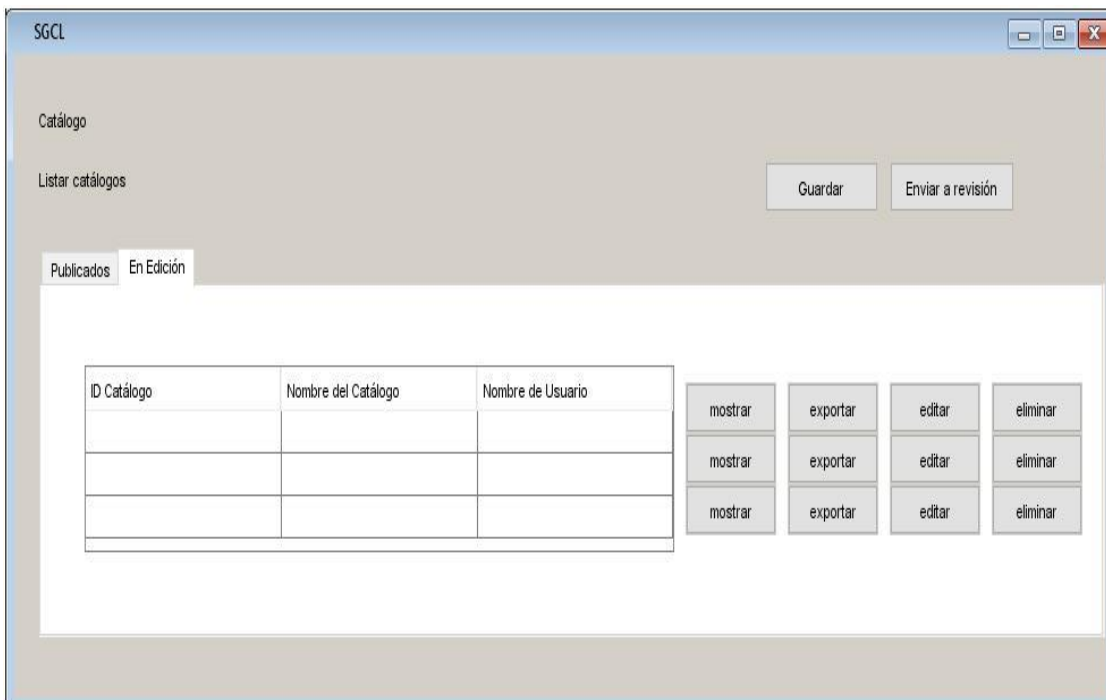
Se despliega la opción Catálogo y se selecciona el listado de catálogos y se muestra un listado con los catálogos en revisión, y se selecciona la opción mostrar catálogo.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

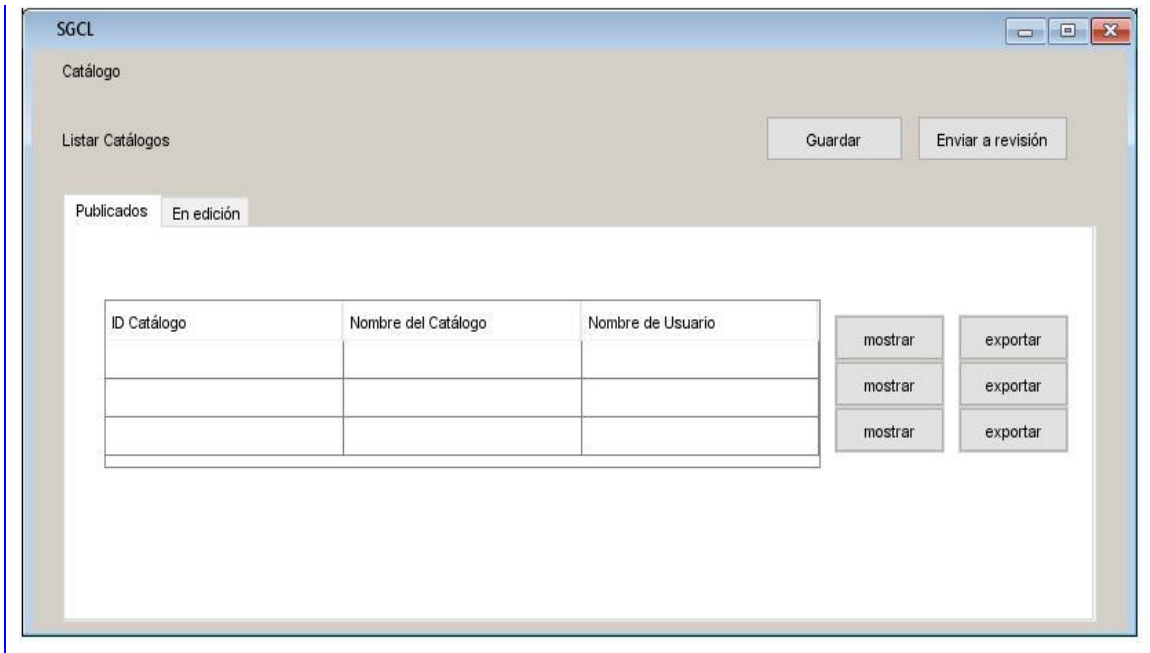
Observaciones:

Prototipo de interfaz:



RF28 Mostrar catálogo publicado

Número: 28	Nombre del requisito: Mostrar catálogo en publicado
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir mostrar los catálogos publicados en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para mostrar los catálogos publicados hay que: - Estar autenticado en el sistema con el rol X. - El catálogo debe estar en lista de publicados.</p> <p>3- Flujo de la acción a realizar: Se despliega la opción Catálogo y se selecciona el listado de catálogos y se muestra un listado con los catálogos publicados, y se selecciona la opción mostrar catálogo.</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones	
Prototipo de interfaz:	



RF29 Buscar catálogo

Número: 29	Nombre del requisito: Buscar catálogo
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir buscar los catálogos en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para buscar los catálogos en el sistema hay que: - Estar autenticado en el sistema con el rol X. - Debe existir al menos un catálogo creado.</p> <p>3- Flujo de la acción a realizar: Se despliega la opción Catálogo y se muestra un campo de texto para buscar el catálogo. El usuario introduce un criterio de búsqueda y el sistema verifica la existencia del catálogo. Se muestra la información del catálogo.</p>	

Flujo alternativo. El catálogo no existe

-Regresa a la vista previa

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones:

Prototipo de interfaz:



RF30 Listar catálogo almacenado en el sistema

Número: 30		Nombre del requisito: Listar catálogo almacenado en el sistema	
Programador: Ernesto Alay Navarrete		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir listar los catálogos en el sistema			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para listar los catálogos en el sistema hay que:			
- Estar autenticado en el sistema con el rol X.			
- Debe existir en el sistema al menos un catálogo.			
3- Flujo de la acción a realizar:			
Cuando el usuario despliega Catálogo, selecciona la opción Listar Catálogo aparecen			

todos los catálogos registrados en el sistema. Además, el usuario tiene la posibilidad de ver detalles de un catálogo, editarlo y eliminarlo.

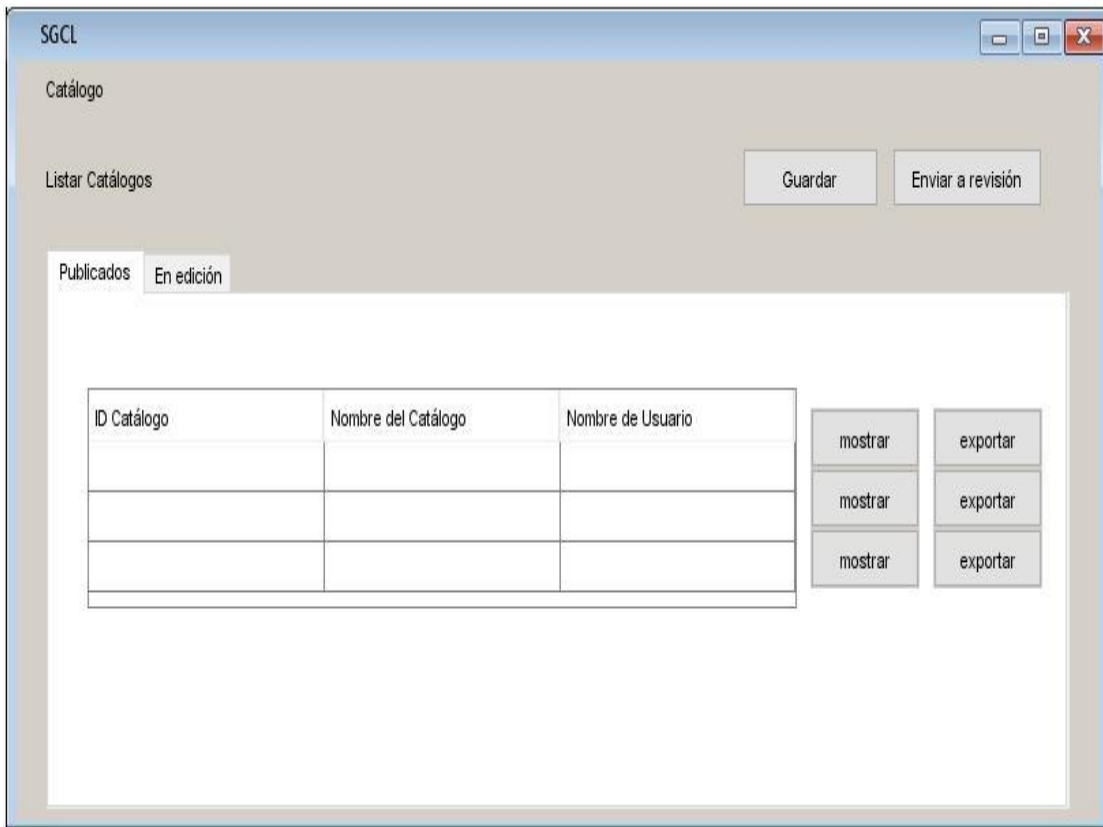
Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

-Concluye el requisito

Observaciones:

Prototipo de interfaz:



RF31 Mostrar catálogo en edición

Número: 31		Nombre del requisito: Mostrar catálogo en edición	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	

Riesgo en Desarrollo: N/A

Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir mostrar los catálogos en edición en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los catálogos en edición hay que:

- Estar autenticado en el sistema con el rol X.
- El catálogo debe estar en lista de edición.

3- Flujo de la acción a realizar:

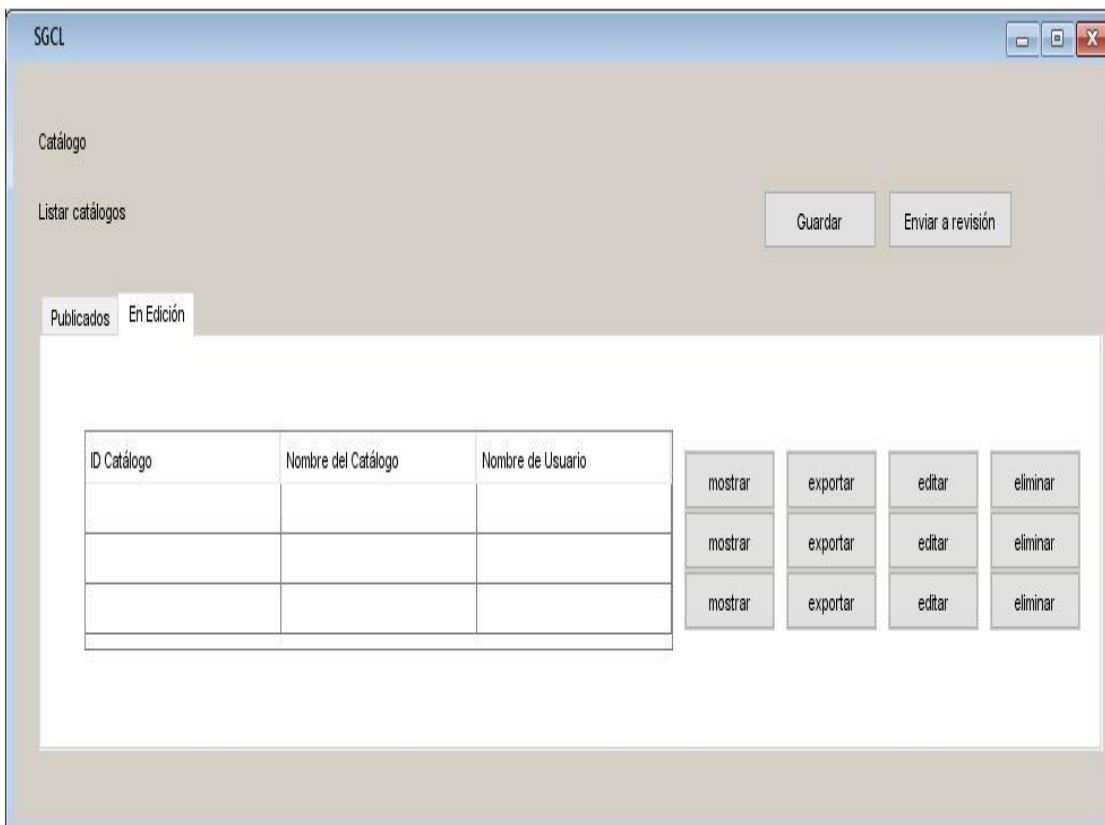
Se despliega la opción Catálogo y se selecciona el listado de catálogos y se muestra un listado con los catálogos en edición, y se selecciona la opción mostrar catálogo.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF32 Insertar comentario

Número: 32	Nombre del requisito: Insertar comentario
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir insertar un comentario en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para mostrar los catálogos en edición hay que: - Estar autenticado en el sistema con el rol Revisor. - El catálogo debe estar en el proceso de edición.</p> <p>3- Flujo de la acción a realizar: Se muestran los campos a revisar, el usuario identifica los campos erróneos y despliega la pestaña comentario. Escribe los comentarios correspondientes y selecciona la opción de guardar comentario. El sistema valida que el comentario tenga la extensión requerida.</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones	
Prototipo de interfaz:	

Comentario

Insertar comentario

Guardar comentario Publicar Corregir

RF33 Enviar catálogo a revisión una vez editado

Número: 33	Nombre del requisito: Enviar catálogo a revisión una vez editado
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar un catálogo a revisión una vez editado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar un catálogo a revisión una vez editado hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El catálogo debe estar en procesos de modificación. <p>3- Flujo de la acción a realizar: Se despliega la opción Catálogo y se selecciona en el listado de catálogos, el catálogo modificado que se quiere enviar a revisión. Se selecciona la opción enviar a edición.</p> <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	

SGCLc

Catálogo

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

RF34 Reutilizar catálogo

Número: 34	Nombre del requisito: Reutilizar catálogo
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir reutilizar un catálogo en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para reutilizar un catálogo hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El catálogo debe estar previamente publicado. <p>3- Flujo de la acción a realizar: Se despliega la opción Catálogo y se selecciona en el listado de catálogos, el catálogo publicado que se desea reutilizar, seleccionando la opción Reutilizar. Se muestra el catálogo con todos sus campos, se modifican los campos necesarios, el sistema valida los datos y se reutiliza el catálogo si los datos son correctos.</p> <p>Flujo alternativo. Datos incorrectos</p> <ul style="list-style-type: none"> -El sistema señala los datos incorrectos y permite corregirlos -Se corrigen los datos -Regresa a la vista previa <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	

SGCLc

Catálogo

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

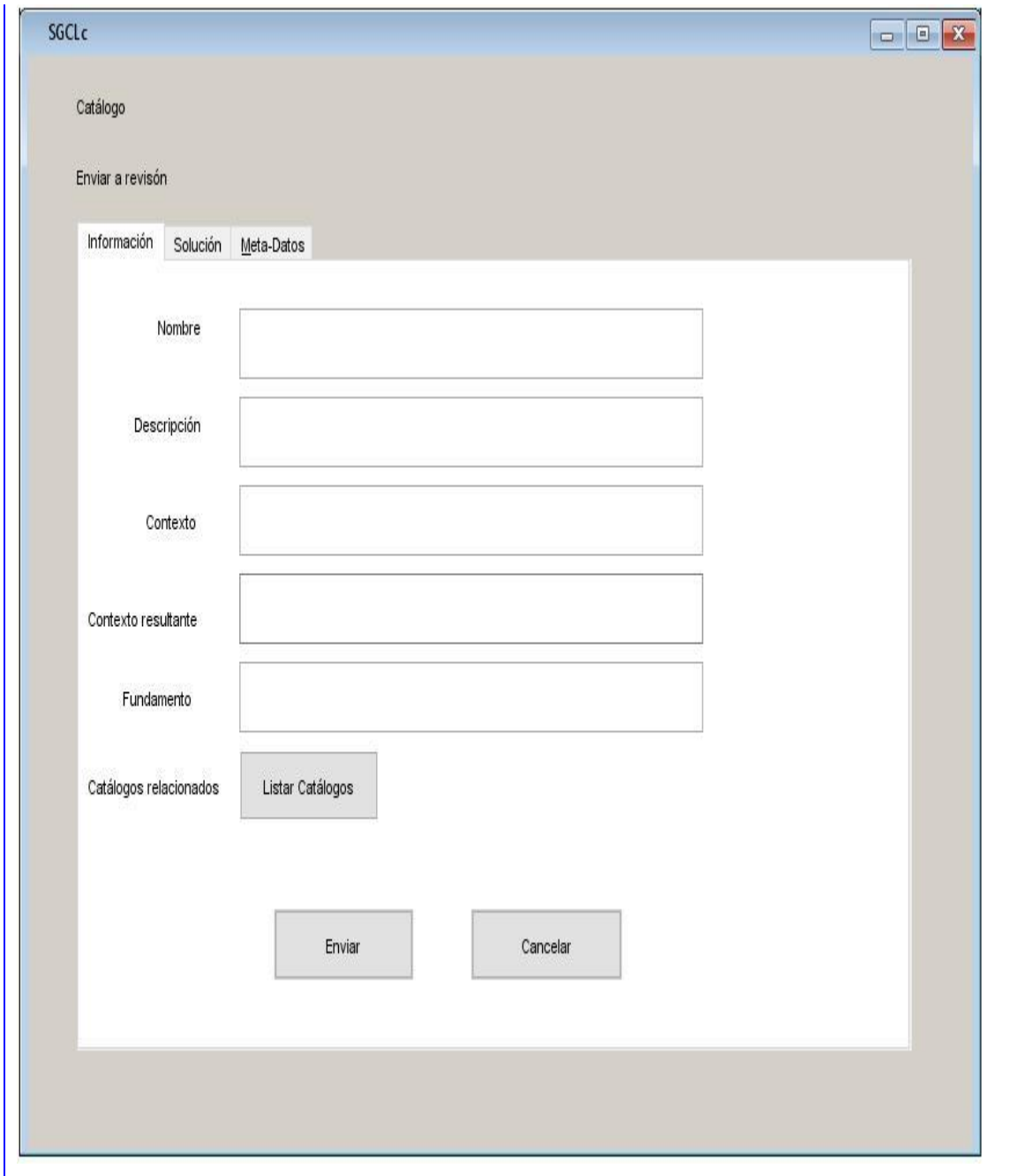
Catálogos relacionados Listar Catálogos

Editar Cancelar

RF35 Enviar catálogo a revisión luego de rehusado

Número: 35	Nombre del requisito: Enviar catálogo a revisión luego de rehusado
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar catálogo a revisión luego de rehusado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar catálogo a revisión luego de rehusado hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El catálogo debe estar previamente publicado y rehusado. <p>3- Flujo de la acción a realizar: Se despliega la opción Catálogo y se selecciona en el listado de catálogos, el catálogo publicado que se quiere enviar a revisión. Se selecciona la opción enviar a edición.</p> <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	



RF36 Editar catálogo luego de ser rehusado

Número: 36	Nombre del requisito: Editar catálogo luego de ser rehusado
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir editar catálogo luego de ser rehusado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar catálogo luego de ser rehusado hay que: - Estar autenticado en el sistema con el rol X. - El catálogo debe estar rehusado.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar: Se despliega la opción Catálogo y se selecciona en el listado de catálogos, el catálogo rehusado que se quiere editar. Se modifican los campos que el usuario desee y se selecciona la opción modificar. El sistema valida los datos insertados, y se actualiza.</p> <p>Flujo alternativo. Datos incorrectos -El sistema señala los datos incorrectos y permite corregirlos -Se corrigen los datos -Regresa a la vista previa</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones	
Prototipo de interfaz:	

SGCLc

Catálogo

Enviar a revisión

Información Solución Meta-Datos

Nombre

Descripción

Contexto

Contexto resultante

Fundamento

Catálogos relacionados

RF37 Generar lenguaje



Número: 37	Nombre del requisito: Generar lenguaje
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir generar un nuevo lenguaje.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para generar un lenguaje hay que: - Estar autenticado en el sistema.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema debe permitir incluir y/o seleccionar los datos para incluir un nuevo lenguaje desplegando la opción Lenguaje. - Se selecciona la opción Generar lenguaje - Se muestra un formulario con los siguientes campos a rellenar - Nombre - Tipo de relación - Descripción del problema - Contexto - Contexto resultante - Fundamento - Lenguajes relacionados - Clasificación 	

- Relaciones entre patrones
- Se define la opción confirmar
- El sistema valida los datos insertados
- Si los datos son correctos el sistema genera el lenguaje
- Concluye el requisito

Flujo alternativo. Datos incorrectos o incompletos

- El sistema señala los datos incorrectos o incompletos y permite corregirlos.
- Se corrigen los datos.

Flujo alternativo. El usuario cancela la acción

- Concluye el requisito

Observaciones:

Prototipo de interfaz:

SGCLc

Lenguaje

Generar lenguaje

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

RF38 Editar lenguaje

Número: 38	Nombre del requisito: Editar lenguaje
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir modificar los datos de un lenguaje existente</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar los datos de un lenguaje hay que: - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un lenguaje.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar: - El sistema debe permitir editar un lenguaje, esta acción puede realizarse seleccionando la opción editar desplegando la opción Lenguaje y luego seleccionar Listar Lenguaje. - Cuando el usuario edita de forma correcta los datos necesarios y selecciona la opción Actualizar, se muestra un mensaje de información de que el lenguaje fue editado de forma correcta.</p> <p>Flujo alternativo. Datos incorrectos o incompletos</p>	

- El sistema señala los datos incorrectos o incompletos y permite corregirlos.
- Se corrigen los datos.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa
- Concluye el requisito

Observaciones:

Prototipo de interfaz:

SGCLc

Lenguaje

Revisar lenguaje

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

Comentario

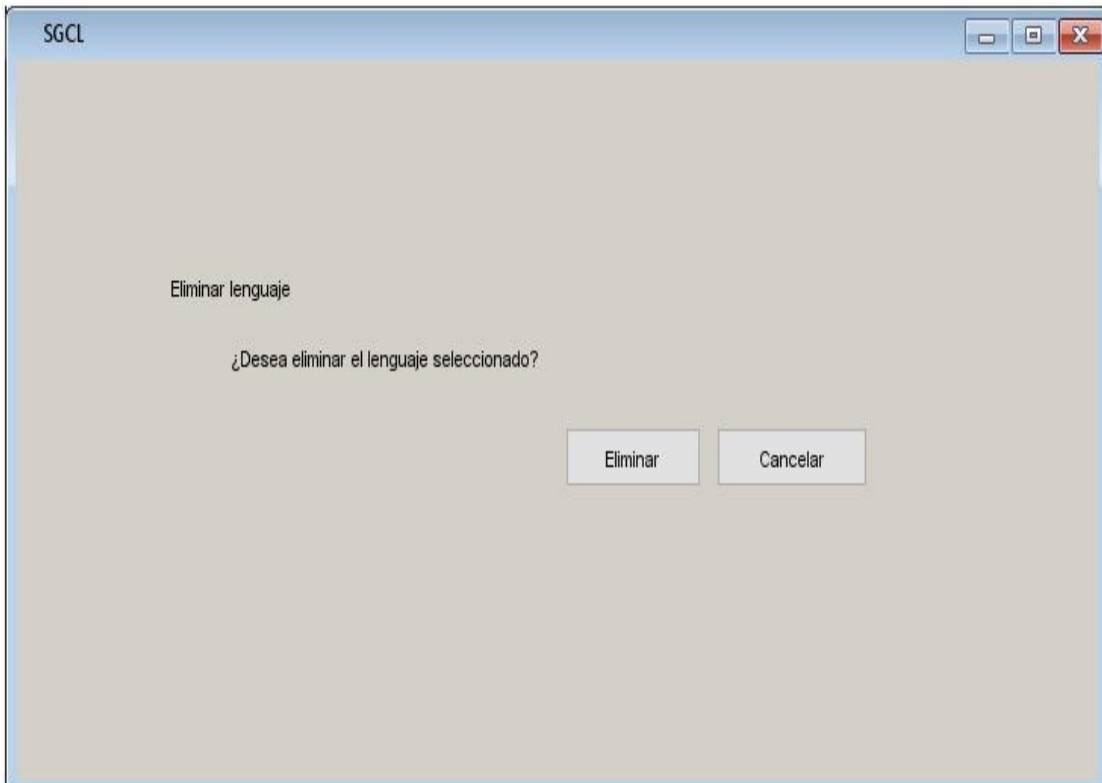
RF39 Eliminar lenguaje

Número: 39	Nombre del requisito: Eliminar lenguaje
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir eliminar un lenguaje existente</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para eliminar un lenguaje hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un lenguaje. <p>3- Flujo de la acción a realizar: Se despliega la opción Lenguaje y luego Listar Lenguaje, donde se seleccionan los lenguajes a eliminar</p> <ul style="list-style-type: none"> - El sistema permite eliminar uno o varios lenguajes para eliminar varios lenguajes el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción Aplicar. <p>en caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento.</p> <ul style="list-style-type: none"> - Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando. - Si selecciona la opción Aceptar se actualiza el listado y el sistema muestra un mensaje de información. <p>Flujo alternativo. El usuario cancela la acción</p>	

- Regresa a la vista previa
- Concluye el requisito

Observaciones:

Prototipo de interfaz:



RF40 Revisar lenguaje en listado de revisión

Número: 40	Nombre del requisito: Revisar lenguaje en listado de revisión
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
Descripción:	

1- Objetivo:

Permitir revisar los lenguajes en el listado de revisión

2- Acciones para lograr el objetivo (precondiciones y datos):

Para revisar los lenguajes en el listado de revisión hay que:

- Estar autenticado en el sistema con el rol de Revisor.
- Debe existir en el sistema al menos un lenguaje en el listado de revisión.

3- Flujo de la acción a realizar:

- Para revisar un lenguaje primeramente el revisor despliega la opción Lenguaje y en el Listar Lenguaje selecciona el lenguaje que desea revisar, verifica si los datos del lenguaje están correctos.

Flujo alternativo. Datos incorrectos

-Si el revisor detecta datos incorrectos o incompletos, le deja un comentario para que luego el diseñador de lenguajes lo pueda corregir.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa
- Concluye el requisito

Observaciones:**Prototipo de interfaz:**

SGCLc

Lenguaje

Revisar lenguaje

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

RF41 Publicar lenguaje

Número: 41	Nombre del requisito: Publicar lenguaje
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir publicar un lenguaje</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para publicar un lenguaje hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol revisor. - El lenguaje debe estar revisado. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El revisor despliega en la opción Lenguaje, Listar Lenguaje, y selecciona el lenguaje que desea publicar, que haya sido revisado anteriormente. Para luego publicarlo mediante la opción Publicar, cuando seleccione el botón aceptar, se mostrará un mensaje para informar que el lenguaje se ha publicado correctamente <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

Lenguaje

Revisar lenguaje

Información Solución Meta-Datos

Nombre

Tipo de relación

 UML Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Listar lenguajes

Clasificación

Cancelar

Comentario

Ingresar comentario

Guardar Comentario

Publicar

Corregir

RF42 Enviar lenguaje a revisión

Número: 42	Nombre del requisito: Enviar lenguaje a revisión
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar lenguaje a revisión</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar lenguaje a revisión hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un lenguaje. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El usuario selecciona el lenguaje que desea enviar a revisión desplegando Lenguaje, Listar Lenguaje respectivamente. El lenguaje es enviado a lista de lenguajes en revisión. <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

SGCLc

Lenguaje

Generar lenguaje

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

RF43 Guardar lenguaje sin terminar

Número: 43	Nombre del requisito: Guardar lenguaje sin terminar
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir guardar un lenguaje sin terminar.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para guardar un lenguaje sin terminar hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un lenguaje lo mismo que en creación, modificación o corrección. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El usuario selecciona en la vista de creación de lenguajes si desea guardar sin terminar de crearlo, y selecciona el botón Guardar, luego se mostrará un mensaje para informar que el lenguaje se ha guardado correctamente, el sistema guarda los datos introducidos por el usuario <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa -Concluye el requisito 	
Observaciones:	
Prototipo de interfaz:	

SGCLc

Lenguaje

Generar lenguaje

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

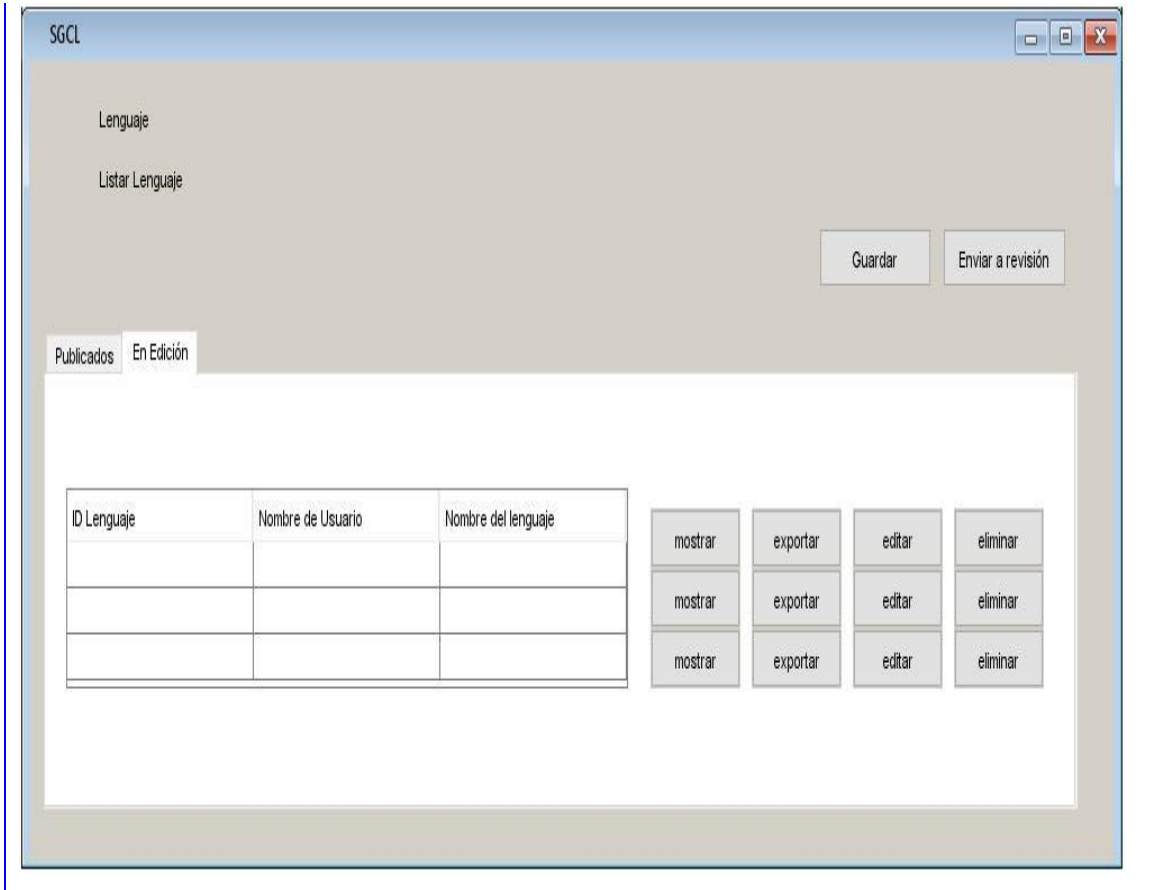
Fundamento

Lenguajes relacionados

Clasificación

RF44 Mostrar lenguaje en revisión

Número: 44	Nombre del requisito: Mostrar lenguaje en revisión
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir mostrar los lenguajes en revisión en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para mostrar los lenguajes en revisión hay que: - Estar autenticado en el sistema con el rol X. - El lenguaje debe estar en lista de revisión.</p> <p>3- Flujo de la acción a realizar: Se despliega la opción Lenguaje y se selecciona el listado de lenguajes y se muestra un listado con los lenguajes en revisión, y se selecciona la opción mostrar lenguaje.</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones:	
Prototipo de interfaz:	



RF45 Mostrar lenguaje publicado

Número: 45	Nombre del requisito: Mostrar lenguaje en publicado
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
Descripción:	
1- Objetivo:	
Permitir mostrar los lenguajes publicados en el sistema.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para mostrar los lenguajes publicados hay que:	

- Estar autenticado en el sistema con el rol X.
- El lenguaje debe estar en lista de publicados.

3- Flujo de la acción a realizar:

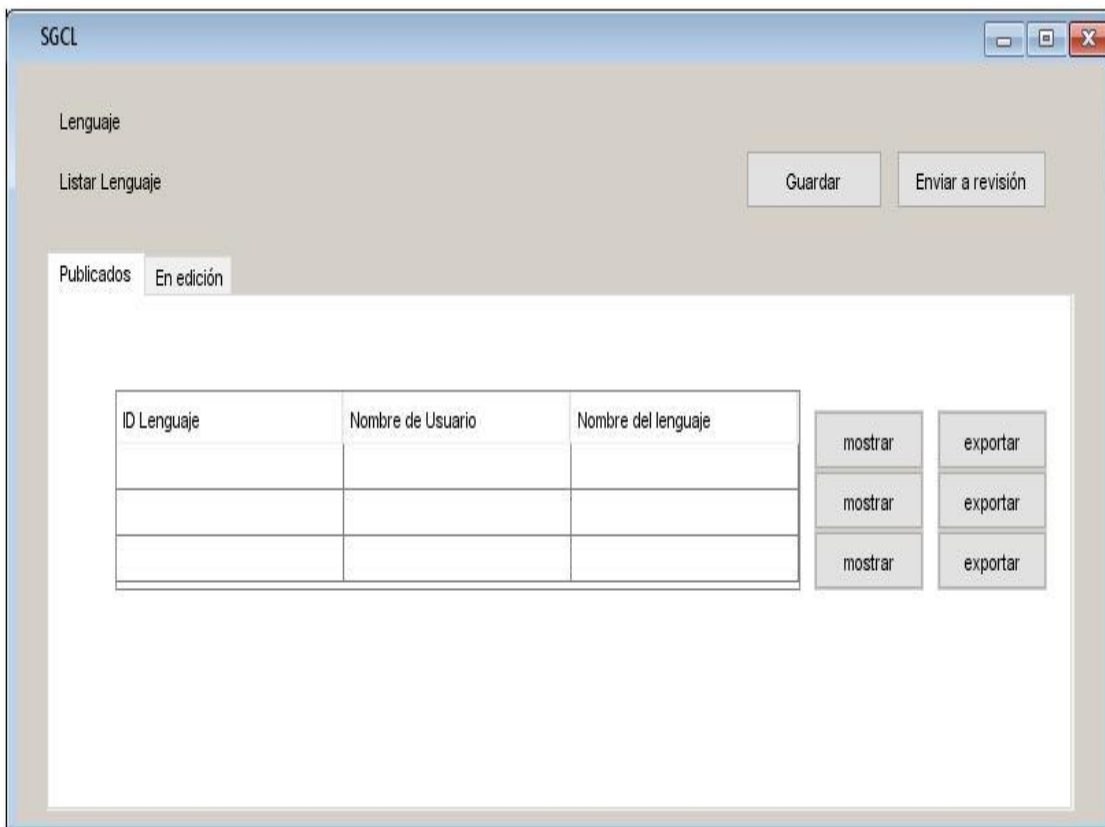
Se despliega la opción Lenguaje y se selecciona el listado de lenguajes y se muestra un listado con los lenguajes publicados, y se selecciona la opción mostrar lenguaje.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF46 Buscar lenguaje

Número: 46	Nombre del requisito: Buscar lenguaje

Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir buscar los lenguajes en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para buscar los lenguajes en el sistema hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir al menos un lenguaje creado.

3- Flujo de la acción a realizar:

Se despliega la opción Lenguaje y se muestra un campo de texto para buscar el lenguaje. El usuario introduce un criterio de búsqueda y el sistema verifica la existencia del lenguaje. Se muestra la información del lenguaje.

Flujo alternativo. El lenguaje no existe

- Regresa a la vista previa

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa

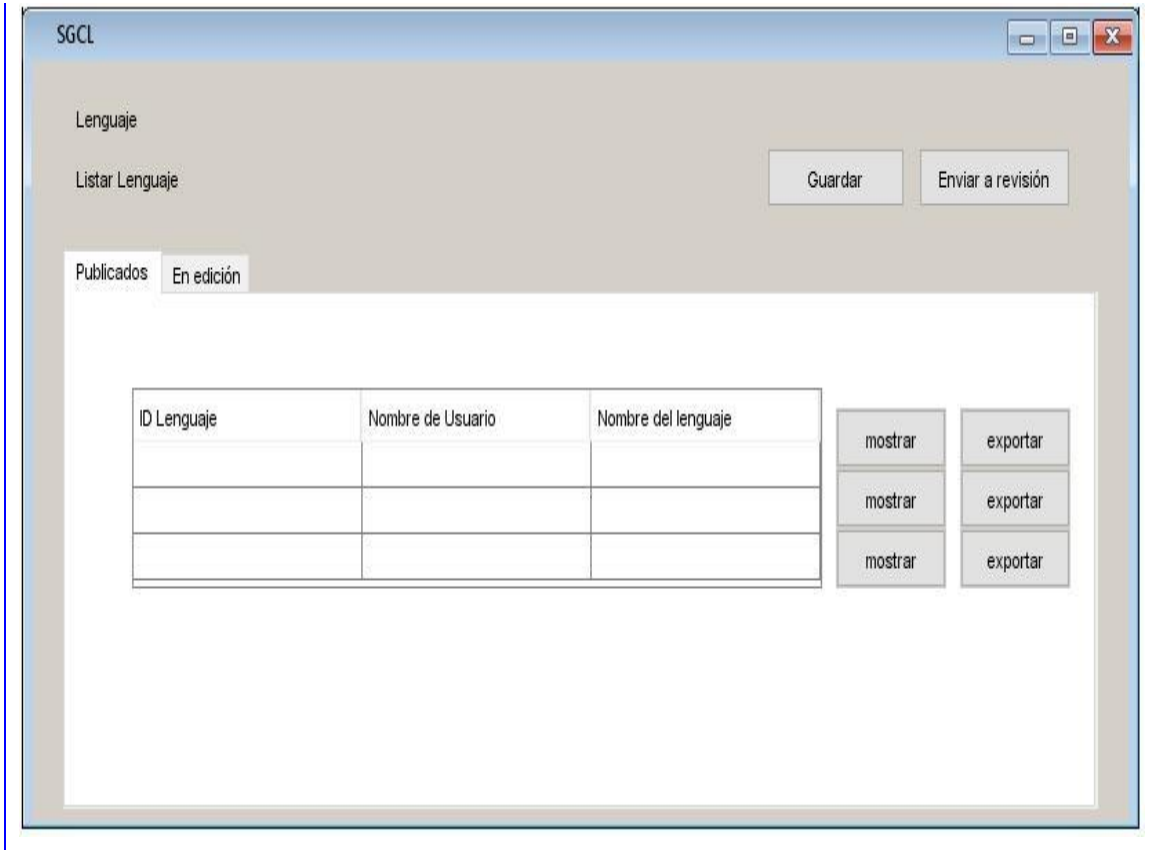
Observaciones:

Prototipo de interfaz:



RF47 Listar lenguaje almacenado en el sistema

Número: 47	Nombre del requisito: Listar lenguaje almacenado en el sistema
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir listar los lenguajes en el sistema</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para listar los lenguajes en el sistema hay que: - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un lenguaje.</p> <p>3- Flujo de la acción a realizar: Cuando el usuario despliega Lenguaje, selecciona la opción Listar Lenguaje aparecen todos los lenguajes registrados en el sistema. Además, el usuario tiene la posibilidad de ver detalles de un lenguaje, editarlo y eliminarlo.</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa -Concluye el requisito</p>	
Observaciones:	
Prototipo de interfaz:	



RF48 Mostrar lenguaje en edición

Número: 48	Nombre del requisito: Mostrar lenguaje en edición
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir mostrar los lenguajes en edición en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para mostrar los lenguajes en edición hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El lenguaje debe estar en lista de edición. 	

3- Flujo de la acción a realizar:

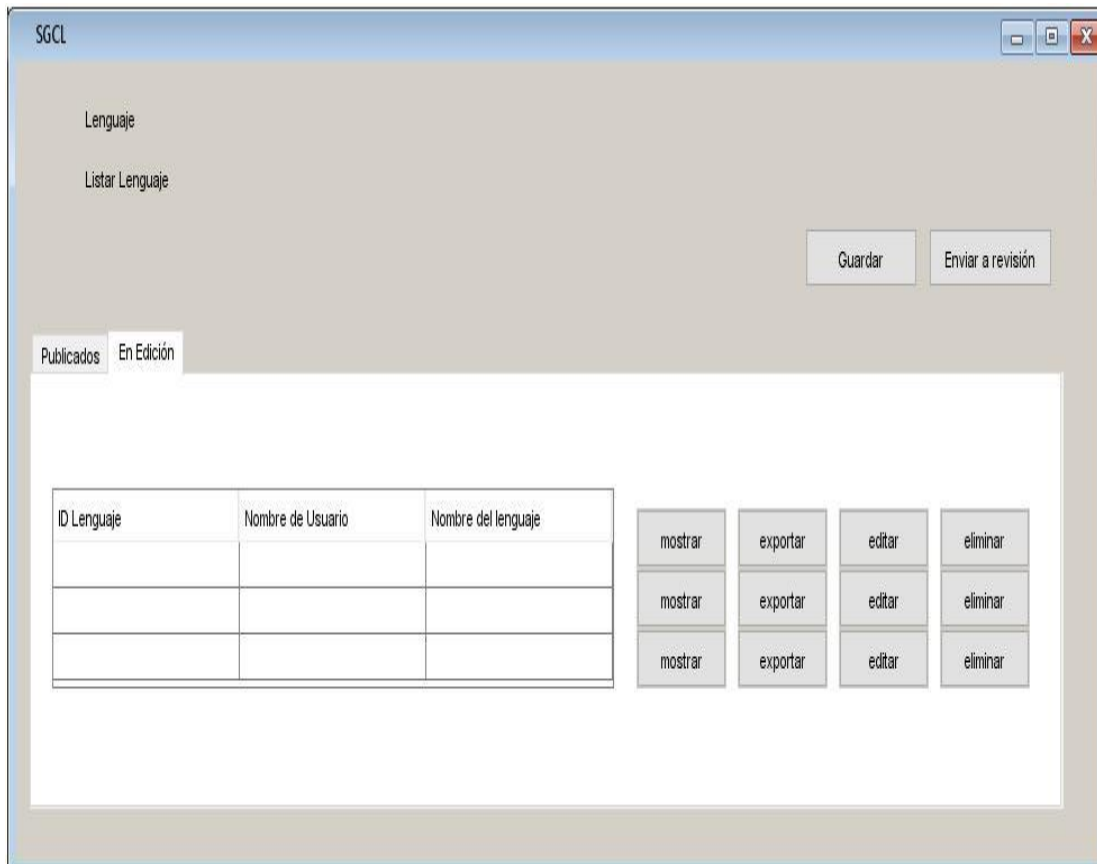
Se despliega la opción Lenguaje y se selecciona el listado de lenguajes y se muestra un listado con los lenguajes en edición, y se selecciona la opción mostrar lenguaje.

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF49 Insertar comentario

Número: 49		Nombre del requisito: Insertar comentario	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	

Riesgo en Desarrollo: N/A

Tiempo Real: N/A

Descripción:

1- Objetivo:

Permitir insertar un comentario en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar los lenguajes en edición hay que:

- Estar autenticado en el sistema con el rol Revisor.
- El lenguaje debe estar en el proceso de edición.

3- Flujo de la acción a realizar:

Se muestran los campos a revisar, el usuario identifica los campos erróneos y despliega la pestaña comentario. Escribe los comentarios correspondientes y selecciona la opción de guardar comentario. El sistema valida que el comentario tenga la extensión requerida.

Flujo alternativo. El usuario cancela la acción

- Regresa a la vista previa

Observaciones

Prototipo de interfaz:

Comentario

Insertar comentario

Guardar comentario

Publicar

Corregir

RF50 Enviar lenguaje a revisión una vez editado

Número: 50

Nombre del requisito: Enviar lenguaje a revisión una vez

editado	
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar un lenguaje a revisión una vez editado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar un lenguaje a revisión una vez editado hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El lenguaje debe estar en procesos de modificación. <p>3- Flujo de la acción a realizar: Se despliega la opción Lenguaje y se selecciona en el listado de lenguajes, el lenguaje modificado que se quiere enviar a revisión. Se selecciona la opción enviar a edición.</p> <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	

SGCLc

Lenguaje

Enviar a revisión

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

RF51 Reutilizar lenguaje

Número: 51	Nombre del requisito: Reutilizar lenguaje
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir reutilizar un lenguaje en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para reutilizar un lenguaje hay que: <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El lenguaje debe estar previamente publicado. </p> <p>3- Flujo de la acción a realizar: Se despliega la opción Lenguaje y se selecciona en el listado de lenguajes, el lenguaje publicado que se desea reutilizar, seleccionando la opción Reutilizar. Se muestra el lenguaje con todos sus campos, se modifican los campos necesarios, el sistema valida los datos y se reutiliza el lenguaje si los datos son correctos.</p> <p>Flujo alternativo. Datos incorrectos <ul style="list-style-type: none"> -El sistema señala los datos incorrectos y permite corregirlos -Se corrigen los datos -Regresa a la vista previa </p> <p>Flujo alternativo. El usuario cancela la acción <ul style="list-style-type: none"> -Regresa a la vista previa </p>	
Observaciones	
Prototipo de interfaz:	

SGCLc

Lenguaje

Enviar a revisión

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

RF52 Enviar lenguaje a revisión luego de rehusado

Número: 52	Nombre del requisito: Enviar lenguaje a revisión luego de rehusado
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir enviar lenguaje a revisión luego de rehusado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para enviar lenguaje a revisión luego de rehusado hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - El lenguaje debe estar previamente publicado y rehusado. <p>3- Flujo de la acción a realizar: Se despliega la opción Lenguaje y se selecciona en el listado de lenguajes, el lenguaje publicado que se quiere enviar a revisión. Se selecciona la opción enviar a edición.</p> <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	

SGCLc

Lenguaje

Enviar a revisión

Información Solución Meta-Datos

Nombre

Tipo de relación UML
 Semántica

Descripción

Contexto

Contexto resultante

Fundamento

Lenguajes relacionados

Clasificación

RF53 Editar lenguaje luego de ser rehusado

Número: 53	Nombre del requisito: Editar lenguaje luego de ser rehusado
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir editar lenguaje luego de ser rehusado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar lenguaje luego de ser rehusado hay que: - Estar autenticado en el sistema con el rol X. - El lenguaje debe estar rehusado.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar: Se despliega la opción Lenguaje y se selecciona en el listado de lenguajes, el lenguaje rehusado que se quiere editar. Se modifican los campos que el usuario desee y se selecciona la opción modificar. El sistema valida los datos insertados, y se actualiza.</p> <p>Flujo alternativo. Datos incorrectos -El sistema señala los datos incorrectos y permite corregirlos -Se corrigen los datos -Regresa a la vista previa</p> <p>Flujo alternativo. El usuario cancela la acción -Regresa a la vista previa</p>	
Observaciones	

Prototipo de interfaz:

The screenshot shows a window titled "SGCLc" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area has a light gray background. At the top left, the text "Lenguaje" is displayed. Below it is the text "Enviar a revisión". A tabbed interface is present with three tabs: "Información" (selected), "Solución", and "Meta-Datos". The "Información" tab contains the following fields and controls:

- Nombre:** A text input field.
- Tipo de relación:** Two radio buttons labeled "UML" and "Semántica".
- Descripción:** A text input field.
- Contexto:** A text input field.
- Contexto resultante:** A text input field.
- Fundamento:** A text input field.
- Lenguajes relacionados:** A button labeled "Listar lenguajes".
- Clasificación:** A text input field.

At the bottom of the form area, there are two buttons: "Editar" and "Cancelar".

RF54 Añadir metadatos al patrón

Número: 54

Nombre del requisito: Añadir metadatos al patrón

Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir añadir metadatos al patrón.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para añadir metadatos al patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema. - Al menos un patrón creado. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Se despliega la opción Patrón y luego se selecciona el botón Meta-datos - Se muestra un formulario con los siguientes campos a rellenar <ul style="list-style-type: none"> - Estado - Rol contribución - Versión - Densidad semántica - Dificultad - Rol contribución - Formato - Costo - Localización - Se define la opción generar - El sistema valida los datos insertados - Si los datos son correctos el sistema crea el metadato - Concluye el requisito <p>Flujo alternativo. Datos incorrectos</p> <ul style="list-style-type: none"> -El sistema señala los datos incorrectos y permite corregirlos. -Se corrigen los datos. 	

Flujo alternativo. El usuario cancela la acción

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

The screenshot shows a web application window titled "SGCLc" with a "Patrón" section. Below the title is a "Generar patrón" button. The main content area has three tabs: "Información", "Solución", and "Meta-Datos", with "Meta-Datos" selected. The form contains the following fields and controls:

- Estado:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Rol Contribución:** A text input field containing the value "rol del ciclo de vida".
- Versión:** A text input field containing the value "1,2,3...".
- Densidad semántica:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Dificultad:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Rol Contribución:** A text input field containing the value "rol de Meta-Dato".
- Formato:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Costo:** Two radio buttons labeled "No" and "Si".
- Localización:** A text input field containing the value "url".

At the bottom of the form are two buttons: "Confirmar" and "Cancelar".

RF55 Añadir metadatos al catálogo

Número: 55	Nombre del requisito: Añadir metadatos al catálogo
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir añadir metadatos al catálogo.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para añadir metadatos al patrón hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema. - Al menos un catálogo creado. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Se despliega la opción catálogo y luego se selecciona el botón Meta-datos - Se muestra un formulario con los siguientes campos a rellenar <ul style="list-style-type: none"> - Estado - Rol contribución - Versión - Densidad semántica - Dificultad - Rol contribución - Formato - Costo - Localización - Se define la opción generar - El sistema valida los datos insertados - Si los datos son correctos el sistema crea el metadato - Concluye el requisito 	

Flujo alternativo. Datos incorrectos

-El sistema señala los datos incorrectos y permite corregirlos.

-Se corrigen los datos.

Flujo alternativo. El usuario cancela la acción

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

The screenshot shows a web application window titled "SGCLc". Inside, there is a section labeled "Catálogo" with a sub-section "Generar catálogo". Below this, there are three tabs: "Información", "Solución", and "Meta-Datos", with "Meta-Datos" being the active tab. The form contains the following fields and controls:

- Estado:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Rol Contribución:** A text input field containing the text "rol del ciclo de vida".
- Versión:** A text input field containing the text "1,2,3...".
- Densidad semántica:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Dificultad:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Rol Contribución:** A text input field containing the text "rol de Meta-Dato".
- Formato:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Costo:** Two radio buttons labeled "No" and "Si".
- Localización:** A text input field containing the text "url".

At the bottom of the form, there are two buttons: "Confirmar" and "Cancelar".

RF56 Añadir metadatos al lenguaje

Número: 56

Nombre del requisito: Añadir metadatos al lenguaje

Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir añadir metadatos al lenguaje.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para añadir metadatos al lenguaje hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema. - Al menos un lenguaje creado. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Se despliega la opción Lenguaje y luego se selecciona el botón Meta-datos - Se muestra un formulario con los siguientes campos a rellenar <ul style="list-style-type: none"> - Estado - Rol contribución - Versión - Densidad semántica - Dificultad - Rol contribución - Formato - Costo - Localización - Se define la opción generar - El sistema valida los datos insertados - Si los datos son correctos el sistema crea el metadato - Concluye el requisito <p>Flujo alternativo. Datos incorrectos</p> <ul style="list-style-type: none"> -El sistema señala los datos incorrectos y permite corregirlos. -Se corrigen los datos. 	

Flujo alternativo. El usuario cancela la acción

-Concluye el requisito

Observaciones:

Prototipo de interfaz:

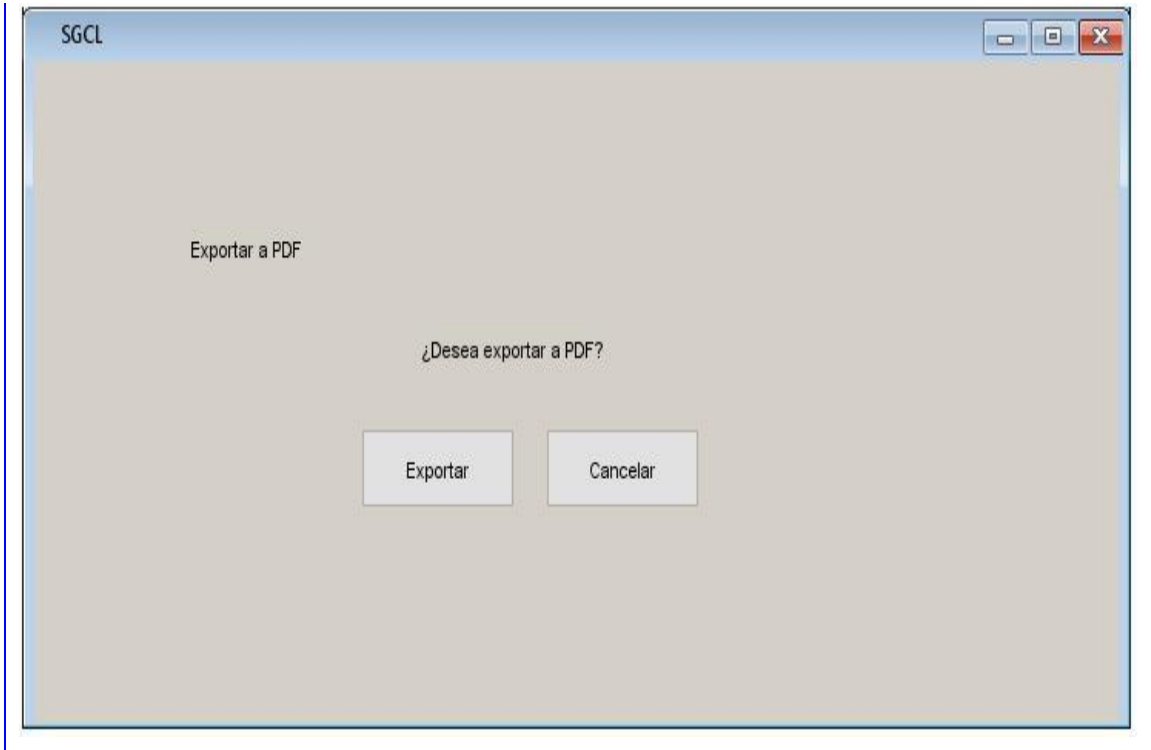
The screenshot shows a window titled "SGCLc" with a standard Windows-style title bar. Inside the window, the text "Lenguaje" is at the top left, followed by "Generar lenguaje". Below this, there are three tabs: "Información", "Solución", and "Meta-Datos", with "Meta-Datos" being the active tab. The form contains the following fields and controls:

- Estado:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Rol Contribución:** A text input field containing the text "rol del ciclo de vida".
- Versión:** A text input field containing the text "1,2,3...".
- Densidad semántica:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Dificultad:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Rol Contribución:** A text input field containing the text "rol de Meta-Dato".
- Formato:** A dropdown menu with the text "Seleccionar" and a downward arrow.
- Costo:** Two radio buttons labeled "No" and "Si". The "No" radio button is selected.
- Localización:** A text input field containing the text "url".

At the bottom of the form, there are two buttons: "Confirmar" and "Cancelar".

RF57 Exportar a PDF

Número: 60	Nombre del requisito: Exportar a pdf
Programador: Ernesto Alay Navarrete	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir exportar a pdf.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para Exportar a pdf hay que:</p> <ul style="list-style-type: none"> - Estar registrado en el sistema con el rol X. - Debe existir al menos un elemento de los que se desea exportar <p>3- Flujo de la acción a realizar: Se selecciona en la lista de elementos el que se desea exportar, seleccionando el botón Exportar. Se escoge en que formato se desea exportar, marcando la opción pdf. El sistema muestra un mensaje informando que se ha exportado correctamente.</p> <p>Flujo alternativo. El usuario cancela la acción</p> <ul style="list-style-type: none"> -Regresa a la vista previa 	
Observaciones	
Prototipo de interfaz:	



RF58 Minimizar barra

Número: 58	Nombre del requisito: Minimizar barra
Programador: Elianys Ferriol Medero	Iteración Asignada: N/A
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir minimizar barra en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para minimizar barra hay que: - Estar autenticado en el sistema con el rol X.</p> <p>3- Flujo de la acción a realizar: Se selecciona el botón de opción superior izquierdo</p>	

Flujo alternativo. El usuario cancela la acción
-Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF59 Maximizar barra

Número: 59		Nombre del requisito: Maximizar barra	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo: Permitir maximizar barra en el sistema.			
2- Acciones para lograr el objetivo (precondiciones y datos): Para maximizar barra hay que: - Estar autenticado en el sistema con el rol X.			
3- Flujo de la acción a realizar:			

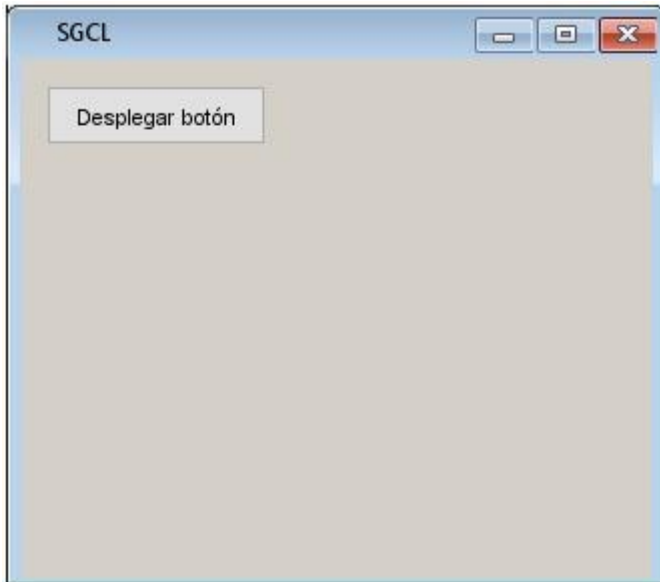
Se selecciona el botón de opción superior izquierdo

Flujo alternativo. El usuario cancela la acción

-Regresa a la vista previa

Observaciones

Prototipo de interfaz:



RF60 Asignar rol

Número: 60		Nombre del requisito: Asignar rol	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir asignar un rol a cada usuario autenticado.			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para asignar un rol a cada usuario autenticado hay que:			
- Debe existir al menos un administrador			

- Debe existir al menos un usuario autenticado.

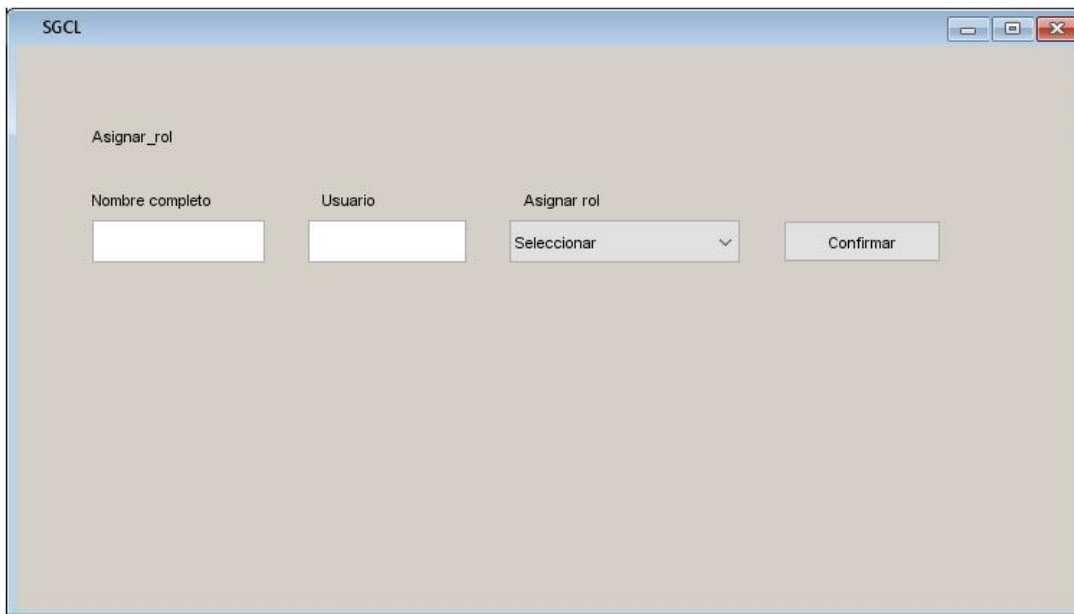
3- Flujo de la acción a realizar:

Se selecciona el botón de herramienta asignar rol, y luego se despliega y se selecciona el rol correspondiente al usuario que se desee.

Luego se selecciona la opción de confirmar

Observaciones

Prototipo de interfaz:



RF61 Asignar revisor

Número: 61		Nombre del requisito: Asignar revisor	
Programador: Elianys Ferriol Medero		Iteración Asignada: N/A	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir asignar el rol revisor al elemento correspondiente.			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para asignar un revisor hay que:			

- Debe existir al menos un revisor autenticado
- Debe existir al menos un elemento a revisar.

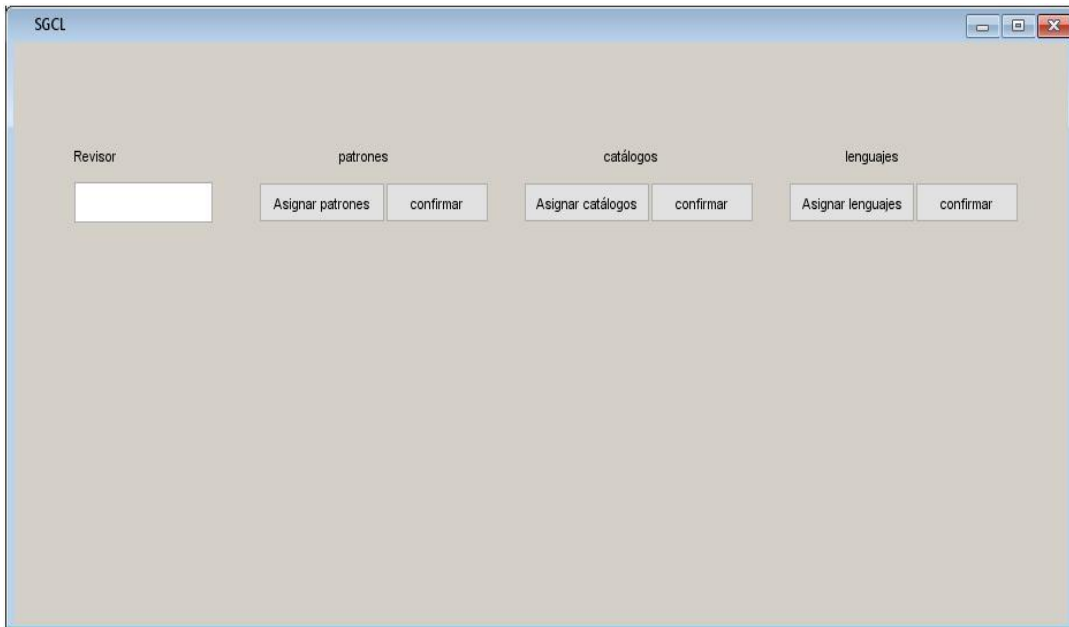
3- Flujo de la acción a realizar:

Se selecciona el botón de herramienta asignar revisor, y luego se selecciona para el revisor correspondiente el elemento a revisar mediante la opción asignar patrón.

Luego se selecciona la opción de confirmar

Observaciones

Prototipo de interfaz:



Anexo 2.

A continuación se muestran los casos de prueba SC2 y SC3 en correspondencia con los requisitos RF29 y RF41 respectivamente:

Descripción General:

Permitir mostrar los catálogos en revisión en el sistema.

Condiciones de ejecución:

- Estar autenticado en el sistema con el rol de revisor.
- El catálogo debe estar en lista de revisión.

SC2. Mostrar catálogo en revisión

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción de mostrar los datos del catálogo	Selecciona la opción de ver los datos de un catálogo y consulta sus datos.	Muestra los siguientes datos del catálogo: Nombre – Descripción del problema– Fundamento – Clasificación - Catálogo al que pertenece – Solución Permite además: - Editar los datos del catálogo. - Eliminar los datos del catálogo. - Cancelar la operación en cualquier momento.	Catálogo/ Catálogos en revisión/Mostrar
EC 1.2 Opción de cancelar la operación	El usuario selecciona la opción de Cancelar.	Regresa al listado de patrones y muestra un mensaje de información.	Catálogo/ Catálogos en revisión/Mostrar/Cancelar
EC 1.3 Opción de modificar los datos	Selecciona la opción de Editar los datos del catálogo.	El sistema brinda la posibilidad de modificar los datos del catálogo.	Catálogo/ Catálogos en revisión/Mostrar/Editar
EC 1.4 Opción de eliminar el catálogo	Selecciona la opción de Eliminar un catálogo.	El sistema brinda la posibilidad de eliminar un catálogo.	Catálogo/ Catálogos en revisión/Mostrar/Eliminar

Descripción General:

Permitir modificar los datos de un lenguaje existente

Condiciones de ejecución:

- Estar autenticado en el sistema con el rol de diseñador.
- Debe existir en el sistema al menos un lenguaje.

SC4. Eliminar lenguaje

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Eliminar lenguaje	Selecciona la opción de eliminar un lenguaje.	Muestra un mensaje de confirmación. Y permite: - Aceptar. - Cancelar.	Lenguaje/Lenguaje publicados/Mostrar/Eliminar
EC 1.2 Opción Aceptar	Selecciona la opción Aceptar.	<i>Elimina el lenguaje.</i> Regresa al listado de lenguajes actualizado y muestra un mensaje de información.	Lenguaje/Lenguaje publicados/Mostrar/Eliminar/Aceptar
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.	Regresa a la vista anterior.	Lenguaje/Lenguaje publicados/Mostrar/Eliminar/Cancelar

Anexo3.

CUESTIONARIO PARA EVALUAR LA ACEPTACIÓN DE SGCL (SISTEMA PARA LA GESTIÓN DE CATÁLOGOS Y LENGUAJES DE PATRONES DE DISEÑO) POR LOS USUARIOS POTENCIALES

En la Universidad de las Ciencias Informáticas se desarrolla un trabajo de diploma relacionado con la gestión de patrones para el diseño de recursos educativos. El resultado obtenido es el SGCL que permite el modelado visual de las relaciones entre los patrones que los conforman. Sobre este sistema responda las preguntas que aparecen a continuación.

Muchas gracias por su contribución.

Centro:	
Categoría docente:	Categoría Científica:
Rol o cargo:	

Años de experiencia en temas relacionados con tecnología educativa:

1. ¿Con frecuencia trabaja con patrones de diseño de recursos educativos?

Muchas veces

Frecuentemente

A veces

Pocas veces

Nunca

2. ¿Utilizaría el sistema SGCL para gestionar patrones de diseño de recursos educativos y sus colecciones?

Sí No No sé

3. ¿Considera usted que la implementación del modelado visual de las relaciones entre patrones facilita el entendimiento y el razonamiento dinámico de los usuarios?

Sí No No sé

4. ¿Qué considera de los componentes diseñados para el desarrollo de SGCL y su integración?

(Responda seleccionando una de las siguientes opciones):

Me satisfacen mucho

Más satisfecho que insatisfecho

Me es indiferente

Más insatisfecho que satisfecho

No me satisface

No sé qué decir

5. ¿Qué elementos del sistema SGCL considera positivos?

6. ¿Qué elementos considera negativos de este sistema?

7. ¿Qué sugerencias tiene para mejorar el sistema SGCL?