Universidad de las Ciencias Informáticas

Facultad 3

"Módulos Pedido y Nomencladores para la herramienta informática Sistema para la gestión de préstamos"

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas.

Autora:

YelinaCayón Romero

Tutores:

Ing. Yoslenys Roque Hernández

Ing. José Luis Guisao Jorge

La Habana, Septiembre de 2020

"Año 62 de la Revolución"

"Estoy convencido de que la mitad de lo que separa a los emprendedores exitosos de los que no triunfan, es la perseverancia."

-Steve Jobs-

DECLARACIÓN DE AUTORÍA

Declaro ser autora del presente trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

_____

YelinaCayón Romero

Firma de la autora

_____ _____

Ing. Yoslenys Roque Hernández Ing. José Luis Guisao Jorge

Firma del tutorFirma del tutor

DATOS DE CONTACTO

Autora: YelinaCayón Romero

Correo electrónico: ycayon@estudiantes.uci.cu

Tutor: Ing. Yoslenys Roque Hernández

Correo electrónico:yrhdez@uci.cu

Tutor: Ing. José Luis Guisao Jorge

Correo electrónico:jlguisao@uci.cu

AGRADECIMIENTOS

Agradezcoa mi familia por confiar en mí, por apoyarme desde lejos, por recordarme que mi objetivo en la universidad era salir como ingeniera, aunque tuviese dificultades no podía parar hasta llegar a ese objetivo que me había trazado al entrar en primer año.

Agradezco a los compañeros de aula que me ayudaban con las dudas ante de las pruebas parciales muchas veces hasta las 3 de la mañana, que tiempos de estrés aquellos.Agradezco a todos los profesores que me dieron clases, por su paciencia y comprensión.Agradezco a mis amistades por darme buenos consejos, por estar en los momentos difíciles, por ser mi familia más cercana.

Agradezco a la universidad por instruirme no solo académicamente sino también culturalmente, por tener la oportunidad de estudiar en una de las mejores universidades del país.Agradezco a las tías de los edificios en especial a Made que me ayudó y se preocupó mucho cada vez que me enfermaba.

Agradezco a mi hermanita Lena que fue mi fortaleza y a ella le debo mucho más de lo que puedo expresar.

Agradezco a las personas que creerían que no llegaría tan lejos y mira aquí llegue hasta el final.

Agradezco a mis tutores, los mejores, todo sin condiciones, por tenerme paciencia, por alarme las orejas, por estresarme, por ser tan entregados, por luchar por mí, por ser ellos y no otros les estaré eternamente agradecida.

Agradezco a mí por mis esfuerzos al no rendirme, cuando otros se quedaban en el camino, pasar por momentos muy tristes cuando suspendía una prueba me hacían recordar que tenía que estudiar más y barquear menos, que tenía que aprovechar el tiempo para ver el resultado, que todo se puede lograr si de veras ponía empeño. Agradezco por todo lo bueno y lo malo que pasé, porque gracias a todas las experiencias hoy soy una persona de bien.

A todos los que de una forma u otra han contribuido a la realización del presente trabajo. A todos aquellos que no he mencionado por falta de espacio pero que estuvieron ahí brindando su ayuda.A todos, muchas gracias.

DEDICATORIA

A mis padres y a mis abuelos por ser mi inspiración y a toda mi familia por serlo todo para mí.

A mis amigos por estar ahí cuando los necesito.

A todos muchas gracias por ayudarme a llegar a este día.

RESUMEN

La Dirección de Extensión Universitaria de la Universidad de las Ciencias Informáticas facilita el desarrollo cultural de la universidad. En el área de Administración de la dirección antes mencionada, el departamento de Servicioes el encargado de gestionar el proceso de entrada y salida de vestuario, atrezo y calzado para los Festivales de Artistas Aficionados y otras actividades extensionistas que se convocan. Actualmente, los pedidos se registran de forma manual, el control del vestuario se lleva en un libro donde el personal de administración autorizado recoge los datos personales de quien hace la solicitud.Por ello, el presente trabajo de diploma se traza como objetivo principal el desarrollode losMódulos Pedido y Nomencladores para la herramienta informática Sistema para la gestión de préstamos,de forma tal que contribuya a la celeridad en el proceso. Para guiar la propuesta de solución se empleó como metodología de desarrollo desoftwareProceso Unificado Ágil en su variación para el proceso productivo de la Universidad de las Ciencias Informáticas.Además, se utilizaron las herramientas, lenguajes y tecnologías teniendo en cuenta las características del presente trabajo. Al mismo tiempo, para garantizar la calidad del sistema se realizaron las validaciones correspondientes y se aplicaron las pruebas pertinentes obteniéndose como resultadolosMódulos Pedido y Nomencladores para la herramienta informática Sistema para la gestión de préstamos.

Palabras clave:celeridad, pedido, préstamo, vestuario

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

A partir del desarrollo de las nuevas tecnologías de la información y las comunicaciones (TIC), en el mundo se han informatizado diversos procesos, contribuyendo a su agilización y optimización. Actualmente, entre las proyecciones estratégicas de Cuba se encuentra la informatización de la sociedad(INAF, 2017). Algunos de los sectores que muestran pasos de avance en este sentido son: la economía, la educación, la rama del derecho y la salud.

La Universidad de las Ciencias Informáticas (UCI) juega un papel significativo en este proceso, pues parte de la misión de este centro es apoyar a la industria cubana de la informática. La Dirección de Extensión Universitaria (DEU) de la UCI, facilita el desarrollo cultural de la universidad y tiene varios departamentos: Programación, Movimiento Artístico, Promoción y Formación. Estos a su vez tienen departamentos asociados como son: de Compras, Mantenimiento de equipos e instalaciones, Control de documentos y registros, Tecnologías y el de Servicio.

Específicamente en el área de Administración de la DEU, el departamento de Servicio, es el encargado de gestionar el proceso de entrada y salida de vestuario, atrezo y calzado para los Festivales de Artistas Aficionados (FAA) y otras actividades extensionistas que se convoquen. La gestión de préstamos,es el servicio de procesos que proporcionan de manera eficiente la administración y agilización en la entrega de recursos materiales con la posibilidad de establecer un control de su entrega y devolución(Rodríguez, 2012).

Actualmente, los pedidosse registran de forma manual, el control del vestuario se lleva en un libro donde el personal de administración autorizado recoge los datos personales de quien hace la solicitud. En este proceso ocurren borrones y tachaduras, propias de la actividad humana que dificultan la interpretación de los pedidos. Las hojas se deterioran y se desprenden del libro por la interacción casi diaria con el mismo.Para solicitar préstamos hay que dirigirse al área donde se encuentra el cuarto de vestuario en la DEU. En múltiples ocasiones, las personas que allí acuden en busca del servicio no pueden acceder al vestuario que necesitan, pues el mismo se encuentra prestado o en la lavandería, lo que trae consigo pérdida de tiempo e inconformidad.

Las búsquedas de determinadas prendas o calzado y sus cantidades respectivas pueden causar demoras debido a que el vestuario disponible se encuentra catalogado en formato duro. Esto trae consigo que se necesite consultar el inventario para saber la cantidad exacta que hay de determinado tipo. En la etapa de festivales de las diferentes facultades ocurren pérdidas por el poco tiempo para controlar la masividad de

los pedidos. Las agrupaciones institucionales[1] tienen un libro de pedidos diferente lo que hace mayor el gasto de recursos, la pérdida de tiempo y dificulta el control. De no existir estas dificultades los procesos se realizarían con mayor agilidad, permitiendo que sus trabajadores disminuyan cualquier posibilidad de equivocación.

Ante la problemática anteriormente planteada surge el siguiente **problema a resolver:** ¿Cómo gestionar los pedidos yla información con poca variabilidad en el tiempoenel Sistema para la gestión de préstamos de forma tal que se contribuya a la celeridad en el proceso?

Tomando en cuenta el problema antes propuesto se define como**objeto de estudio:** sistemas de gestión.

De esta forma se determina como **objetivo general:** desarrollarlos módulos para la gestión de pedidos y nomencladores enel Sistema para la gestión de préstamos, de forma tal que se contribuya a la celeridad en el proceso.

Para ello se identifica como **campo de acción:** gestión de préstamos.

Por lo tanto, se plantea la siguiente **idea a defender**: si se desarrollanlos módulos para la gestión de pedidos y nomencladoresen el Sistema para la gestión de préstamos, se contribuye a la celeridad en el proceso.

Se desglosan del objetivo general, los siguientes **objetivos específicos:**

- Identificar los referentes teóricos en los que se sustenta la propuesta de solución sobre el desarrollo de sistemas informáticos dirigidos a la gestión de préstamos.
- Realizar la identificación de requisitos, análisis y diseño delosmódulos para la gestión de pedidos y nomencladorescomo aproximación a la implementación.
- Implementar losmódulos para la gestión de pedidos y nomencladorespara contribuir a la celeridad en el proceso.
- Validar el correcto funcionamiento de los módulos aplicando métricas y pruebas de software para garantizar la calidad del mismo, así como la variable de la investigación.

Para darle cumplimiento a los objetivos específicos se utilizaron los siguientes métodos clasificados en teóricos y empíricos:

1 Grupo de personas con un fin común que representan a una institución (Española, 2020). Agrupaciones artísticas (danza, música, teatro, artes visuales) que representan a la UCI.

Métodos teóricos:

- **Histórico-Lógico**: analiza la trayectoria completa del fenómeno y el método lógico se basa en el estudio histórico del fenómeno (Martínez Pérez, y otros, 2017). El empleo de este método posibilitó la realización del estudio del estado del arte para sentar las bases sobre los sistemas relacionados con la gestión de préstamos. Además, el mismo permitió adquirir conocimientos de la lógica objetiva del desarrollo respecto a cómo se realizan los préstamos en la DEU.
- **Análisis-Síntesis:**el análisis permite la división mental del fenómeno en sus múltiples relaciones y la síntesis establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas(Martínez Pérez, y otros, 2017). La utilización de este método permitió realizar el análisis teórico e identificar los principales conceptos a incluir en la fundamentación teórica, así como extraerlos elementos importantes relacionados con los sistemas de gestión de préstamos.
- **Modelación**: se utilizó para la realización de los diagramas necesarios en el proceso de desarrollo de software, donde se refleja la estructura de las relaciones, haciendo una representación abstracta de la solución, facilitando así el desarrollo de la misma.

Métodos empíricos:

- **Entrevista:** es una conversación planificada entre el investigador y el entrevistado para obtener información (Martínez Pérez, y otros, 2017).Se empleó con asesores y miembros calificados de la DEU para entender el proceso de negocio, comprender la estructura y funcionamiento de la organización, así como las deficiencias existentes que permitieron definir el problema a resolver y establecer el objeto de estudio. Además, se obtuvieron referencias e información útil para elaborar los requisitos de la investigación.
- **Observación:** posibilitó obtener el conocimiento necesario sobre el funcionamiento del Departamento de Serviciosdurante la gestión de los préstamos de la DEU. Permitió agruparpor datos a partir de criterios que se registran en los libros de pedidos. Facilitóla categorizaciónpor tipos la información que se registra como evidencia de los préstamos que se realizan.
- **Encuesta:** permitió mediante un cuestionario pre-elaborado (Ver anexo 1), obtener información sobre el tiempo real que demoran los trabajadores del Departamento de Servicios en realizar los préstamos en la DEU.

El presente trabajo de diploma está estructurado en tres capítulos de la siguiente forma:

Capítulo I. Fundamentación teórica: se aborda lo relacionado a la fundamentación teórica con el objetivo de definir los elementos necesarios para la realización de la investigación. Se especifican los conceptos asociados al dominio del problema. Se realiza el estudio de soluciones homólogaspara definir sus características y la posibilidad de su uso en la presente investigación.Se describe la metodología a seguir y se caracterizan las herramientas y lenguajes que serán utilizados durante el desarrollo de la solución.

Capítulo II. Descripción de la solución propuesta:en este capítulo se realiza una descripción de las principales características de los módulos a desarrollar. Se lleva a cabo la disciplina de requisitos, en la cual se engloban los requisitos funcionales y no funcionales. Se obtienen de igual forma los artefactos correspondientes a la disciplina de Análisis y Diseño aplicando los patrones de diseño definidos como buenas prácticas, los diagramas de clase del diseño, el modelo de datos y el patrón arquitectónico que da soporte a la propuesta de solución. Por último, se describen los estándares de codificación a tener en cuenta durante la implementación.

Capítulo III. Evaluación de la solución propuesta:en el capítulo se muestran los resultados obtenidos luego de aplicar las técnicas de validación de requisitos. Se valida el diseño aplicando las métricas Tamaño operacional de las clases y Relaciones entre clases, se aplican pruebas de software para verificar y revelar la calidad del producto de software antes de ser entregado al cliente. Por último, se realiza la validación de la variable de la investigación para comprobar en qué grado se cumplirá con la celeridad.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan los conceptos de interés asociados al dominio del problema. Se muestra un estudio del estado del arte de los sistemas de gestión que se relacionan con los servicios de préstamo a nivel mundial y en Cuba. Sedescribe la metodología propuestade acuerdo con las políticas que sigue la UCI para estandarizar el proceso de desarrollo de software. Se caracterizan las herramientas y lenguajes que serán utilizados para obtener los artefactos ingenieriles teniendo en cuenta la soberanía tecnológica a la que aspira el país.

1.1 Principales conceptos

Atrezo:(del italiano*attrezzo*), tanto en el teatro como en el cine y la televisión, es el conjunto de objetos que aparecen en escena(RAE, 2019).Son accesorios utilizados por los personajes para interactuar durante una representación artística o pequeños elementos que complementan la escenografía (como un jarrón o un cuadro)(Nieva, 2015).

Celeridad: rapidez, velocidad o prontitud en el movimiento o ejecución de algo(Pérez Porto, y otros, 2019). En el contexto de la investigación el término se utiliza para señalar la agilidad en los procesos.

Sistema de gestión:es una estructura para la gestión y mejora continua de las políticas, procedimientos y procesos de una organización, engloba información compartida, evaluaciones, trabajo en equipo y funcionamiento acorde con los principios de calidad y medioambiente(ONERP, 2020).

Vestuario: del vocablo *vestiré* que procede del latín, puede traducirse como ropa(RAE, 2019). Es el conjunto de prendas con que se cubre el cuerpo, incluye calzado y atrezo(Nieva, 2015).

1.2 Sistemas homólogos

Luego del análisis de los principales conceptos asociados al dominio del problema, se realiza un estudio de los sistemas homólogos tanto nacionales como internacionales con el objetivo de definir sus características y la viabilidad de su uso en la solución de la problemática existente.

QERP Moda de España: esun sistema de planificación de recursos empresariales que se hace cargo de distintas operaciones internas de una empresa. El sistema modular de esta plataforma ofrece un*Enterprise Resourse Product*(ERP[2]) para distintos productos, adaptados a las necesidades y peculiaridades del negocio. Entre las principales características destacan:

2*Enterprise Resourse Product* son los sistemas de información gerenciales que integran y manejan muchos de los negocios

- Sistema intuitivo y de fácil aprendizaje.
- Información para gestionar las compras de los artículos por temporadas, punto de venta, proveedor o por cualquier otra clasificación.
- Pedidos de compra y reposición con aviso automatizado.
- Recepción automática de alertas y comunicaciones en dispositivos móviles o fijos.
- Análisis de ventas o comportamientos de artículos.
- Consultas e informes sobre ventas, traspasos, ventas en tienda online, artículos más vendidos, horarios de más ventas(Pymes, 2018).

Sistema de gestión de préstamos para computadores portátiles: el Centro de gestión administrativa y fortalecimiento empresarial de Colombia ofrece a los aprendices, instructores y personal administrativo el préstamo de computadores portátiles; el objetivo de este servicio es facilitar la consulta a cada uno de ellos. Este servicio es administrado por una persona que es la encargada de hacer el préstamo y tener un control de cada uno delos computadores. Ante esta situación se vio la necesidad de crearun sistema de gestión de préstamos de portátiles el cual facilite este proceso y a la vez lo haga más rápido. El objetivo general es ofrecer,al administrador del servicio de préstamo de portátiles, un mejor manejo en cuanto a la información del usuario y el estado de los computadores(SCRIBD Inc, 2020).

Solución de automatización de procesos de trabajo XEROX para el procesamiento de solicitudes de préstamo (España):optimiza los procesos de trabajo de información y documentos asociados con las solicitudes de préstamos al consumidor, residenciales, hipotecarios, comerciales y de otros tipos, ayuda a:

- Reducir el tiempo dedicado al procesamiento de las solicitudes de préstamo.
- Aumentar el potencial de satisfacción del cliente.
- Reducir el coste del proceso en general.
- Lograr una comprensión y un control mayores sobre el proceso de solicitud de préstamo.
- Reducir las tasas de error.
- Conseguir una mayor visibilidad del estado de la solicitud de préstamo en la organización(XEROX corporation, 2020).

Sistema de gestión de préstamos de implementos deportivos: en la UCI se desarrolló esta aplicación con el fin de facilitar a la cátedra de cultura física el flujo de información asociado a los préstamos de implementos deportivos. Además,para satisfacer las demandas de forma rápida y sencilla proporcionando calidad en ese servicio. Actualmente este sistema no está desplegado. El objetivo general consiste

asociados con las operaciones de producción y de los aspectos de distribución de una compañía.

enofreceral administrador del servicio de préstamo de implementos deportivos, un mejor manejo en cuanto a la información del usuario. Las tecnologías utilizadas para el desarrollo de este sistema fueron Symfony v3 como marco de trabajo, PHP 5 y JavaScript para la implementación; Codelgniter como entorno de desarrollo abierto para crear webs dinámicas con PHP; entorno integrado plataforma Eclipse y Netbeans v6.9(Rodríguez, 2012).

Sistema Integrado de Gestión Bibliotecaria: Módulo de Préstamos y otros servicios: este sistema tiene como objetivo automatizar los procesos de la biblioteca de la UCI, orientado al servicio de préstamos y reservaciones, reproducción y digitalización.En el desarrollo del mismo se utilizócomo tecnología de programación del lado del cliente Javascript y del lado del servidor PHP(Monné, 2004).

Conclusiones del estudio de los sistemas homólogos

Tabla 1. Comparación de los sistemas homólogos estudiados.

Sistema	Licencia	Desplegado	Funcionalidades en común
QERP Moda	privativa	Si	Si
SCRIBD	privativa	Si	No
XEROX	privativa	Si	No
SIGPID	libre	No	No
SiGB	libre	Si	No

Fuente: elaboración propia.

Los sistemas internacionales analizados no resuelven el problema actual porque son herramientas privativas, no garantizan la soberanía tecnológica a la que aspira el país y no se pueden aprovechar sus funcionalidades. Por otra parte, los sistemas que se han desarrollado en Cubano tienen puntos en común con la actual propuesta y la mayoría de los sistemas de gestión de préstamos están relacionados a préstamos bancarios. En el caso del Sistema de gestión de préstamos de implementos deportivosse encuentra en desarrollo, por lo que no resuelve los problemas de la investigación. También, se sugiere tomar como ejemplo algunas de las funcionalidades del sistema QERP Modacomo la recepción automática de alertas y comunicaciones mediante el correo electrónico.

El estudio de sistemas homólogos permitió obtener conocimientos que se tuvieron en cuenta en la implementación de la propuesta.Además, lo anterior demuestra la necesidad de desarrollar la solución,con el propósito decontribuir a la política de soberanía tecnológica a la que aspira el país.

1.3 Metodología de desarrollo

Una metodología de desarrollo se refiere a un marco que se utiliza para estructurar, planificar y controlar el proceso. La metodología en el transcurso de la creación de una aplicación informática define quién debe hacer qué, cuándo y cómo hacerlo. En la actualidad existe una variada cantidad de metodologías de desarrollo de aplicaciones informáticas, clasificándose en dos grupos, tradicionales y ágiles. Las tradicionales están guiadas por una fuerte planificación, centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo(Pressman, 2010).

Las metodologías ágiles se centran principalmente en el factor humano o en la creación del sistema. Este tipo de metodologías le dan mayor importancia al individuo, a la colaboración con el cliente y al desarrollo incremental de la aplicación informática con iteraciones muy cortas(Sánchez, 2014).En el caso de la presente investigación, el proceso es guiado por la metodología definida por la UCI para organizar el proceso de desarrollo de software, la cual es una variación del Proceso Unificado Ágil (AUP).

Proceso Unificado Ágil (AUP) variación para la UCI

El Proceso Unificado Ágil (*Agile Unified Process* (AUP), por sus siglas en inglés) variante UCI (AUP-UCI) surge de la necesidad de converger a una única metodología de desarrollo que cubra las particularidades de los distintos centros productivos y estandarice el proceso productivo de la UCI. La misma describe de una manera simple y fácil de comprender, la forma de desarrollar software utilizando técnicas ágiles.

Con la adaptación de AUP que se propone para la actividad productiva de la UCI(Sánchez, 2015):

- Se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3.
- Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos.
- Se redujo a uno la cantidad de metodologías que se usaban y de más de veinte roles en total que se definían se redujeron a once.

AUP-UCI define 3 fases (Inicio, Ejecución, Cierre), 8 disciplinas (Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación, Pruebas de aceptación y Despliegue). Esta metodología es compatible con equipos de trabajos pequeños como es el caso de la

presente investigación y tiene un enfoque centrado en el cliente con iteraciones cortas lo cual permite la detección de errores tempranos(Sánchez, 2015).

En la metodología seleccionada existen 4 escenarios para la disciplina de Requisitos, el escenario 4 es el utilizado en la investigación ya que aplica a proyectos que obtengan un negocio bien definido. El cliente estará acompañando al equipo de desarrollo para convenir los detalles de los requisitos encontrados y poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no extensos, ya que las Historias de usuarios (HU) no deben poseer demasiada información.

1.4 Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Ordenador (*CASE por sus siglas en inglés:Computer Aided Software Engineering*) constituyen aplicaciones o programas informáticos destinados a aumentar el balance en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas sirven de ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras(Ambler, 2018).

Visual Paradigm V8.0

Constituye una herramienta CASE profesional, que utiliza *Unified Modeling Language*(UML[3] por sus siglas en inglés) como lenguaje de modelado y soporta el ciclo de vida completo del desarrollo de una aplicación informática, desde la fase de análisis hasta el despliegue. Algunas de sus características son:

- Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. El mismo proporciona abundantes tutoriales del lenguaje de modelado, demostraciones interactivas de UML y proyectos UML.
- Se integra con herramientas Java, como son: Eclipse/IBM, NetBeansIDE, entre otras.
- Es apoyado por un conjunto de lenguajes tanto en la generación del código como en la ingeniería inversa(Gaines, y otros, 2017).

Una vez analizadas las características de esta herramienta y que la UCI posee una licencia académica para su uso se decide utilizar Visual Paradigm en su versión 8.0. Además, por su robustez, usabilidad, portabilidad y que la misma soporta el ciclo completo del proceso de desarrollo de software.

[3]Lenguaje Unificado de Modeladoque propicia un conjunto de ayudas para el desarrollo de programas informáticos (Lucid, 2018).

1.5 Lenguaje de modelado UML v2.0

Es un lenguaje basado en diagramas para la especificación, visualización, construcción y documentación de cualquier sistema complejo. Estábasado en el paradigma orientado a objetos; es un modelo que simplifica la realidad que se construye para comprender mejor el sistema que se quiere desarrollar. Proporciona los planos del sistema, incluyendo una visión global como lo más detallado de alguna de sus partes(Pressman, 2010).Algunas de las actualizaciones de los diagramas UML son(Lucid, 2018):

- Mayor integración entre modelos estructurales y de comportamiento.
- Capacidad de definir jerarquía y desglosar un sistema de software en componentes y subcomponentes.
- Eleva el número de diagramas de nueve a trece.

Se selecciona el lenguaje de modelado UML para la confección de varios productos de trabajo de la solución debido a que es la notación utilizada por la herramienta CASE que se emplea para la creación de los módulos. Además, es el lenguaje estándar de modelado para software que emplea la metodología AUP.

1.6 Marco de trabajo

Un marco de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo del mismo.Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar(Acosta, 2017).

Symfony v4.3

Para el desarrollo del *backend,* que es la parte que procesa la entrada desde el *frontend*se utiliza este marco de trabajo completo diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Proporciona herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación(Potencier, 2019).

Para el desarrollo de la propuesta de solución se definió la utilización de Symfonyen su versión 4.3 debido a que es fácil de instalar y configurar en cualquier plataforma. De igual forma, las aplicaciones desarrolladas con Symfony son compatibles con la mayoría de las plataformas que existen y se adaptan a

entornos de negocio en cambio permanente, requiriendo menos esfuerzo para su mantenimiento. La presente investigación utiliza este marco de trabajo, teniendo en cuenta las características antes descritas.

Angular v6

El *frontend*es el responsable de recolectar los datos de entrada del usuarioy transformarlos hasta ajustarlos a las especificaciones que demanda el *backend* para poder procesarlo. Para el desarrollo del *frontend* se selecciona este marco de trabajo de JavaScript de código abierto para crear y mantener aplicaciones web de una sola página. Este marco de trabajo adapta y amplía el HTML tradicional para servir mejor contenido dinámico.

Los principales objetivos de diseño que se siguen con su uso son los siguientes(Wilson, 2018):

- Guiar a los desarrolladores a través de todo el proceso del desarrollo de una aplicación: desde el diseño de la interfaz de usuario, a través de la escritura de la lógica del negocio, hasta las pruebas.
- Usar la inyección de dependencias, para llevar servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente. En consecuencia, gran parte de la carga en el *backend* (capa de acceso a datos) se reduce, lo que conlleva a aplicaciones web ligeras.
- Separar el lado del cliente, de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.

Para el desarrollo del sistema se decide utilizar Angular en su versión 6.0, teniendo en cuenta las características antes descritas. Además, los módulos de la presente solución forman parte de un sistema que define este marco de trabajo como parte de su arquitectura base.

Boostrap v4.1

Boostrap es de código abierto y está disponible en *GitHub*[4]. Es un marco de trabajo o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en *HyperText Markup Language*(HTML por sus siglas en inglés) y *Cascading Style Sheets*(CSS por sus siglas en inglés), así como extensiones de JavaScript adicionales. A diferencia de muchos marcos de trabajo, solo se ocupa del desarrollo *frontend*(capa de presentación).Los desarrolladores pueden adaptar el mismo archivo de Boostrap, seleccionando los componentes que deseen usar en su proyecto(Boostrap Team, 2018).

[4]GitHub: plataforma de desarrollo colaborativo(Robles, 2018).

Para el desarrollo de los módulos se decide la utilización de Bootstrap en su versión 4.1 debido a las nuevas funcionalidades que el mismo integra, por ejemplo: brinda la posibilidad de asignarle ancho y altura a los elementos de la página web de forma automática por lo que se adapta a todos los dispositivos, proporciona una galería de iconos de forma gratuita, lo que facilita incorporar cada uno de estos elementos a la propuesta de solución. Además, se hace un mejor uso de las transiciones, animaciones y efectos en las páginas web.

1.7 Lenguajes de programación

Se utilizan para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas(Domínguez, 2018).

PHPv7.2

Hypertext Preprocessor(PHP) es un lenguaje del lado del servidor (esto significa que funciona en un servidor remoto que procesa la página web antes de que sea abierta por el navegador del usuario). A continuación, se muestra algunas de las funcionalidades que brinda(The PHP Group, 2019):

- Es un lenguaje de secuencia de comandos de servidor, diseñado específicamente para la web. Dentro de una página web puede incluir código PHP que se ejecute cada vez que se visite una página.
- El código PHP es interpretado en el servidor web y genera código HTML y otro contenido que el visitante verá.
- Dispone de una conexión propia a los sistemas de base de datos.

Para el desarrollo de la actual propuesta se decide la utilización de PHP en su v7.2 teniendo en cuentaque es el lenguaje definido para el Sistema para la gestión de préstamos (XIGPREST). Los módulos de la presente investigación forman parte del sistema por lo que se asume este lenguaje para la implementación de los mismos.

JavaScript v1.8

Es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado[5] y dinámico. Se utiliza principalmente en su forma del lado del cliente (*client-side*),

[5]Tipado: se refiere a cómo declaramos los tipos de variables.

implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (*Server-side JavaScript o SSJS*(MDN, 2018).

Para el desarrollo de los módulos de la presente solución se decide la utilización de JavaScript en su versión 1.8, ya que está en concordancia con la versión seleccionada para el XIGPREST.

HTML 5

HTML estádiseñado para estructurar textos y presentarlos en forma de hipertexto[6], que es el formato estándar de las páginas web. HTML 5 es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección muestra la forma en que se presenta la información y la manera de interactuar con ella. Permite una mayor interacción entre páginas web y contenido media (video, audio, entre otros) así como una mayor facilidad a la hora de codificar el diseño básico.

Esta nueva versión se basó en el diseño más común de las páginas web alrededor del mundo, para llegar a un estándar de etiquetas que realicen las mismas tareas de manera más rápida y eficiente. Además, ocupa menos recursos en la computadora del cliente y es compatible con las versiones anteriores de HTML(Bos, 2019).Para el desarrollo del sistema se decide la utilización de HTML en su versión 5, teniendo en cuenta las características antes descritas.

CSS 3

Lashojas de estilo en cascada es una tecnología usada para el diseño web. Posibilitala opción de dar formato y estilo a varias páginas web al mismo tiempo. Las nuevas características de CSS 3 respecto a sus versiones anteriores incluyen soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones y transparencia, permitiendo una mayor elegancia y estética en aquellas aplicaciones donde se utilice (Bos, 2019).

Se decide la utilización de CSSen su versión 3,teniendo en cuenta las ventajas antes descritas. Además, permite tener separados los estilos de cada página haciendo más sencilla su manipulación.

Typescript v3.2

Es un lenguaje de programación libre y de código abierto. Principalmente se caracteriza por implementar las ventajas de un lenguaje orientado a objetos y las ventajas de JavaScript. Este lenguaje de

[6] Es texto que contiene enlaces a otros textos (WordReference.com, 2019).

programación está especialmente indicado para desarrollar proyectos de una cierta envergadura, teniendo en cuenta la integración con distintos editores y entornos de desarrollo(Microsoft, 2018).

Para el desarrollo XIGPREST se decide utilizar TypeScript en su versión 3.2, ya que es el requerido para la versión de Angular seleccionada. Por tanto, se utiliza esta versión para el desarrollo de los módulosde la propuesta de solución.

1.8 Sistema Gestor de Bases de Datos

Es una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Son programas que admiten crear y conservar las bases de datos asegurando su integridad, confidencialidad y seguridad (Alvarez, 2012).

PostgreSQL v10.1

Es uno de los sistemas gestores de bases de datos de código abierto más avanzado del mundo.PostgreSQL incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD[7](The PostgreSQL Global Development Group, 2019).

Dentro de sus principales ventajas se encuentran:

- Soporte de protocolo de comunicación encriptado por SSL[8].
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Se decide la utilización de PostgreSQL v10.1, teniendo en cuenta las características antes descritas y las exigencias del cliente. Además, se tiene en cuenta el dominio de la herramienta por parte de la autora del trabajo y que la arquitectura base del XIGPREST define esta versión para el desarrollo de la misma.

[7]Licencia BSD: licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva. La licencia BSD permite el uso del código fuente en software no libre(SBD, 2016).

[8]Seguridad de la capa de transporte: Constituye un cifrado que ayuda a proteger los datos y garantizar la privacidad de las comunicaciones entre los puntos finales (Incorporated, 2016).

PgAdmin3 v1.22

Es una aplicación de diseño y manejo de bases de datos para su uso con postgreSQL. Se puede utilizar para manejar postgreSQL 7.3 y superiores y funciona sobre casi todas las plataformas.La interfaz gráfica es compatible con todas las características de postgreSQL. Este softwarefue diseñado para responder a diversas necesidades, desde la escritura de simples consultas SQL(*Structured Query Language)* a la elaboración de bases de datos complejas (Page, y otros, 2018).

Se decide la utilización dePgAdmin3 v1.22 por su compatibilidad con la versión de PostgreSQL seleccionada. Además, se tienen en cuenta las características antes descritas en la propuesta de solución.

1.9 Mapeador Objeto-Relacional

Permite convertir los datos de los objetos en un formato correcto para poder guardar la información en una base de datos (mapeo) creándose una base de datos virtual donde los datos que se encuentran en la aplicación, quedarán vinculados a la base de datos. Esto consiste en transformar toda la información que se recibe de la base datos, principalmente en tablas, en los objetos de la aplicación y viceversa (Zoega, 2010).

Doctrine v2.4

Es un mapeador de objetos-relacional (ORM por sus siglas en idioma inglés) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Además,permite reducir el código necesario para llevar a cabo las operaciones de persistencia y recuperación de objetos. Es una capa de abstracción que se sitúa justo encima de un SGBD. Dentro de sus características tiene(Doctrine Team, 2019):

- Bajo nivel de configuración que necesita para empezar un proyecto. Puede generar clases a partir de una base de datos existente y después el programador puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas.
- Combina varios archivos PHP del marco de trabajo en uno solo para evitar el descenso de rendimiento que provoca incluir varios archivos PHP.

Para el desarrollo de la actual propuesta se decide la utilización de Doctrine en su v2.4teniendo en cuenta las características antes descritas y las exigencias del cliente.

1.10 Servidor web

Es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Se trata de programas que median entre el usuario de Internet y el servidor en donde está la información que solicita(Raffino, 2019).

Apache v2.4

Se caracteriza por ser estable, modular, tener código abierto y ser gratuito. Además, es altamente configurable de acuerdo a las necesidades de la organización que lo utilice. Ofrece módulos especializados en distintas actividades, integración con lenguajes de programación en el lado del cliente, Apache v2.4 es una profunda revisión del servidor Apache, centrándose en la escalabilidad, seguridad y rendimiento. Las principales revisiones de código se han llevado a cabo para crear una arquitectura realmente escalable(Doctrine Team, 2019).

Se decide la utilización de Apache v2.4, teniendo en cuenta las características antes descritas. Además, presenta mejoras en cuanto al rendimiento al minimizar el consumo de memoria y en las concurrencias de las peticiones.

1.11 Sistema para el control de versiones en la propuesta de solución.

Un sistema de control de versiones (SCV), es un software que controla y organiza las distintas revisiones que se realizan sobre un archivo o conjunto de archivos a lo largo del tiempo de tal manera que sea posible recuperar versiones especificas más adelante(Durante Lerate, y otros, 2009).

Git v2.17

Es un software de control de versiones diseñado por Linux Torvalds. Proporciona las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida. Algunas de las características más importantes son(Robles, 2018):

- Gestión distribuida: los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes y re-almacenamiento periódico en paquetes.

Se decide utilizar Git en la propuesta de solución teniendo en cuenta las características antes descritas y las tendencias actuales de la universidad.

1.12 Entorno de desarrollo (IDE)

Un entorno de desarrollo integrado o entorno de desarrollo interactivo (*Integrated Development Environment*: IDE por sus siglas en inglés), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software. Consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de estos tienen auto-completado inteligente de código(IntelliSense). Algunos IDE contienen un compilador, un intérprete, o ambos, tales comoNetBeans y Eclipse(Ramos Salavert, y otros, 2011).

Netbeans v8.2

Es un entorno de desarrollo integrado modular, escrito en el lenguaje de programación Java. El proyecto NetBeansconsiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de aplicación. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software(Domínguez, 2018).

Se decide utilizar Netbeans v8.2 en la propuesta de solución teniendo en cuenta las características antes descritas y las exigencias del cliente. Además, proporciona soporte para el cliente de control de versiones seleccionado.

1.13 Conclusiones del capítulo

- El análisis de los conceptos asociados al problema a resolver facilitó la comprensión del objeto de estudio de la presente investigación.
- El estudio del estado del arte permitió obtener las características de sistemas homólogos nacionales e internacionales y ratificó la necesidad de crear una solución que resuelva las exigencias del cliente.
- La selecciónde la metodología AUP en su variación para la UCIpone en línea la presente solución con las políticas que sigue la universidad en el uso de la misma para obtener productos demayor calidad que satisfagan las necesidades del cliente.
- La selección de las herramientas y lenguajes de desarrollo está en armoníacon la independencia tecnológica a la que aspira el país.

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

En el presente capítulo se abordan los principales elementos relacionados con la propuesta desolución. Se identifican los requisitos funcionales y no funcionales, obteniéndose la base de la arquitectura del sistema, así como particularidades del diseño e implementación. A partir de los requisitos identificados se modelan los diagramas de clases del diseño los cuales permiten obtener una visión más clara de la implementación del sistema. Además, se obtiene el diagrama de componentes que representa la relación y las dependencias entre los mismos.

2.1 Descripción de la propuesta de solución

La propuesta de solución formaparte delXIGPREST. La misma consiste en el desarrollo de los módulos: pedido y nomencladores como se muestra en la siguiente figura:

Figura 1. Módulos del XIGPREST.

Fuente:elaboración propia.

En el caso del módulo para los nomencladores se podrá gestionar la información que tiene poca variabilidad en el tiempo. El administrador del sistema podrá gestionar el tipo de vestuario, calzado y atrezo, los estados de los pedidos y de las notificaciones que serán manejados al sistema. Por su parte, el módulo para Pedidotendrá dentro de sus funcionalidades después de inventariados los medios del cuarto de vestuario, realizarsolicitudes de vestuario, atrezo y calzado en dependencia de las necesidades.

2.2 Requisito de software

Un requisito es la condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo. También se define como la condición o capacidad que debe exhibir o poseer un sistema para satisfacer un contrato, estándar, especificación, u otra documentación formalmente impuesta(Sanchez, 2018).

2.2.1 Requisitos funcionales

Los requisitos funcionales (RF) de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a tener en cuenta en el sistema y otros requerimientos de negocio(Pressman, 2010).Se identificaron20 RF, en la Tabla 2 se muestra la especificación de estos requisitos:

Tabla 2. Lista de RF de los módulos pedidos y nomencladores del XSIGPREST.

No.	Nombre	Descripción
RF 1.	Adicionartipo de prenda	Funcionalidad que permitirá adicionar un tipo de prenda al sistema.
RF 2.	Modificartipo de prenda	Funcionalidad que permitirá modificar los datos de untipo de prenda existente en el sistema.
RF 3.	Eliminartipo de prenda	Funcionalidad que permitirá eliminar untipo de prenda existente en el sistema.
RF 4.	Listar tipos de prenda	Funcionalidad que permitirá listar los tipos de prenda existentes en el sistema.
RF 5.	Adicionartipo de calzado	Funcionalidad que permitirá adicionar tipo de calzado al sistema.
RF 6.	Modificartipo de calzado	Funcionalidad que permitirá modificar los datos de un tipo de calzado existente en el sistema.
RF 7.	Eliminartipo de calzado	Funcionalidad que permitirá eliminar un tipo de calzado existente en el sistema.
RF 8.	Listar tipo de calzado	Funcionalidad que permitirá listar lostipos de calzado existentes en el sistema.
RF 9.	Adicionartipo de atrezo	Funcionalidad que permitirá adicionar tipo de

		atrezo al sistema.
RF 10.	Modificartipo de atrezo	Funcionalidad que permitirá modificar los datos de un tipo de atrezo existente en el sistema.
RF 11.	Eliminartipo de atrezo	Funcionalidad que permitirá eliminar un tipo de atrezo existente en el sistema.
RF 12.	Listar tipos de atrezo	Funcionalidad que permitirá listar lostipos de atrezo existentes en el sistema.
RF 13.	Adicionarestado	Funcionalidad que permitirá adicionar un estado al sistema.
RF 14.	Modificarestado	Funcionalidad que permitirá modificar los datos de un estado existente en el sistema.
RF 15.	Eliminarestado	Funcionalidad que permitirá eliminar un estado existente en el sistema.
RF 16.	Listar estados	Funcionalidad que permitirá listar los estados existentes en el sistema.
RF 17.	Adicionarpedido	Funcionalidad que permitirá adicionar un pedido al sistema.
RF 18.	Modificar pedido	Funcionalidad que permitirá modificar los datos de un pedido existente en el sistema.
RF 19.	Eliminar pedido	Funcionalidad que permitirá eliminar un pedido existente en el sistema.
RF 20.	Listar pedidos	Funcionalidad que permitirá listar los pedidos existentes en el sistema para cada usuario.

Fuente: elaboración propia.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales (RnF) representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir(Pressman, 2010).

Seguidamente se presenta un resumen de los mismos teniendo en cuenta los que inciden directamente en los módulos que forman parte de la presente solución.

Usabilidad

RnF 1. Requisito de usabilidad 1.
En el sistema se deben visualizar todos los mensajes en idioma español. La tipografía debe ser uniforme, de un tamaño adecuado.

RnF 2. Requisito de usabilidad 2.
El sistema debe facilitar la interacción con el usuario, posibilitando la utilización de combinaciones de teclas, podrán utilizarse los campos de selección en la interfaz en los casos que sea posible y se agruparán los vínculos y botones por grupos funcionales siempre que se cumpla con las pautas de diseño de las interfaces.

RnF 3. Requisito de usabilidad 3.
El sistema debe ofrecer una interfaz amigable, fácil de operar. Igualmente tiene que mantener la línea de diseño establecida la cual mantiene la uniformidad y representatividad de la solución. Las interfaces deben poseer un diseño sencillo, con pocas entradas, permitiendo un balance adecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios la utilización del mismo.

Restricciones del diseño y la implementación

RnF 4. Requisito de restricción de diseño e implementación 1.
La PC cliente del sistema debe poseer resolución de 1024 por 768 píxeles o superior.

RnF 5. Requisito de restricción de diseño e implementación 2.
Se utilizará la herramienta CASE Visual Paradigm teniendo en cuenta sus ventajas para modelar los diferentes artefactos que se obtienen en los flujos de trabajo y sus diferentes fases. Las restricciones propias del diseño radican en las pautas que se establecerán, así como las diferentes relaciones que se formen durante el modelado del sistema.

Interfaz de usuario

RnF 6. Requisito de interfaz de usuario 1.
En el sistema se evitará el uso de letras en negrita, estas se usarán solo en los encabezados de las tablas y el título de los grupos de las funcionalidades que aparecen en el menú lateral.

RnF 7. Requisito de interfaz de usuario 2.
Se usarán los siguientes *widgets* en los formularios: *checkbox* cuando se tiene menos de 4 elementos posible a seleccionar y se pueden seleccionar varios, en caso de que se deba seleccionar solo uno, se

utilizará radio *button; select* para seleccionar elementos de una lista en las que existan más de 3 posibles elementos; *textfield* para los campos de texto y textarea para textos de mayor longitud. Todos los campos tendrán un *label*, el mismo se encontrará encima del *widget* y contendrán (*) de color rojo en caso de ser obligatorio el campo, los mensajes de validaciones serán de color rojo y se mostrarán debajo de *widget* y de igual forma, pero de un color más opaco se mostrará algún mensaje de ayuda del campo.

RnF 8. Requisito de interfaz de usuario 3.
Las listas se mostrarán en tablas, las mismas se paginarán de 10 elementos por páginas siempre del lado del servidor, las acciones en lotes y demás acciones generales sobre los elementos de la misma, serán ubicadas al lado superior izquierdo de la tabla, las acciones individuales sobre los elementos se ubicarán en cada una de las filas, siempre al final, en la última columna, en forma de botones.

RnF 9. Requisito de interfaz de usuario 4.
Los botones tendrán dos tonalidades de colores, los de acciones primarias, que serían aquellas acciones que terminan o completan una acción, lo que siempre se quiere lograr, serían los que llamen la atención a la vista y los secundarios, que son las acciones que se pueden hacer, pero no es lo que debe de ocurrir normalmente, ejemplo la opción“cancelar”, serán de un color más claro.

2.3 Historias de usuario

Constituyen descripciones cortas y esquemáticas que resumen la necesidad concreta de un usuario al utilizar un producto o servicio, así como la solución que la satisface. Su función principal es identificar problemas percibidos, proponer soluciones y estimar el esfuerzo que requieren implementar las ideas propuestas(Solvingadhoc, 2017).En la presente solución se definieron 20 historias de usuario de las cuales se presenta, por su alta complejidad, la correspondiente al RF17.

Tabla 3. HU 17. Adicionar pedido.

Número: HU 17	**Nombre delos requisitos:** RF 17.Adicionar pedido.
Programador: YelinaCayón Romero	**IteraciónAsignada:**1
Prioridad: Alta	**TiempoEstimado:** 8 horas
Riesgo en Desarrollo: N/A	**Tiempo Real:** -
Descripción:funcionalidad que permitirá adicionar un pedido al sistema.	

Campos

Solicitar pedido: campo *Checkbox*que permitirá seleccionar una opción entre Atrezo, Calzado o Prenda. Obligatorio.

Tipo de atrezo: Campo de selección que permitirá escoger el tipo de atrezo. Obligatorio.

Imagen: Archivo en formato .png o .jpg que muestra el tipo de atrezo seleccionado.

Cantidad disponible: campo informativo que permitirá mostrar la cantidad disponible de lo solicitado.

Cantidad a solicitar: campo numérico que permitirá escribir la cantidad a solicitar. Obligatorio.

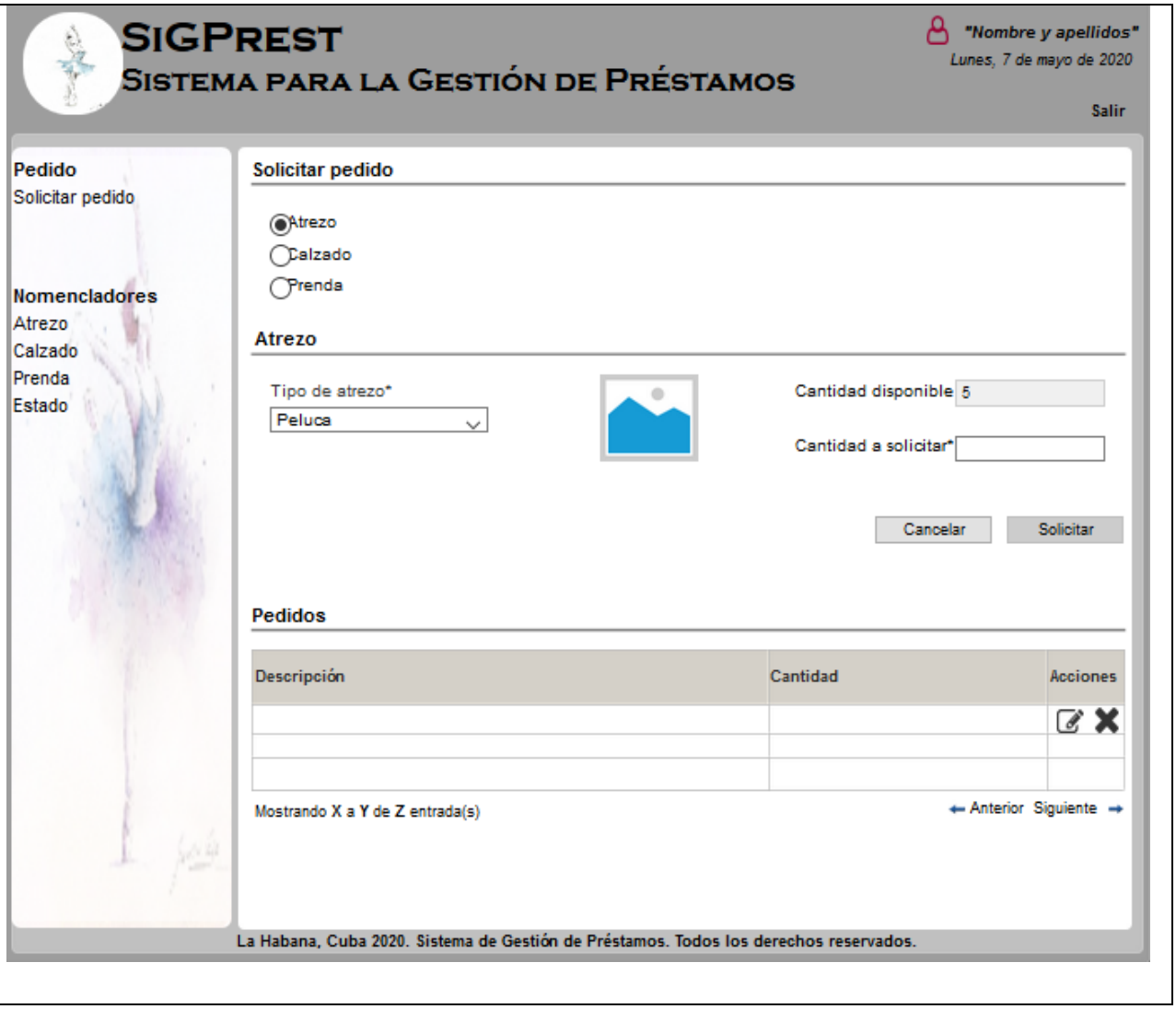
Botones

Solicitar: permite validar y registrar los datos del pedido.

Cancelar: permite regresar a la interfaz principal, sin tener en cuenta los cambios realizados.

Observaciones: debe estar autenticado en el sistema.

Prototipo de interfaz



Fuente: elaboración propia.

2.4 Arquitectura del sistema

La arquitectura de softwarese definecomo la estructura de un sistema, compuesta de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. La arquitectura desoftware tiene un impacto directo sobre lacapacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema (Cervantes, 2018).La solución propuesta tiene como base la arquitectura del XIGPREST, separada en *frontend* y*backend.*

2.4.1 Arquitectura *frontend*

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios(Nicole, 2018).

En la siguiente figura se muestra el diagrama de componentes correspondiente al *frontend* del módulo Pedido, donde se evidencia la arquitectura de la presente propuesta de solución.

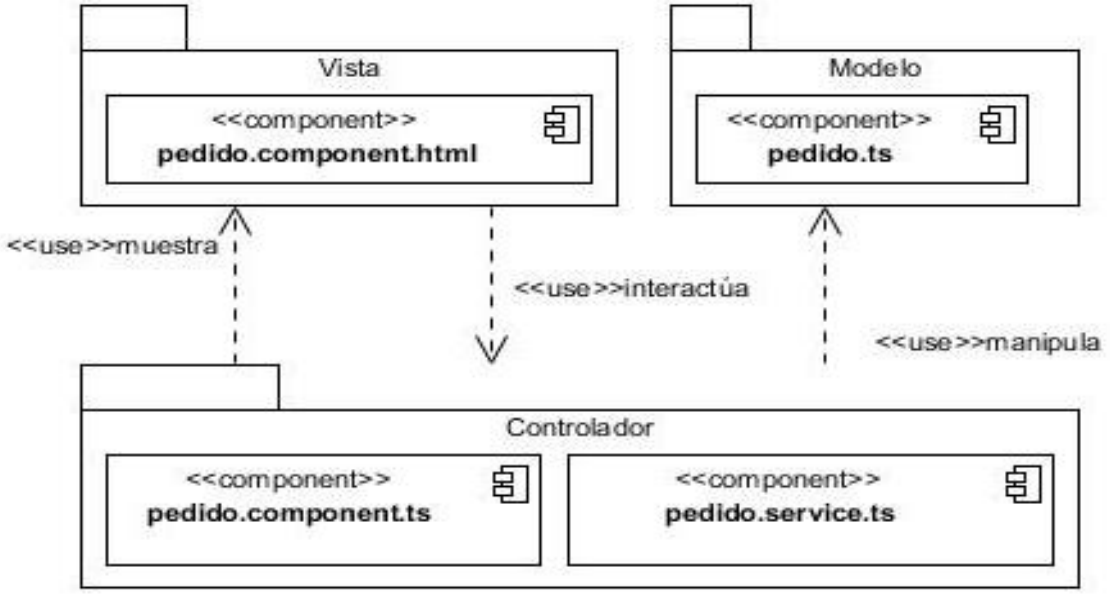


Figura 2: Diagrama de componentes correspondiente al *frontend*del móduloPedido.

Fuente:elaboración propia.

En la figura 2, el modelo para elmódulo está compuesto por el archivo*pedido.ts*, dicho archivo contiene información que se le solicita al usuario para gestionar los datos del pedido. La vista está compuesta por el archivo*pedido.component.html* que representa la información visible al usuario e interactúa con funciones específicas del controlador. Por otra parte, el controlador está compuesto por el archivo*pedido.component.ts* y *pedido.service.ts*, encargados de interactuar con el modelo y mostrar información a la vista.

2.4.2 Arquitectura *backend*

En la parte del *backend,*los módulos propuestos inciden en las capas controladora, gestora y acceso a datos. En la figura 3 se muestra cómo se representa en el *backend* del módulo Pedido la arquitectura de la presente propuesta de solución.

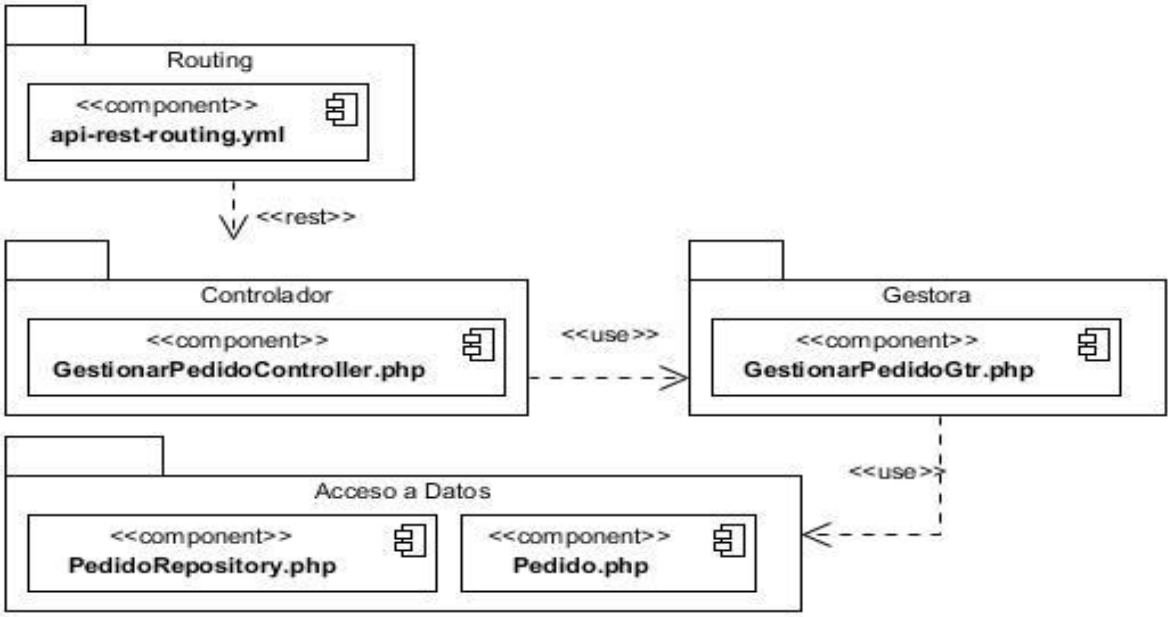


Figura 3: Diagrama de componentes correspondiente al *backend* del móduloPedido.

Fuente:elaboración propia.

En la figura 3, el componente*GestionarPedidoController.php*de la capa Controladora usa los métodos de la capa Gestora *GestionarPedidoGtr.php*, el cual usa las clases de la capa de Acceso a Datos *PedidoRepository*.php. La relación entre los componentes del *frontend* y del *backend* se realiza a través del componente *api-rest.routing.yml* de la capa *Routing.*

2.4.3 Patrón arquitectónico

Un patrón de arquitectura de software resuelve los problemas relacionados con el estilo arquitectónico, representa la relación entre los componentes del sistema y cada uno de ellos está compuesto por pequeños módulos (Advance, 2019).

Symfony basa su funcionamiento interno en el estilo Modelo-Vista-Controlador (MVC), uno de los más usados(Eguiluz, 2011). Este se aplicará para el diseño de las clases y propone tres capas fundamentales como lo indica su nombre, las cuales se describen a continuación:

Vista

La capa de presentación o vista es la encargada de proveer la interacción con el usuario y facilitar la visualización de las funcionalidades del sistema, tales como:

- Mostrar datos, eliminarlos, ordenarlos, solicitarlos, validarlos.
- Informar de los errores lógicos y de ejecución.
- Controlar la navegación entre pantallas.

Esta capa sería la encargada de visualizar al usuario los datos referentes al proceso de solicitarpedidosregistrados en la base de datos. En el caso de la presente investigación la vista estaría separada del resto de las capas como se evidencia en la Figura 2.

Controlador

Esta capa contiene la funcionalidad que implementa la aplicación. Involucra cálculos basados en la información dada por el usuario, datos almacenados y validaciones. Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica de la capa de negocios para uso directo por parte de componentes de presentación o su encapsulamiento como servicio y llamada a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio.

Esta capa sería la encargada de ajustar las funcionalidades para capturar los eventos y registrar los datos en el sistema. Se encontrarían también las clases controladoras encargadas de manejar la comunicación entre la vista y el modelo como se representa en la Figura 3.

Modelo

La capa de datos o modelo no es más que un grupo de clases responsables del almacenamiento de los datos, esta incluye necesariamente un modelo de las entidades del dominio del negocio. El modelo, es la representación real de los datos y representa además la persistencia del estado del sistema.

Esta capa se representan las clases entidades. Cubre también aquellas clases que almacenan las consultas para acceder a los registros de la base de datos como se representa en la Figura 3.

2.5 Patrones de diseño

Los patrones de diseño son técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Los mismos deben ser reutilizables, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias(Pressman, 2010).

Los patrones constituyen una abstracción de una solución en un nivel alto. Además, solucionan problemas que existen en muchos niveles de abstracción. Hay patrones que abarcan las distintas etapas del

desarrollo, entre ellos los patrones GRASP (*General Responsability Assignment Software Patterns* o Patrones Generales para Asignar Responsabilidades) y GoF (*Gang of Four* o Banda de los Cuatro)(Giraldo, et al., 2011).

2.5.1 Patrones GRASP

Los Patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones(Usaola, 2019).

Symfony v4.3 evidencia el uso de varios patrones de diseño que este trae incluidos por defecto en su arquitectura, además de estar concebidos de tal manera que obliga al programador a aplicarlos. Entre ellos se tienen:

- **Experto**: es uno de los que más se utiliza cuando se trabaja con *Symfony* v4.3, con la inclusión de *Doctrine* para mapear la base de datos. El patrón experto en información (también experto o el principio experto) es un principio utilizado para determinar dónde delegar responsabilidades tales como métodos, campos computados entre otros(Source Code Examples, 2019). La figura 4 muestra la clase entidad *Pedido.php,* la cual es la experta en información para el tratamiento delospedidos.

```
/** Pedido ...*/
class Pedido
{
    /** @var int ...*/
    private $id;

    /** @var int ...*/
    private $cantDisponible;

    /** @var int ...*/
    private $cantSolicita;
```

Figura 4:representación del patrón Experto en la clase entidad *Pedido.php*.

Fuente: elaboración propia.

- **Creador:**en las clases controladoras se encuentran las funcionalidades, las acciones definidas para el sistema y se ejecutan cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia cuáles funcionalidades son creadoras de dichas entidades.En la figura 5 se evidencia como, desde la clase *GestionarPedidoGtr.php* se hace instancia de la clase *Pedido.php* al crear un objeto de tipo Pedido.

```
$pedido= new Pedido();
$usuario= $this- >getEm()- >find( className: User::class, $usuarioPedido[¨usuario¨] [¨id¨]);
$numerPedido= $usuario- >getNumeroPedido();
```

Figura 5:representación del patrón Creador en la clase controladora*GestionarPedidoGtr.php*.

Fuente: elaboración propia.

- **Controlador:**todas las peticiones web son manipuladas por un controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases controladoras del sistema. *Symfony*v4.3 tiene una capa específicamente para los controladores, que son el núcleo de este marco de trabajo; aquí cada clase tiene su responsabilidad y es única.Para una mejor comprensión se muestra la figura 6, en la cual se evidencia el uso del patrón en la clase *GestionarPedidoController.php* correspondiente al módulo Pedido de la propuesta de solución:

```
/** Pedido ...*/
class GestionarPedidoController extends BaseApiController
{
    /** @autor Yelina  <ycayon@uci.cu>  ...*/
    public function listarPedidosAction( GestionarPedidoGtr  $gestionarPedidoGtr){...};

/** @autor Yelina  <ycayon@uci.cu>  ...*/
    public function adicionarPedidosAction( GestionarPedidoGtr  $gestionarPedidoGtr){...};
```

Figura 6:representación del patrón Controlador en la clase controladora*GestionarPedidoController.php*.

Fuente: elaboración propia.

- **Alta Cohesión**: una de las características principales del marco de trabajo Symfonyv4.3 es la organización del trabajo en cuanto a la estructura del proyecto. Realizando un diseño donde las clases mantengan una alta cohesión, permite que el software sea flexible a cambios sustanciales con efecto mínimo. Se emplea en las clases controladoras ya que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea cada clase se encargara de realizar solamente las funciones que estén en correspondencia con la responsabilidad que posea.En el módulo Pedido se evidencia este patrón en la figura 3, a través de la interrelación que existe entre las clases controladoras, las clases gestoras y las funcionalidades del sistema, la primera encargada de manejar la lógica de presentación y el flujo de los datos provenientes de la vista y la segunda encargada de manejar la lógica del negocio de cada funcionalidad.

- **Bajo acoplamiento:** este patrón se evidencia dentro del marco de trabajo, en las clases que implementan la lógica del negocio y de acceso a datos que se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista, lo que proporciona que la dependencia en este caso sea baja. Como existe poca dependencia entre esas clases se permite una mayor reutilización. En el módulo Pedido se evidencia este patrón a través del marco de trabajo Symfony en la figura 3, el cual favorece ampliamente el bajo acoplamiento de las clases en el sistema, ya que a cada clase se le asignan solamente las responsabilidades necesarias de manera que no dependan en gran medida de otras.

2.5.2 Patrones GoF

Los patrones GoF se describen como una forma indispensable de enfrentarse a la programación a raíz del libro "*Design Patterns—Elements of Reusable Software"* de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF*, gang of four*)(Guerrero, 2014).

Patrón *Factory Method*:en la implementación de la herramienta este patrón es utilizado cuando se realiza una petición al contenedor de servicios, en la que se le pasa por parámetros el servicio que tiene definido el repositorio que se desea instanciar; internamente el contenedor de servicios determina cuál es la entidad que corresponde a dicho repositorio y realiza una llamada a la clase.

Patrón Decorador:en la implementación de la herramienta este patrón se evidencia en el uso de una plantilla globalpara las vistas que decorará las demás páginas de la aplicación.

2.6 Diagrama de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema, muestra los elementos que lo forman (clases e interfaces) y las relaciones entre ellos (asociaciones). Las clases se representan mediante una caja subdividida en tres partes: en la superior se muestra el nombre, en el medio los atributos y en la inferior las operaciones o métodos. Las asociaciones entre dos clases se representan mediante una línea que las une y esta puede tener una serie de elementos gráficos que expresan características particulares de la asociación (Pressman, 2010).En el siguiente diagrama de clases del diseño se muestran las clases relacionadascon el RF 17. Adicionar pedido.

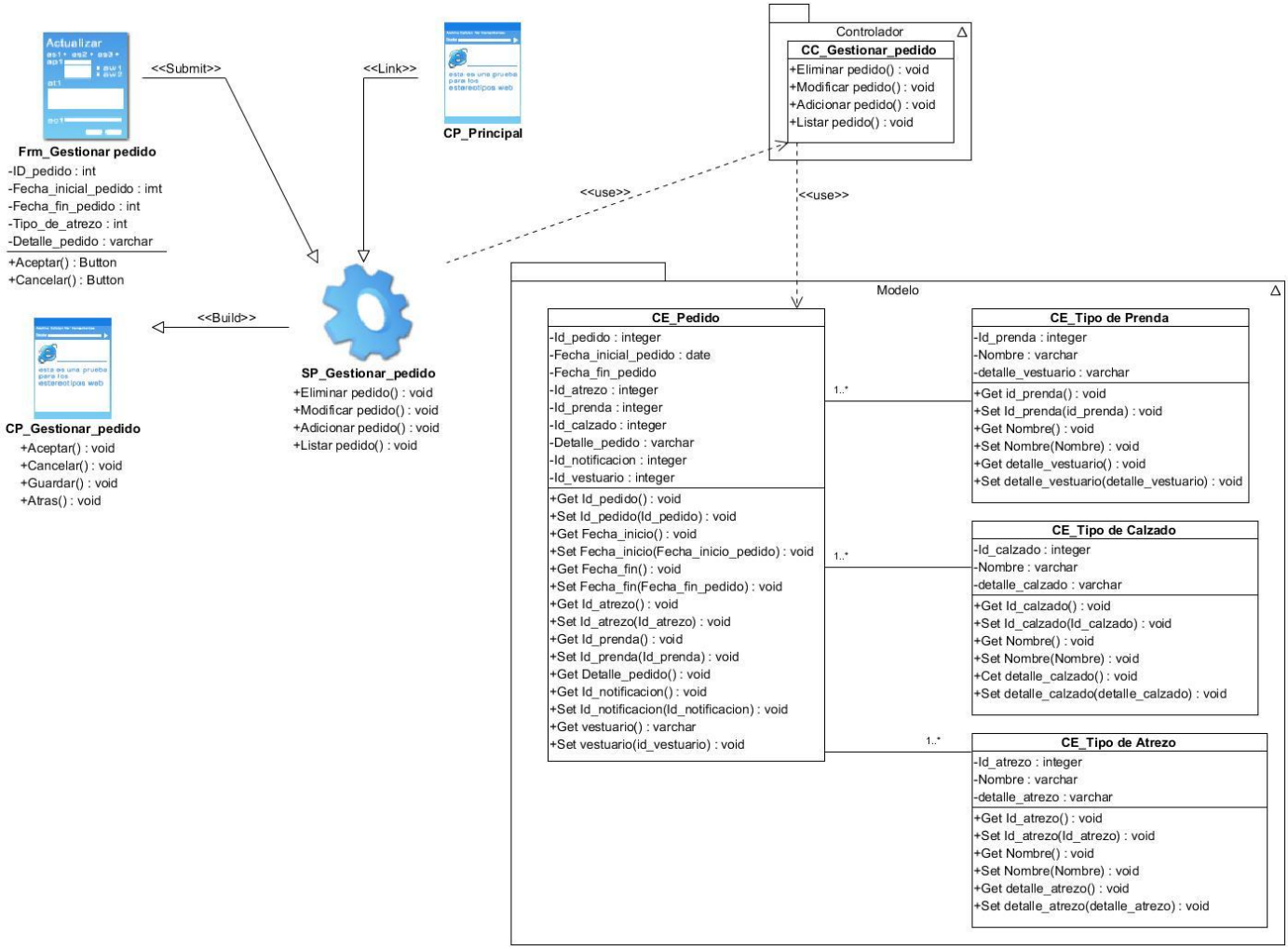


Figura 7. Diagrama de clases del diseño.

Fuente: elaboración propia.

En el diagrama de clases del diseño se evidencia cómo la vista interactúa con la controladora, esta a su vez manipula al modelo y se nutre de un servicio ejecutado medianteuna petición realizada a través del protocolo HTTP. Esta petición llega al controlador CC_Gestionar_pedido y es enviada a la vista. La información obtenida se muestra en la clase CP_Gestionar_pedido.

2.7 Modelo de datos

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente gestionada por el sistema. Su enfoque principal es apoyar y ayudar a los sistemas de información mostrando el formato y la definición de los diferentes datos involucrados. Define la correlación entre las

clases de diseño persistentes y las estructuras de datos persistentes(Tecnologías-Información, 2018). A continuación, se muestra el modelo de datos correspondiente a la solución propuesta.

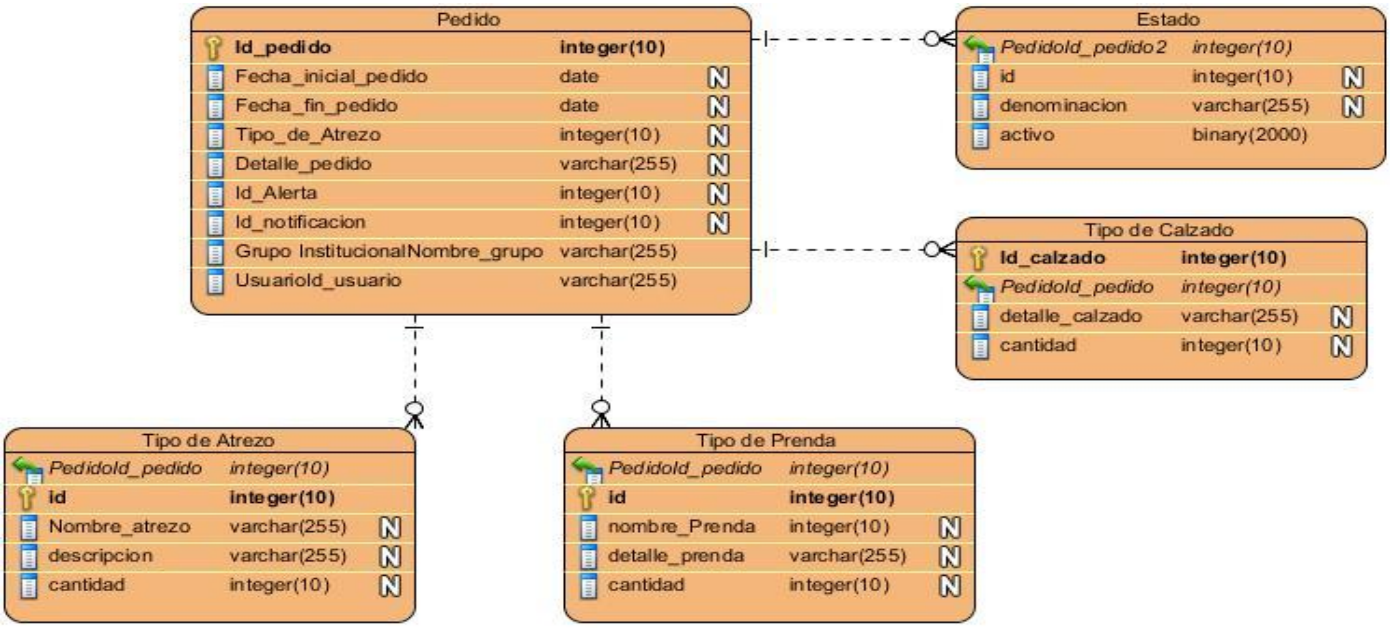


Figura 8. Modelo de datos para la propuesta de solución.

Fuente: elaboración propia.

En la figura 8 se representan las tablas asociadas a la presente solución. Se evidencia cómo las tablas Tipo de Atrezo, Tipo de Prenda, Tipo de Calzado y Estado se relacionan directamente con la tabla Pedido. En cada una se representan los datos que persisten.

2.8 Estándares de codificación

Unestándar de códigoes un conjunto de convenciones establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código. Estos, varían dependiendo del lenguaje de programación elegido y además varían en cobertura, algunos son más extensos que otros.

A continuación, se describen los estándares de codificación utilizados para la implementación de la propuesta de solución.

Identación

El contenido siempre se identará con *tabs*, nunca utilizando espacios en blanco.

Cabecera del archivo

Es importante que todos los archivos *.php inicien con una cabecera específica que indique información de la versión, autor de los últimos cambios, etc. Es decisión de cada equipo decidir si se quieren o no agregar más datos.

Comentarios en las funciones

Todas las funciones deben tener un comentario, antes de su declaración, explicando que hacen. Ningún programador debería tener que analizar el código de una función para conocer su utilidad, tanto el nombre como el comentario que acompañe a la función deben ser suficiente para entender el código.

Ubicación y denominación de archivos

Se ubicarán los archivos según las convenciones establecidas por Angular v6 y Symfony v4.3. Por ejemplo:

Clases

Las clases serán colocadas en un archivo.php aparte, donde sólo se colocará el código de la clase. El nombre del archivo será el mismo del de la clase. Las clases siguen las mismas reglas de las funciones, por tanto, debe colocarse un comentario antes de la declaración de la clase explicando su utilidad. Los nombres de las clases deben iniciar con letra mayúscula. Si un nombre de clase se comprende de más de una palabra, la primera letra de cada nueva palabra debe comenzar con letra mayúscula. No se permiten las letras mayúsculas sucesivas.

Siempre utilizar las etiquetas:<?php ?> para abrir un bloque de código. No utilizar el método de etiquetas cortas.

Estilo y reglas de escritura de código PHP.

- Nombres de variables

Los nombres deben ser descriptivos y concisos. No usar grandes frases ni pequeñas abreviaciones para las variables. Siempre es mejor saber qué hace una variable con solo conocer su nombre. Esto se aplica para los nombres de variables, funciones, argumentos de funciones y clases. Los nombres de las variables y de las funciones pueden iniciar con letra minúscula, pero si estas tienen más de una palabra, cada nueva palabra debe iniciar con letra mayúscula. Las constantes deben de escribirse siempre en mayúsculas y tanto estas como las variables globales deben de tener como prefijo el nombre de la clase a la que pertenecen. En la figura 4 se pueden evidenciar los nombres de las variables contenidas en la clase entidad *Pedido()*.

- Las definiciones de la función

Los nombres de lasfunciones pueden contener solo caracteres alfanuméricos y siempre deben empezar en letras minúsculas. Cuando el nombre de una función conste de más de una palabra, la primera letra de cada nueva palabra debe comenzar con mayúscula.

```
public function listarPedidosAction (GestionarPedidoGtr  $pedidoGtr)
{
    $pedidoUsuario = $pedidoGtr - >obtenerPedidoPorUsuario();
    return $this- >view($pedidoUsuario,  statusCode: StatusCodes:: HTTP_OK );
}
```

Figura 9: Definición de la función *listarPedidos()*.

Fuente: elaboración propia.

- Llamadas a funciones

Deben llamarse las funciones sin los espacios entre el nombre de la función, el paréntesis de la apertura, y el primer parámetro; los espacios entre las comas y cada parámetro, y ningún espacial entre el último parámetro, el paréntesis del cierre, y el punto y coma.

- Llaves

Las llaves de apertura irán al final de la sentencia que delimitan, y las de cierre alineadas con el inicio de la sentencia en una nueva línea. En la figura 9 se puede evidenciar el uso de las llaves en el método *listarPedidos()*.

Variables sin inicializar: si no se tiene control sobre el valor de una variable, se debe verificar que esté inicializada.

2.9 Conclusiones del capítulo

- El empleo de la metodología AUP en su variación para la UCI, permitió organizar el desarrollo de la solución y generar los productos de trabajo necesarios correspondientes a la disciplina de análisis y diseño a través de la descripción de la arquitectura del sistema.
- La arquitectura del sistema dividida en *frontend* y *backend* permite la ampliación de las funcionalidades de la solución y la reutilización de los módulos de la presente solución en aplicaciones futuras.
- Los patrones de diseño y los estándares de implementación definidos, permitieron establecer las pautas para la codificación, obteniéndose una solución con poca dependencia entre clases, flexible al mantenimiento y a la introducción de cambios.

CAPÍTULO III: EVALUACIÓN DE LA PROPUESTA DE SOLUCIÓN

En este capítulo se validan los requisitos con el objetivo de comprobar si estos describen la solución que se desea. Se evalúa el grado de calidad y fiabilidad de los resultados obtenidos a partir de la validación del diseño a través de las métricas: Tamaño Operacional de las Clases y Relación entre Clases. Además, se define la estrategia de pruebas de software para verificar y revelar la calidad de la solución propuesta.

3.1 Técnicas de validación de requisitos

Con el objetivo de ratificar que los requisitos del software obtenidos se ajustan a la propuesta de solución, se llevó a cabo un proceso de validación de los mismos, para el cual se emplearon las siguientes técnicas:

Revisiones de los requisitos: se realizaron revisiones a cada uno de los requisitos por parte del cliente. Las revisiones internas generaron un total de siete no conformidades (NC), las cuales fueron corregidas satisfactoriamente. Con el cliente se realizó la revisión de dichos requerimientos en la cual se añadieron detalles a cuatrorequisitos funcionales y se aprobaron finalmente los mismos.

Diseño de prototipos web:se mostró al cliente un modelo ejecutable que permitió tener una visión preliminar de cómo se veríanlos módulos en el sistema y a través de la interacción con estos se comprobó la satisfacción y aprobación del cliente hacia los prototipos diseñados.En los Anexos 2 y 3 se evidencian algunos de los prototipos obtenidos.

Generación de casos de prueba: como parte del proceso de validación de los requisitos funcionales fueron diseñados casos de pruebas, en total se definieron 20 diseños de casos de prueba, uno por cada requisito. En los Anexos 4 y 5 se muestran la descripción de las variables y el caso de prueba correspondiente al RF 9. Adicionar pedido.

3.2 Validación del diseño

Las métricas de software constituyen los elementos que permiten evaluar la calidad de una determinada característica o artefacto que se genere en un proyecto de software. Para la validación del diseño de la solución propuesta, se estudiaron las diferentes métricas de diseño y sus características.

Con el objetivo de medir el nivel de relaciones entre las clases del sistema y la complejidad de cada clase por separado se seleccionaron dos métricas que permiten realizar mediciones de este tipo: Relaciones entre clasesy Tamaño operacional de las clases. A continuación, se muestran los resultados de la aplicación de estas métricas.

3.2.1 Relación entre clases (RC)

Permite evaluar el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas de unidad necesarias para probar una clase teniendo en cuenta las relaciones existentes entre ellas(Pressman, 2010).Se evalúa a partir de los siguientes atributos de calidad:

- Acoplamiento: un aumento del RC implica un aumento del acoplamiento de la clase.
- Complejidad de Mantenimiento: un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- Reutilización: un aumento del RC implica una disminución en el grado de reutilización de la clase.
- Cantidad de Pruebas: un aumento del RC implica un aumento de la cantidad de pruebas necesarias para probar una clase.

Para obtener un nivel de relación entre clases, el valor obtenido al aplicar dicha métrica debe ser directamente proporcional al acoplamiento y a la complejidad de mantenimiento; además debe ser inversamente proporcional al nivel de reutilización del código. Para aplicar la métrica RC es necesario categorizar cada una de las clases según la cantidad de relaciones que esta contenga. A continuación, se muestra una tabla con las categorías para clasificar cada uno de los atributos de calidad anteriormente mencionados, así como el criterio de evaluación. En la misma se emplean la abreviatura Prom (promedio):

Tabla 4. Métrica RC. Categoría por atributos y criterio de evaluación.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	RC = 0
	Bajo	RC = 1
	Medio	RC = 2
	Alto	RC > 2
Complejidad de Mantenimiento	Baja	RC <= Prom
	Media	Prom<=RC<= 2* Prom
	Alto	RC >2* Prom
Reutilización	Baja	RC >2* Prom
	Media	Promedio< RC <= 2 *Promedio
	Alto	RC <= Promedio
Cantidad de pruebas	Baja	RC <= Promedio
	Media	Promedio<= RC <2*Promedio
	Alto	RC >= 2*Promedio

Fuente:(Pressman, 2010).

En la siguiente figura se representa la cantidad de clases por cantidad de relaciones de usos que las mismas poseen:

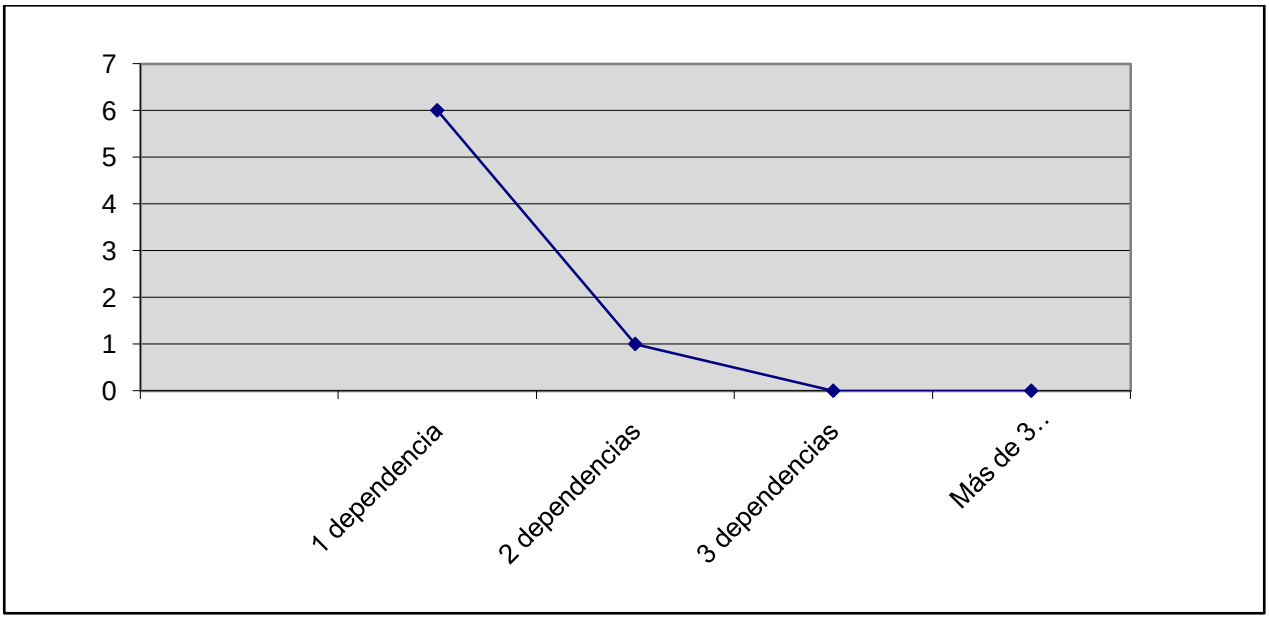


Figura 10. Representación de la cantidad de clases por cantidad de relaciones de usos que poseen.

Fuente: elaboración propia.

En la gráfica se observa como 6 clases presentan 1 sola dependencia, 1 clasepresenta 2 dependencias, no existe ninguna clase con 3 o más dependencias.

Después de haber aplicado la métrica RC, se tomaron los resultados obtenidos individualmente y se agruparon para ser analizados de manera general, promediando los valores obtenidos por categoría en cada atributo. A continuación, se representa en porciento (%)el nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases:

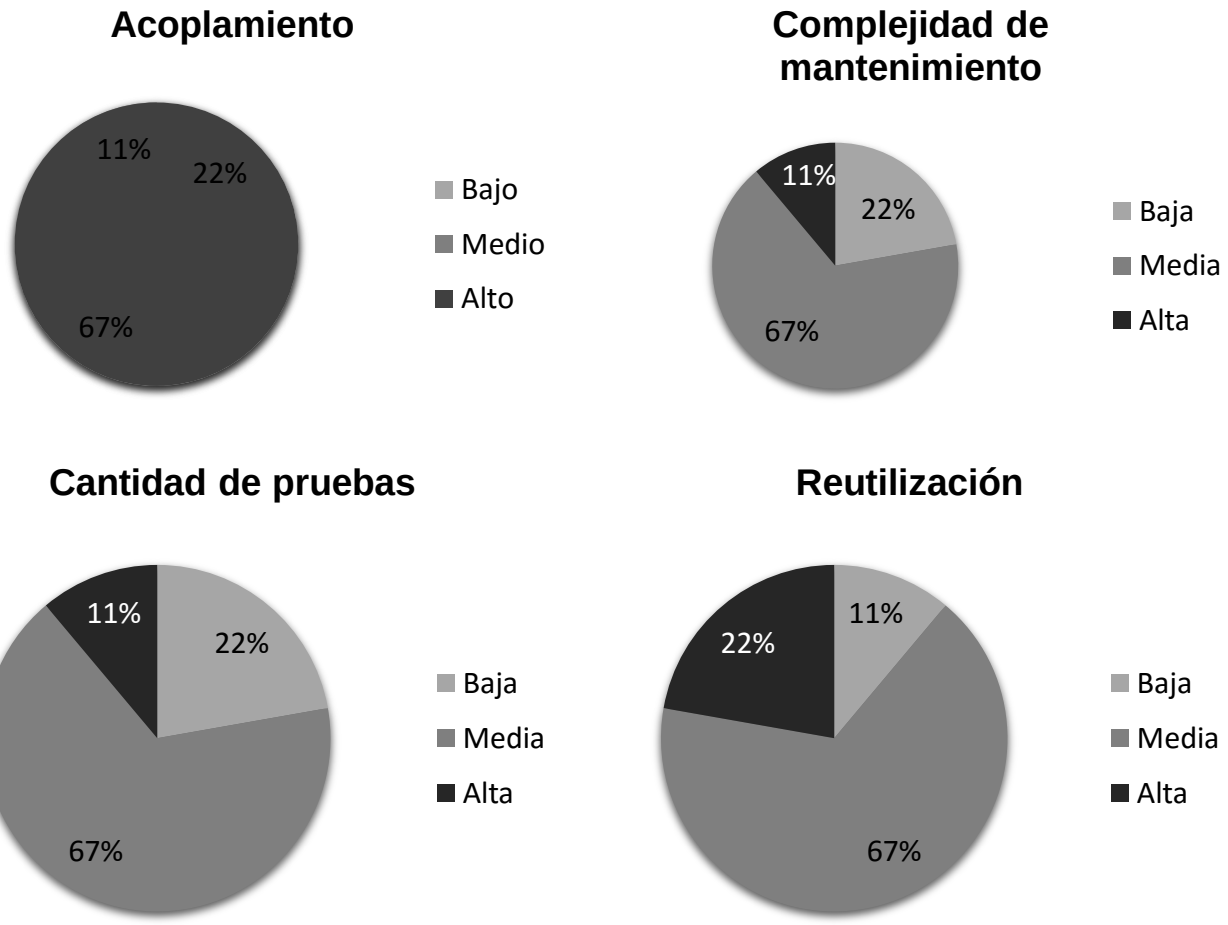


Figura 11. Representación en porciento (%) del nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases.

Fuente:elaboración propia.

Teniendo en cuenta que:

El umbral definido para validar el diseño: Bien= [0.1; 0.3], Regular= [0.4; 0.7], Mal= [0.8; 1].

Después de evaluar los resultados obtenidos de cada uno de los atributos de calidad que mide esta métrica, se obtuvieron los siguientes resultados:

- El 22% de las clases no promuevenacoplamiento.
- El 67% de las clases tienen una media complejidad de mantenimiento y un porciento medio de cantidad de pruebas.
- El 22% de las clases poseen una alta reutilización y el 67% reutilización media.

Se obtiene como resultado que las clases del diseño no promueven acoplamientopor lo que existe poca dependencia entre las clases, trayendo como consecuencia una alta probabilidad de reutilización. La complejidad de mantenimiento y la cantidad de pruebas son bajas, lo que minimiza el tiempo de implementación y pruebas de la solución propuesta.Según lo analizado anteriormente, los valores de RC se comportan de forma satisfactoria. Estos resultados expresan que las clases del diseño de la herramienta presentan bajo y medio acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas son bajas y en consecuencia el grado de reutilización es alto.Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable.

3.2.2 Métrica tamaño operacional de las clases (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad: la responsabilidad, la complejidad de implementación y la reutilización de las clases del diseño(desarrolloweb.com, 2019).

Una vez analizado el indicador tamaño de clase, si el valor resultante tiende al crecimiento, es probable que la clase posea un alto grado de Responsabilidad; en consecuencia, el nivel de Reutilización sería mínimo y la implementación altamente compleja. Para medir los atributos de calidad se definieron los umbrales que se muestran en la tabla 5. En la misma se emplean la abreviatura Prom (promedio):

Tabla 5. Criterios de evaluación para la métrica TOC.

Parámetros	Categoría	Criterios
Responsabilidad	Baja	< =Prom.
	Media	Entre Prom. y2* Prom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y2* Prom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y2* Prom.
	Alta	<= Prom.

Fuente:elaboración propia.

En la siguiente figura se representala cantidad de clases por cantidad de procedimientos que las mismas poseen:

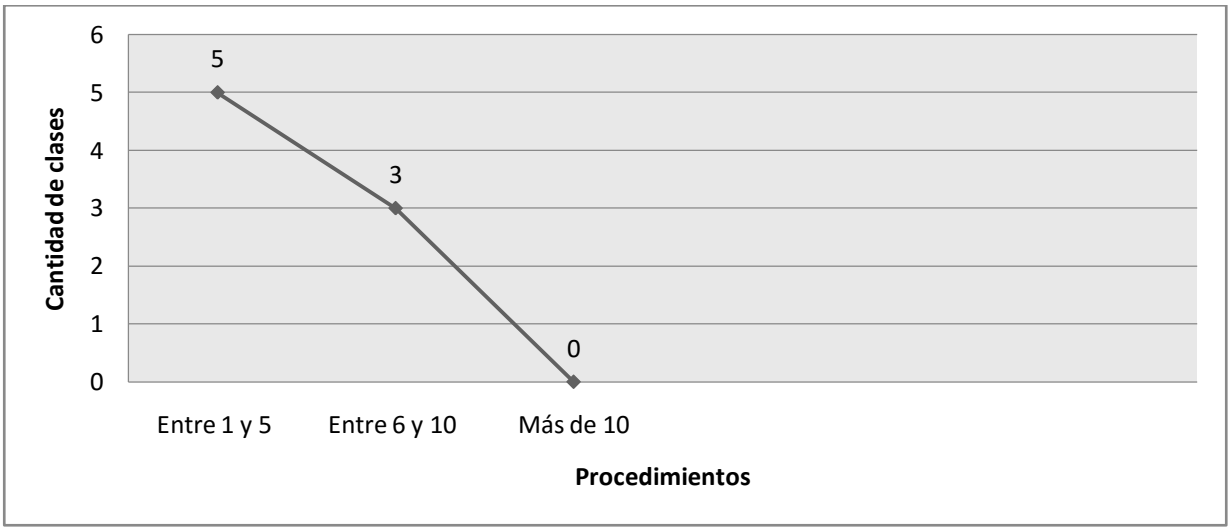


Figura 12. Representación de la cantidad de clases por cantidad de procedimientos que contienen.

Fuente: elaboración propia.

En la gráfica se observa que existen 5 clases que tienen entre 1 y 5 procedimientos, tres clases presentanentre 6 y 10 procedimientos y ninguna clase tiene más de 10procedimientos.

Al aplicar la métrica a las clases del modelo de diseño obtenido, y luego de estudiados los resultados,se representa en porciento (%)el nivel de responsabilidad, complejidad de implementación y reutilización de las clases al aplicar la métrica TOC, tal como se muestra en la siguiente figura:

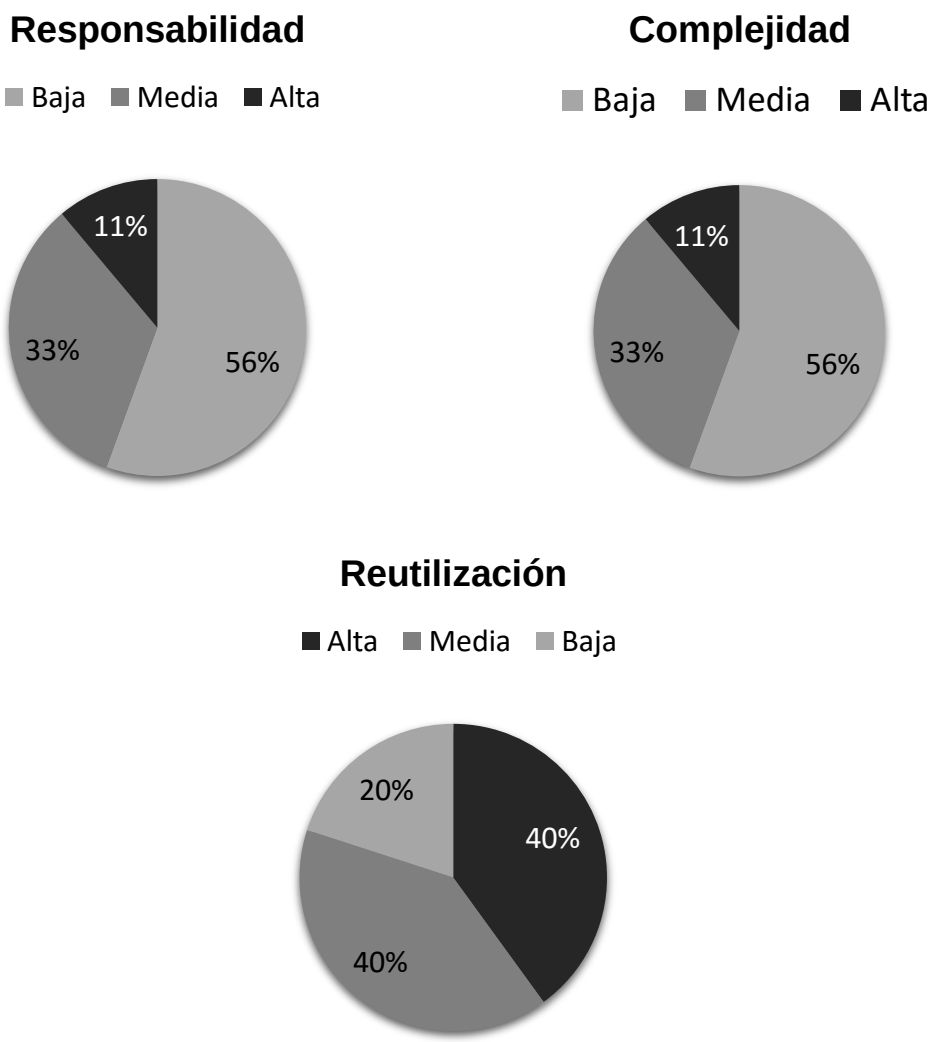


Figura 13. Representación en porciento (%) del nivel de responsabilidad, complejidad de implementación y reutilización de las clases.

Fuente: elaboración propia.

Después de evaluar los resultados obtenidos de cada uno de los atributos de calidad que mide esta métrica, se obtuvo lo siguiente:

- El 56% de las clases poseen una baja responsabilidad.
- El 56% de las clases poseen un bajo nivel de complejidad de implementación.
- El40% de las clases poseen una alta reutilización.

Con el análisis de estos resultados se observa que las clases del diseño no se encuentran sobrecargadas en cuanto a responsabilidades y el nivel de complejidad de las mismas es bajo, teniendo consigo una alta

reutilización.Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica TOC, se puede concluir que el diseño propuesto para el sistema no es complejo y presenta una calidad aceptable.

Luego de la aplicación de las métricas RC y TOC se arriba a la conclusión de que el diseño no es de alta complejidad, que las clases presentan un bajo nivelde acoplamiento y un alto grado de reutilización, lo que trae consigo que las clases puedan reutilizarse, lo que incide positivamente en la implementación.

3.3 Pruebas de software

Las pruebas y validaciones de software son de vital importancia para corroborar que la aplicación desarrollada no presente problemas de ejecución y se encuentre libre de errores, ya que durante el proceso de desarrollo de software es frecuente que los desarrolladores, al programar las diferentes funcionalidades, obvien algunas posibilidades que pueden variar el resultado esperado durante la ejecución del código; estas pruebas y validaciones constituyen la vía a través de la cual se asegura que el producto desarrollado está listo para ser entregado a los usuarios finales. Los errores más frecuentes pueden suceder a causa del mal uso de estructuras de datos, errores lógicos, entre otras(Pressman, 2010).

Teniendo en cuenta la metodología utilizada para el desarrollo de esta herramienta, se decide validar la propuesta de solución a través de las disciplinas pruebas internas, de liberación y aceptación. A continuación, se presenta la estrategia de pruebas a seguir para validar la calidad de los componentes de la presente solución:

Figura 14. Estrategia de pruebas.

Fuente: elaboración propia.

Conclusiones del capítulo

- La aplicación de técnicas de validación de requisitos permitió ratificar la correspondencia de los mismos con las solicitudes del cliente.
- Se realizó la validación del diseño mediante las métricas TOC y RC, lo que permitió definir de forma cuantitativa la calidad del mismo en el desarrollo de los módulos Pedido y Nomencladores.
- Se definió la estrategia de pruebas a tener en cuenta de acuerdo a la Disciplina de pruebas que propone la metodología AUP-UCI para garantizar la calidad final de los módulos que forman parte de la presente solución.

CONCLUSIONES GENERALES

- La elaboración del marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos de los sistemas de gestión de préstamos, facilitó la adquisición de los conocimientos necesarios para la solución propuesta.
- La especificación de los requisitos, así como el análisis y diseño de los módulos Pedido y Nomencladores, permitió obtener una aproximación para concebir los elementos necesarios de la implementación de los mismos.
- Los resultados obtenidos a través de las métricas de validación del diseño corroboraron de forma cuantitativa la calidad esperada de los artefactos obtenidos durante el desarrollo de la solución propuesta.

RECOMENDACIONES

Para la presente investigación se recomienda lo siguiente:

- La realización de las pruebas planificadas para garantizar la calidad final de los módulos propuestos.
- La integración delosmódulos Pedido y Nomencladores al XIGPREST para que sean utilizados desde el menú previsto en la aplicación según los permisos configurados para los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

Acosta, David L. 2017. Framework Design: A Role Modeling Approach. [En línea] 2017. http://sedici.unlp.edu.ar/handle/10915/19306.

Advance, RJ Code. 2019. Patrones de Arquitectura-Estilos (Cap3). *Patrones de Arquitectura-Estilos (Cap3).* [En línea] 2019. http://rjcodeadvance.com/Patrones de Arquitectura-Estilos (Cap3).

Alvarez, Miguel Angel. 2012. DesarrolloWeb.com. *DesarrolloWeb.com.* [En línea] 2012. https://desarrolloweb.com/manuales/tutorial-sql.html.

Ambler, Scott W. 2018. Algunos lineamientos útiles para trabajar y seleccionar herramientas CASE de manera ágil. *Agile Module.* [En línea] 2018. http://www.agilemodeling.com/essays/simpleTools.htm#SelectingCASE.

Boostrap Team. 2018. Bootstrap. *Bootstrap.* [En línea] 2018. https://getbootstrap.com/docs/4.3/getting-started/introduction/.

Bos, Bert. 2019. w3schools. *W3C.* [En línea] 2019. https://www.w3.org/Style/CSS/Overview.en.html.

Cervantes, Humberto. 2018. Software Guru. [En línea] 2018. http://sg.com.mx/revista/27/arquitectura-software.

Consejo, H. 2007.*Reglamento para el prestamo de computadores portatiles.* 2007.

desarrolloweb.com. 2019. desarrolloweb.com. *desarrolloweb.com.* [En línea] 2019. http://www.desarrolloweb.com/articulos-copyleft/articulo-metricas-de-software.html.

Doctrine Team. 2019. Doctrine 2 ORM. *Doctrine 2 ORM.* [En línea] 2019. https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/index.html.

Domínguez, Dorado M. 2018.*Todo Programación. Nº 13. Págs. 32-34.* s.l. : Editorial Iberprensa (Madrid), 2018.

Durante Lerate, Rosa María, y otros. 2009.*Sistemas para el Control de Versiones.* España : s.n., 2009.

Eguiluz, J. 2011.*Desarrollo web ágil con Symfony 2-Primera edición.* 2011.

Española, Real Academia. 2020. Real Academia Española. *Real Academia Española.* [En línea] 2020. [Citado el: 12 de Enero de 2020.] https://dle.rae.es/institucional.

Gaines, Jeff, Boyd, Geraldine y Copley, Della. 2017. Visual Paradigm Online. *Visual Paradigm Online.* [En línea] 2017. https://online.visual-paradigm.com/es/features/.

Giraldo, G., y otros. 2011. Revista Avances en Sistemas e Informática. [En línea] 2011. http://www.redalyc.org/articulo.oa?id=133122679013.. ISSN: 1657-7663..

Guerrero, Carlos A. 2014.*Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web.* 2014.

INAF. 2017. Proyección Estratégica. [En línea] 2017. http://www.inaf.co.cu.

Incorporated, Trend Micro. 2016. Manual del administrador de Hosted Email Segurity 2.0. *Manual del administrador de Hosted Email Segurity 2.0.* [En línea] 2016. https//:docs.trendmicro.com.

Lucid. 2018. lucidchart.com. *lucidchart.com.* [En línea] 2 de 12 de 2018. [Citado el: 5 de 12 de 2018.] https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml.

Martínez Pérez, Raúl y Rodríguez Esponda, Eddy. 2017.*Manual de metodología de la investigación científica.* 2017.

MDN. 2018. MDN web docs. [En línea] 2018. https://developer.mozilla.org/es/docs/Web/JavaScript.

Microsoft. 2018. Typescript. *Typescript.* [En línea] 2018. https://www.typescriptlang.org/.

Monné, Maylen Figueredo. 2004.*Sistema Integrado de Gestión Bibliotecaria: Módulo: Préstamos y otros servicios.* La Habana : s.n., 2004.

NetBeans. 2015. NetBeans. [En línea] 2015. https://netbeans.org/index_es.html.

Nicole. 2018. platzi.com. *platzi.com.* [En línea] 2018. https://platzi.com/blog/que-es-frontend-y-backend/.

Nieva, Francisco. 2015.*Tratado de escenografía.* Madrid : Editorial Fundamentos, 2015. ISBN 84-245-0850-5.

Ohmyroot. 2017. Estandares de codificación. *Estandares de codificación.* [En línea] 12 de enero de 2017. https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/.

ONERP. 2020. ONERP. *ONERP.* [En línea] Evolutive Advanced Software Solutions S.L., 2020. https://onerp.es/que-es-un-software-de-gestion/.

Page, Dave, Saito, Hiroshino y Yeatman, Mark. 2018. PgAdmin. *PgAdmin.* [En línea] 28 de Noviembre de 2018. http://www.pgadmin.org/.

Pérez Porto, Julian y Gardey, Ana. 2019. Definición. de. *Definición. de.* [En línea] 2019. [Citado el: 23 de Enero de 2019.] https://definicion.de/celeridad/.

Potencier, Fabien. 2019. Blackfire.io. *Blackfire.io.* [En línea] 30 de mayo de 2019. [Citado el: 9 de 1 de 2020.] https://symfony.com/blog/symfony-4.3-0-released.

Pressman, Roger S, Ph.D. 2010.*Ingenieria de Software. Un enfoque práctico. Septima Edición.* Mexico, D.F : The Me Graw Hill Companies, 2010.

—. 2010.*Ingeniería de Software un enfoque Práctico.* New York : s.n., 2010.

Pymes. 2018. QERP . *QERP .* [En línea] 2018. [Citado el: 3 de Diciembre de 2019.] https://www.qerpgestion.com/software-tienda-ropa/.

RAE. 2019. Diccionario de la lengua española. Real Academia Española. [En línea] 2019. https://dle.rae.es.

Raffino, María Estela. 2019. concepto.de. *concepto.de.* [En línea] 2019. https://concepto.de/servidor-web/#ixzz62zf4bDEu.

Ramos Salavert, Isidro y Lozano Pérez, María Dolore. 2011.*Ingeniería del software y bases de datos: tendencias actuales.* 2011. págs. Entornos de Desarrollo Integrados, pág. 78».

Robles, Victor. 2018. ¿Que es Git y para que sirve? [En línea] 2018. https://victorroblesweb.es/2018/04/28/que-es-git-y-para-que-sirve/.

Rodríguez, Ariel Nuviola. 2012.*Sistema para la gestión de préstamos de implementos deportivos.* La Habana : s.n., 2012.

—. 2012.*Sistema para la Gestión de préstamos de implementos deportivos en la Universidad de las Ciencias Informáticas.* La Habana : s.n., 2012.

Sanchez, Ruby. 2018. Especificación de Requisitos Software según el estándar de IEEE 830. [En línea] 2 de octubre de 2018. https://www.academia.edu/6647065/Especificaci%C3%B3n_de_Requisitos_Software_seg%C3%BAn_el_est%C3%A1ndar_de_IEEE_830.

Sánchez, Tamara Rodríguez. 2015.*Metodología de desarrollo para la Actividad productiva de la UCI. .* La Habana. Cuba : s.n., 2015.

Sánchez, Tamara Rodriguéz. 2014.*Metodologías ágiles en el desarrollo de software.* La Habana : s.n., 2014.

SBD. 2016. umh2820.edu.umh.es. *umh2820.edu.umh.es.* [En línea] 2016. http://umh2820.edu.umh.es/wp-content/uploads/sites/885/2016/02/Licencia-BSD.pdf.

SCRIBD Inc. 2020. SCRIBD. *SCRIBD.* [En línea] 2020. [Citado el: 27 de 11 de 16.] https://es.scribd.com/.

Solvingadhoc. 2017. Solvingadhoc. *Solvingadhoc.* [En línea] diciembre de 2017. https://solvingadhoc.com/las-historias-usuario-funcion-agilidad/.

Source Code Examples. 2019. Source Code Examples. *Source Code Examples.* [En línea] 2019. https://www.sourcecodeexamples.net/2018/06/information-expert-grasp-pattern.html..

Tecnologías-Información. 2018. Modelo de datos. *Modelo de datos.* [En línea] 2018. https://www.tecnologias-informacion.com/modeladodatos.html.

The jQuery Foundation. 2019. Jquery write less do more. *Jquery write less do more.* [En línea] 2019. https://blog.jquery.com/2018/01/19/jquery-3-3-0-a-fragrant-bouquet-of-deprecations-and-is-that-a-new-feature/.

The PHP Group. 2019. PHP. *PHP.* [En línea] 2019. https://www.php.net/docs.php.

The PostgreSQL Global Development Group. 2019. PostgreSQL. *PostgreSQL.* [En línea] 2019. https://www.postgresql.org/docs/10/release-10-1.html.

Usaola, Macario Polo. 2019.*Patrones GRASP.* 2019.

Wilson. 2018. AngularJS. [En línea] 2018. https://docs.angularjs.org/api.

WordReference.com. 2019. Diccionario Español en WordReference. [En línea] 2019. www.wordreference.com.

XEROX corporation. 2020. XEROX. [En línea] 2020. [Citado el: 31 de 1 de 2020.] https://www.xerox.es.

Zoega, Geir Valle. 2010.*Mapeador de objetos realacionales ORM.* 2010.

ANEXOS

ANEXO 1: encuesta realizada a los trabajadores del Departamento de Servicios en realizar los préstamos en la DEU.

Esta encuesta es considerada anónima y personal, dirigida a los trabajadores del Departamento de Servicio que realizan los préstamos en la DEU.

El objetivo de la encuesta es determinar el tiempo promedio que demoran habitualmente en realizar un préstamoa una persona,la celeridad es muy importante para el perfeccionamiento de nuestro trabajo.

Por favor complete la encuesta cuidadosamente al leerla primero, y luego señale sus respuestas con una "X".

1. ¿Qué tiempo demora en recopilar los datos de la persona que realiza el pedido
   ___ Menos de 1 minuto.
   ___ De 1 a 5 minutos.
   ___ De 5 a 10 minutos.
   ___Otro ¿Cuál? _____
2. ¿Qué tiempo demora al buscar un vestuario?
   ___ Menos de 20 minutos.
   ___ De 20 a 30 minutos.
   ___ De 30 minutos a 1 hora.
   ___ De 1 a 2 horas.
   ___ Otro ¿Cuál?_____

ANEXO 2: Prototipo de interfaz de usuario para el RF 4. Listar tipo de prenda.

ANEXO 3: Prototipo de interfaz de usuario para el RF 9. Adicionar tipo de atrezo.

ANEXO 4: Descripción de las variables del Caso de prueba 9. Adicionar pedido.

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	*Tipo de atrezo*	*Lista desplegable*	*No*	*NA*
2	*Cantidad disponible*	*Campo numérico*	*No*	*Campo inhabilitado que muestra la cantidad disponble de lo que se va a solicitar*
3	*Cantidad a solicitar*	*Campo numérico*	*No*	*Solo admite números*

ANEXO 5: Caso de prueba 9. Adicionar pedido.

Condiciones de ejecución

El usuario debe estar autenticado en el sistema

SC <Adicionar pedido>

Escenario	Descripción	Tipo de atrezo	Cantidad disponible	Cantidad a solicitar	Respuesta del sistema	Flujo central
EC 1.1 Adicionar pedido de forma correcta	El sistema permite adicionar un pedido a la lista de pedidos que solicita el usuario autenticado de forma correcta.	N/A	N/A	V 5	Se habilita el botón Solicitar y se añade el pedido a la Tabla de pedidos.	Menú lateral derecho/Solicitar pedido
EC 1.2 Adicionar pedido de forma incorrecta	El sistema no permite adicionar un pedido.	N/A	N/A	I T/9	No se habilita el botón Solicitar.	Menú lateral derecho/Solicitar pedido

[Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.]