

**Universidad de las Ciencias Informáticas**



**Facultad 2**

# **“Sistema de gestión de útiles y herramientas de la Facultad 2”**

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

Autor: Gilberto Remedios Arencibia

Tutores: Ing. Gusbey Pérez Carrazana

Ing. Osmar Capote Vázquez

La Habana, junio de 2019

“Año 61 de la Revolución”

**Declaración de Autoría**

Declaro ser el único autor del presente trabajo de diploma con título Sistema de gestión de útiles y herramientas de la Facultad 2. Reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales, con carácter exclusivo.

Para que así conste, firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2019.

---

Gilberto Remedios Arencibia

Autor

---

Ing. Gusbey Pérez Carrazana

Tutor

---

Ing. Osmar Capote Vázquez

Tutor

**Dedicatoria**

Este trabajo está dedicado a personas importantes que han estado presentes en mi vida y en especial a los que ya no lo están:

A mis padres, MARIA ELENA Y LUIS, quiero que sepan que los amo, para mi guerrera y por mi eterno guerrillero.

A mi tía MARTA y mi abuelo DIEGO, para mi niña y por mi viejo.

A mis hermanos GORDILLO, JOSE GABRIEL, LIA, BRIAN, LISANDRA, TONY, GÁLVEZ, MAIKOL y YOVALYS.

A toda la familia.

Gilberto Remedios Arencibia.

## **Agradecimientos**

Gracias a Gusbey porque sin duda alguna no habría terminado esta tesis sin su ayuda.

A Osmar y Cyndi por la guía y apoyo necesaria para que culminara este trabajo de diploma.

A mi madre por todo el apoyo, preocupación y confianza, por no dudar ni un solo momento en que volvería y terminaría lo que comencé, a mi padre por siempre avivar la aspiración de convertirme en un profesional.

A DARIEL, ASLEY, VICTOR, OSMAR y EDDRY, por su preocupación y apoyo incondicional en todo momento, este logro también es de ustedes.

A Ramón y Landy por las enseñanzas que me brindaron acerca de la programación.

A mis abuelos CARIDAD, ISOLINA y GILBERTO, que siempre me acompañan desde su eterna existencia.

A los miembros de mi otra familia ORLANDO, LUISA, IVÁN, ERNESTO y BRAYAN, muchas gracias por sus consejos y apoyo incondicional.

A KATY, EFRAÍN, VEGA Y YOSBEL por el cariño, confianza y enseñanzas que me dieron en esa etapa tan linda que compartimos, muchas gracias por siempre estar ahí para mí.

A mis tíos Ariel y Odalys por aguantar mis malcriadeces, a todos ellos, GRACIAS.

Gilberto Remedios Arencibia

**Resumen**

En el presente trabajo se presenta una solución para la gestión de información asociada al manejo de útiles y herramientas en el Vicedecanato de Administración (VA) de la Facultad 2 de la Universidad de las Ciencias Informáticas. En el área del VA aún existen flujos de datos manuales o seminformatizados, y se detectaron un grupo de necesidades específicas que ningún sistema implementa. Como principal limitación se detectó que ninguno de los sistemas analizados cumple con la Norma 40 del 2007 del Ministerio de Educación Superior (MES), que establece un formato para los diferentes modelos y anexos empleados en la gestión de información de útiles y herramientas. Atendiendo a lo anteriormente planteado, se desarrolló un sistema informático que realiza de forma centralizada el proceso de gestión de información referente al manejo de útiles y herramientas en el área del VA de la Facultad 2. Además, dicho sistema implementa lo establecido en la Norma 40 del 2007 del MES, ya que genera los modelos y anexos siguiendo el formato que establece la misma.

**Palabras clave:** gestión de información, Norma 40 del 2007 del MES, sistema de gestión, útiles y herramientas.

**Abstract**

In the present work we present a solution for information management associated with the management of tools and tools in the Vice Dean of Administration (VA) of the Faculty 2 of the University of Informatics Sciences. In the VA area there are still manual or semi-computerized data flows, and a group of specific needs was detected that no system implements. As a main limitation, it was found that none of the analyzed systems complies with the Standard 40 of 2007 of the Ministry of Higher Education (MES), which establishes a format for the different models and annexes used in the information management of tools and tools. In view of the foregoing, a computer system was developed that centrally performs the process of information management regarding the use of tools and tools in the VA area of Faculty 2. In addition, said system implements the provisions of Rule 40 of the MES 2007, since it generates the models and annexes following the format established by it.

**Key words:** information management, MES Standard 40 of 2007, management system, tools and tools.

**Índice**

Introducción.....	1
CAPÍTULO 1. Fundamentación teórica de los sistemas de información relacionados con la gestión de útiles y herramientas .....	6
1.1 Sistema de información del proceso de gestión de útiles y herramientas.....	6
1.1.1 Normativa del proceso de gestión de útiles y herramientas .....	7
1.2 Sistemas de información aplicados al proceso de gestión de útiles y herramientas .....	8
1.2.1 Sistemas internacionales.....	8
1.2.2 Versat-Sarasola.....	11
1.3 Selección de la metodología de desarrollo de software y las tecnologías a emplear en la solución .....	15
1.3.1 Metodologías de Desarrollo de Software (MDS).....	16
1.3.2 Herramientas y lenguajes informáticos .....	17
1.3.3 Sistema Gestor de Base de Datos.....	21
1.3.4 Marco de trabajo (Framework) .....	22
1.3.5 Entorno de Desarrollo Integrado (IDE).....	22
CAPÍTULO 2. Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2 .....	24
2.1 Procesos del vicedecanato de administración de la Facultad 2.....	24
2.2 Definición de requisitos .....	28
2.2.1 Requisitos funcionales.....	28
2.2.2 Requisitos no funcionales.....	29
2.3 Propuesta del sistema .....	30
2.4 Exploración de la propuesta de solución.....	30
2.4.1 Historias de usuario (HU) .....	31
2.5 Planificación.....	46
2.5.1 Estimación de esfuerzo .....	46

2.5.2 Plan de iteraciones .....	47
2.5.3 Plan de Entrega.....	48
CAPÍTULO 3. Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2 .....	49
3.1 Arquitectura del sistema.....	49
3.1.1 Arquitectura Cliente-Servidor.....	49
3.2 Patrones de arquitectónicos.....	49
3.2.1 Modelo Vista Plantilla .....	50
3.3 Patrones de diseño .....	51
3.3.1 Patrones de asignación de responsabilidades.....	51
3.4 Diseño del sistema.....	52
3.4.1 Modelo de datos.....	52
3.4.2 Tarjetas Clase Responsabilidad Colaboración (CRC) .....	53
3.5 Implementación .....	56
3.5.1 Tareas de ingeniería.....	56
3.6 Pruebas de validación del sistema.....	61
3.6.1 Pruebas unitarias .....	62
3.6.2 Pruebas de aceptación.....	62
3.6.3 Pruebas no funcionales .....	69
Conclusiones generales .....	75
Recomendaciones.....	76
Referencias bibliográficas .....	77
Bibliografía .....	80
Anexos.....	86

## Introducción

El proceso de informatización de la sociedad se define en Cuba como la utilización ordenada y masiva de las Tecnologías de Información y las Comunicaciones para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad.(Blanco Encinosa, 2002)

Entre los responsables de realizar la informatización del país está la Universidad de las Ciencias Informáticas (UCI). Esta institución, además de concertar compromisos de negocios con otras organizaciones, trabaja también en la informatización de sus procesos principales, estratégicos y de apoyo.

La Dirección General de Logística es la encargada de garantizar la adquisición, recepción y distribución de los recursos entre las áreas para apoyar los procesos de la universidad. Las áreas se organizan por centros de costo que a su vez se dividen en áreas de responsabilidad. Los recursos que se reciben son clasificados de acuerdo a su operatividad en el área quedando clasificados como medios básicos, útiles, herramientas o material gastable. Los procedimientos de gestión y documentación de los recursos se encuentran normados por la Norma 40 del 2007 “Sobre los procesos de registro y control de los útiles, herramientas, enseres menores y textos en las entidades subordinadas al Ministerio de Educación Superior (MES)”.

Para la gestión de información de los recursos de las diferentes áreas de la universidad, la dirección de informatización de la UCI desarrolló el Sistema de Gestión Administrativa XABAL-KAINOS. Este sistema está diseñado por módulos que se encargan de diferentes grupos de procesos relacionados con la gestión administrativa de recursos.

Entre las áreas receptoras de los recursos que distribuye la Dirección de Logística se encuentran las facultades de la universidad. Los Vicedecanatos de Administración (VA) de cada una de ellas son los encargados de la gestión de los recursos, así como de los procesos de aseguramiento no tecnológicos de la docencia y producción. Los vicedecanos de administración, como parte de sus funciones, se encargan de la gestión de información asociada a los útiles y herramientas. Se consideran útiles y herramientas a aquellos bienes de vida útil prolongada y de escaso valor que, sin perder sus cualidades al ponerse en uso se cargarán a gastos, siendo objeto de control por las diferentes áreas. (Ministerio de Educación Superior, 2007)

Para la realización del proceso de gestión de útiles y herramientas en las facultades, los responsables se apoyan en las herramientas del paquete ofimático. Tal es el caso de la generación y elaboración de los modelos asociados a la norma 40 del 2007 del MES, los cuales se realizan a través de ficheros ofimáticos con extensiones XLSX, XLS, DOCX, DOC o similares. Se trabaja de forma engorrosa ya que la información debe fluir por los documentos antes mencionados o por correo electrónico.



Al analizar las causas de la no utilización del sistema XABAL-KAINOS en la gestión de información de útiles y herramientas, se detectan un grupo de necesidades específicas que este no incluye:

1. Permitir el acceso de todo el personal autorizado a consultar datos e información, además de realizar otras funcionalidades según su rol: El sistema ha sido diseñado para que lo usen personas con determinados conocimientos y privilegios, y no todo el personal que necesita trabajar realmente con el sistema.
2. Guardar un registro de todas las acciones que se realizan durante la gestión de los recursos.
3. Garantizar un mayor seguimiento y control de los útiles y herramientas por cada una de las áreas que disponen de estos tipos de recursos.
4. Realizar un correcto manejo y control de los procesos asociados a los útiles y herramientas que se establecen en la Norma 40 del 2007 del Ministerio de Educación Superior (MES): Esta es la principal limitación del sistema en cuestión, ya que dicha norma establece un conjunto de buenas prácticas y modelos de gestión sobre los útiles y herramientas que todas las entidades adscritas al MES deben cumplir de forma obligatoria.
5. Generación de documentos siguiendo las regulaciones a nivel ministerial: XABAL-KAINOS no proporciona funcionalidades de generación para los modelos que establece la Norma 40 del 2007 del MES.

Esta forma de trabajar implica un mayor esfuerzo por parte de los encargados de la gestión de información asociada a los útiles y herramientas, aumentando el tiempo empleado en la generación de la documentación asociada a las actividades del proceso. La manera actual de ejecutar el proceso de gestión de útiles y herramientas no permite la posibilidad de conocer reportes e información general para la toma de decisiones a nivel de facultades y sus centros de costo.

No existen evidencias prácticas de un sistema que solucione el problema actual que presentan los vicedecanatos de administración de las facultades de la UCI y los trabajadores encargados de gestionar los útiles y herramientas en sus respectivas áreas. Para el desarrollo de la investigación se toma como muestra la Facultad 2, que en su estructura contiene tres centros de costo: Facultad 2, Centro de Informatización de la Gestión Documental (CIGED) y Centro de Telemática (TLM).

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema a resolver**: ¿Cómo mejorar el sistema de información relacionado con la gestión de los útiles y herramientas de la Facultad 2 para agilizar el proceso y organizar la documentación de acuerdo a las normativas vigentes?

Como **objeto de estudio** de esta investigación se tiene: sistema de información relacionado con la gestión de los útiles y herramientas en la facultad 2.

Como **objetivo general** se define: Desarrollar un sistema de gestión de información para el proceso de gestión de los útiles y herramientas de la Facultad 2 que cumpla con la Norma 40 del 2007 del MES y agilice su ejecución. Este objetivo se enmarca en el **campo de acción**: Sistemas de Gestión de Información del proceso de útiles y herramientas de la Facultad 2.

Para guiar el desarrollo de la presente investigación surgen las siguientes **preguntas científicas**:

- ¿Cuáles son las posiciones teóricas que fundamentan el proceso de gestión de útiles y herramientas y el desarrollo de sistemas para este proceso?
- ¿Cuáles son las características en cuanto al funcionamiento y comportamiento del sistema de gestión de información para el proceso de gestión de útiles y herramientas del Vicedecanato de Administración de la Facultad 2?
- ¿Qué resultados del proceso de validación evidencian que el sistema de gestión de información para el proceso de gestión de útiles y herramientas implementado funciona y cumple con la Norma 40 del 2007 del MES?

Con el propósito de dar cumplimiento a lo anteriormente planteado, se elaboraron las siguientes **tareas de investigación**:

1. Fundamentación teórica del sistema de información relacionado con la gestión de útiles y herramientas en la Facultad 2.
2. Análisis de sistemas homólogos para conocer aspectos regulares en cuanto a características y diseño de los sistemas de información aplicados al proceso de gestión de útiles y herramientas.
3. Selección de las tecnologías y metodología de desarrollo de software empleadas en el desarrollo de la aplicación.
4. Identificación de las principales funcionalidades del sistema que serán incluidas en la solución a partir del análisis del proceso de gestión de útiles y herramientas del Vicedecanato de Administración de la Facultad 2.
5. Implementación de un sistema de gestión de información de útiles y herramientas a partir del diseño y funcionalidades definidas.
6. Validación de la solución a partir del diseño, estrategia y aplicación de pruebas funcionales y no funcionales al sistema implementado.

**Métodos de Investigación Teóricos** (Hernández León y Coello González, 2008)

Permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de

las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis.

Análítico-Sintético (Hernández León y Coello González, 2008)

Análisis: permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio.

Síntesis: Establece mentalmente la unión entre las partes previamente analizadas, permite descubrir sus características generales y las relaciones esenciales entre ellas.

Aplicado para analizar la información obtenida en la investigación del proceso de gestión de útiles y herramientas en la Facultad 2. Su uso posibilitó la identificación de aspectos que pueden ser empleados en el desarrollo de la presente investigación. Además, para el análisis de la información disponible dentro y fuera del país, acerca de los sistemas de información aplicados al proceso de gestión de útiles y herramientas. Se identificaron mediante su empleo conceptos, definiciones, avances y otros elementos de utilidad para la investigación.

Inducción-Deducción (Hernández León y Coello González, 2008)

Inducción: Es una forma de razonamiento a través del cual se pasa de un conocimiento de cosas particulares a un conocimiento más general que va a reflejar lo que hay de común en esos fenómenos individuales.

Deducción: Es un procedimiento que permite, a partir de conocimientos generales, inferir casos particulares por un razonamiento lógico.

Utilizado en el análisis de la normativa del proceso de gestión de útiles y herramientas.

**Métodos de Investigación Empíricos** (Hernández León y Coello González, 2008)

Describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

La Entrevista: Es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación. (Hernández León y Coello González, 2008)

Aplicada al jefe de departamento de desarrollo del sistema XABAL-KAINOS perteneciente a la dirección de informatización. Con su aplicación se pudo constatar las condiciones actuales con las que se cuentan para la gestión de útiles y herramientas en la universidad. Ver Anexo 4.

Observación: La observación científica es la percepción planificada dirigida a un fin y relativamente prolongada de un hecho o fenómeno. Es el instrumento universal del científico, se realiza de forma consciente y orientada a un objetivo determinado. (Hernández León y Coello González, 2008)

Su puesta en práctica permite monitorizar de cerca el negocio, y analizar los resultados en el proceso de gestión de información de útiles y herramientas de la Facultad 2.

Modelado: Es la reproducción simplificada de la realidad que permite descubrir nuevas relaciones y cualidades del objeto. Este método permite la creación de modelos, (propuestas, alternativas, estrategias, etcétera.) (Alarcón, 2006)

En esta investigación, a través del método empírico, se realiza el modelado mediante diagramas, permitiendo reflejar la estructura, las relaciones y características de la solución propuesta. En este caso, con el uso del Modelado de Procesos de Negocio (BPMN) por sus siglas en inglés, se facilita también el diseño de clases necesario para la implementación de la aplicación.

La presente investigación se encuentra estructurada de la siguiente forma:

**CAPÍTULO 1. Fundamentación teórica de los sistemas de información relacionados con la gestión de útiles y herramientas:** se fundamenta teóricamente la gestión de útiles y herramientas a partir de los procesos y definiciones que establece la Norma 40 del 2007 del MES para la gestión de útiles y herramientas. Además, se analizan los sistemas que incluyen características y funcionalidades similares al sistema que se desea desarrollar. Se seleccionan y justifican las diferentes tecnologías, herramientas y Metodología de Desarrollo de Software (MDS) empleadas en el desarrollo e implementación del sistema.

**CAPÍTULO 2. Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2:** en este capítulo se obtienen los roles y funcionalidades que conformarán la propuesta de solución. Se definen las Historias de Usuario (HU) como parte de los artefactos definidos por la MDS empleada. La estimación de esfuerzo de las HU permitió elaborar el plan de entrega de las iteraciones del proyecto.

**CAPÍTULO 3. Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2:** se describe la arquitectura del sistema, los patrones utilizados en el diseño y el modelo de datos como punto de partida para la implementación del sistema. Se conforman las tarjetas CRC y las tareas de ingeniería asociadas a las HU. Además, se materializa la propuesta del sistema que da cumplimiento al objetivo trazado en la investigación. La validación de la aplicación se realiza a través de pruebas funcionales y no funcionales.

## **CAPÍTULO 1. Fundamentación teórica de los sistemas de información relacionados con la gestión de útiles y herramientas**

### **1.1 Sistema de información del proceso de gestión de útiles y herramientas**

El Decreto Ley número 281 del 2009 “Del Sistema de Información del Gobierno” define la información como un conjunto organizado de datos procesados que constituyen un mensaje sobre un determinado ente o fenómeno. Proporciona significado o sentido a las cosas y su uso racional es la base del conocimiento, facilitando la solución de problemas y la toma de decisiones.

Por su parte la gestión de información se define como un conjunto de principios, métodos y procedimientos destinados al tratamiento armonizado de la información para obtener resultados que satisfagan las necesidades de los usuarios y garantizar la obtención de la información relevante.

En cambio, un sistema de información lo constituyen un conjunto organizado de personas, procesos y recursos, incluyendo la información y sus tecnologías asociadas, que interactúan de forma dinámica, para satisfacer las necesidades informativas que posibilitan alcanzar los objetivos de una o varias organizaciones.

El proceso de gestión de información de útiles y herramientas en las áreas de la universidad se estructura de acuerdo a la jerarquía y dependencia que existe entre los diferentes centros de costo y áreas de responsabilidad material. Atendiendo a la afirmación realizada anteriormente se definen las siguientes definiciones:

Un centro de costo es una unidad de trabajo, o grupo de trabajo que ejecutan una serie de tareas que se distinguen por la uniformidad con que realiza una actividad perfectamente identificable, contando con un grupo de equipos y trabajadores en un área determinada de responsabilidad en la fabricación de un producto o prestación de servicios. (Delgado, 2014)

Un área de responsabilidad material, es un centro de actividades que desarrolla un conjunto de funciones en la división estructural de la empresa, al frente de la cual se encuentra un responsable que constituye la base informativa del centro de costo. (Delgado, 2014)

El sistema de información del proceso de gestión de útiles y herramientas se encuentra normado por la Norma 40 del 2007 “Sobre los procesos de registro y control de los útiles, herramientas, enseres menores y textos en las entidades subordinadas al Ministerio de Educación Superior”.

### **1.1.1 Normativa del proceso de gestión de útiles y herramientas**

La Norma 40 del 2007 del MES establece un conjunto de descripciones, procedimientos y documentos para el proceso de gestión de útiles y herramientas, a continuación, se relacionan los de obligatorio conocimiento para comprender el funcionamiento del sistema:

Los responsables de áreas responden por todos los útiles y herramientas de su respectiva área, el Modelo 4: “Acta de Responsabilidad Material” ver Anexo 1.1, registra la relación de útiles y herramientas pertenecientes a una determinada área de responsabilidad material. No obstante, si dentro de un área se produce la entrega de útiles a determinados trabajadores, por requerimientos laborales, estas entregas se efectuarán mediante el Modelo 2: “Acta de entrega” ver Anexo 1.2. Cada acta estará debidamente firmada por el usufructuario, siendo este registro el documento que lo responsabiliza ante el responsable de área por la custodia y conservación del medio.

Se considerarán en uso, aquellos útiles y herramientas solicitados al almacén, para ser sometidos al uso y explotación, se registran en el Modelo 1: “Útiles en uso” ver Anexo 1.7. Los útiles en uso serán objeto de verificación física, en los casos de detección de faltante, sobrantes o roturas, serán objeto de evaluación por una comisión, la que actuará como está previsto. Los Modelos 3: “Inventario en Uso de Utensilios, Herramientas y Enseres Menores” ver Anexo 1.8, después de determinadas las roturas, diferencias y movimientos entre áreas si los hubiese se conservarán como evidencias de los resultados de los chequeos físicos practicados.

Se definen como sobrantes aquellos útiles y herramientas que se identifican durante verificaciones físicas o actividad normal de los responsables de áreas y que no se encuentran incluidos en el registro de útiles en uso del área de responsabilidad material. Para efectuar el registro del útil y/o herramienta en la contabilidad del área, se establece el Anexo I: “Sobrantes” ver Anexo 1.3.

Existen dos tipos de movimientos: Los movimientos internos se realizan entre áreas de responsabilidad material pertenecientes a un mismo centro de costo. La realización de un movimiento interno genera el Anexo IV: “Movimiento interno” ver Anexo 1.4. Ejemplo de movimiento interno: útil y/o herramienta trasladado de un laboratorio a otro pertenecientes al centro de costo CIGED.

Los movimientos externos se realizan entre áreas de responsabilidad material pertenecientes a centros de costo diferentes. La realización de un movimiento externo genera el Anexo V: “Movimiento externo” ver Anexo 1.5. Ejemplo de movimiento externo: útil y/o herramienta trasladado desde un laboratorio perteneciente al centro de costo CIGED a otro laboratorio perteneciente al centro de costo TLM.

Las bajas por conceptos tales como: desgaste o deterioro imputable al uso negligente, accidente o sustracción, robo y extravío, deben ser evaluadas por una comisión creada al efecto y precisada en un Acta de baja ver Anexo 1.6, la cual será firmada por los integrantes de la comisión.

## **1.2 Sistemas de información aplicados al proceso de gestión de útiles y herramientas**

El análisis de los sistemas se realizó teniendo en cuenta que: Entre sus funcionalidades incluyeran la gestión de recursos, activos fijos o presentaran un módulo dedicado específicamente a la gestión de útiles y herramientas. Además, se presenta especial atención a un conjunto de características como: licencia, lenguaje de desarrollo, plataformas de funcionamiento, soporte de bases de datos, cumplimiento de la Norma 40 del 2007 del MES.

### **1.2.1 Sistemas internacionales**

#### **OpenERP**

Software desarrollado en Argentina, soporta múltiples monedas y múltiples entidades. Se caracteriza por ser un sistema con componentes separados en un esquema Cliente-Servidor. Es multiplataforma; funciona sobre GNU/Linux y Microsoft Windows. Implementado en el lenguaje Python, presenta como soporte para bases de datos PostgreSQL, liberado bajo la licencia de GPL<sup>1</sup>. OpenERP posee un módulo de gestión de inventario que, a diferencia de los sistemas comunes de manejo de inventario, solo necesita un formulario para todos los movimientos, en lugar de un formulario para cada acción: recibir mercancía, enviar mercancía, movimientos internos, recepción de mercancía, envío de mercancía de vuelta al proveedor, etcétera. (Moss, 2013)

Funciones básicas del módulo de gestión de inventarios de OpenERP: (OpenERP Spain, 2010)

- Planificación de almacén, inventario, trazabilidad.
- Gestión de inventarios por partida doble.
- Diferentes métodos de evaluación de inventario.

---

<sup>1</sup> GPL: Licencia Pública General, GPL por sus siglas en inglés, es una licencia de autor usada en el mundo del software libre y código abierto.

## Capítulo 1: Fundamentación teórica de los sistemas de información relacionados con la gestión de útiles y herramientas

- Sistemas de costos (costo estándar, costes de producción, etcétera.)

**Observaciones:** OpenERP, se describe a sí mismo como el sistema de planificación de recursos de código abierto más destacado y sencillo que existe hasta el momento. Es una de las aplicaciones que gestiona de manera eficiente la gestión de inventarios, pero que a pesar de sus ventajas como software multiplataforma y liberado bajo licencia de GPL, su módulo de gestión de inventarios está enfocado al manejo de almacenes y no al de útiles y herramientas, por lo que no se adapta a las condiciones reales de la situación problemática del presente trabajo. La figura 1.1 muestra una de las interfaces del sistema OpenERP.

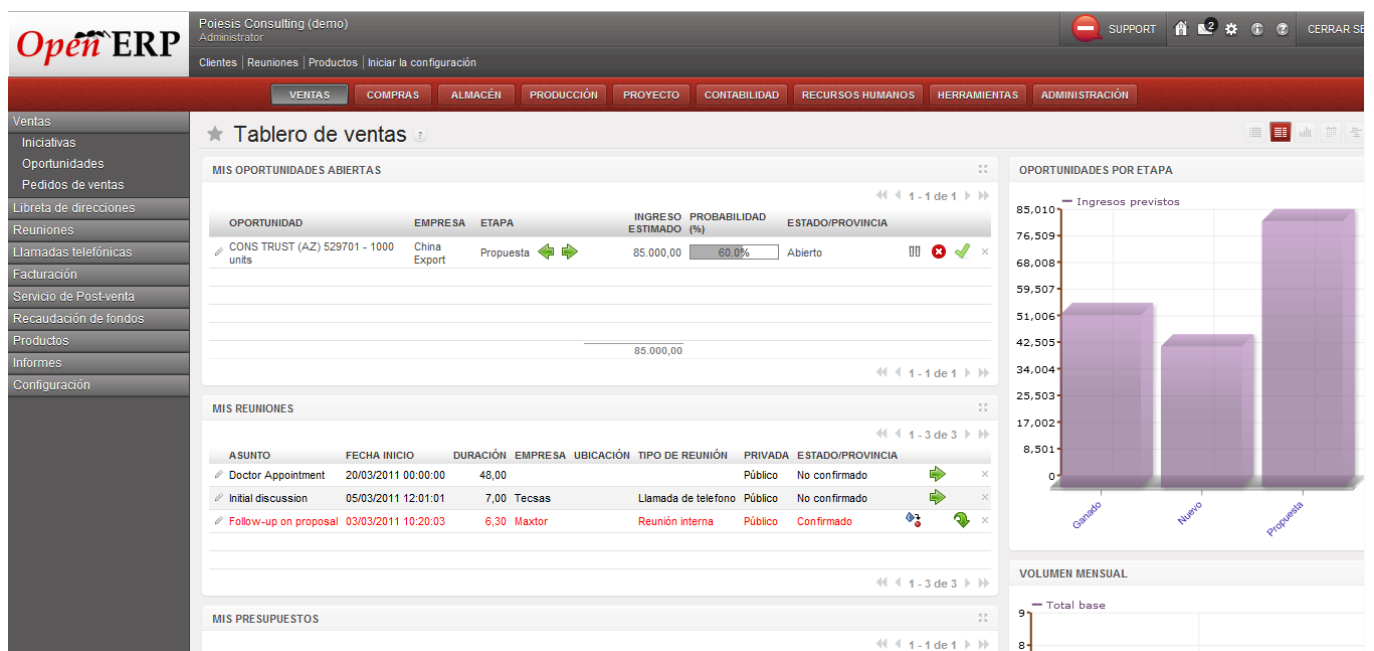


Figura 1.1 Interfaz de OpenERP. (Ramírez, 2010)

### SAP ERP (6.0)

Desarrollado por la compañía SAP (Sistemas, Aplicaciones y Productos). Combina un software de Planificación de Recursos Empresariales completo, flexible y escalable con una plataforma de tecnología abierta que puede aprovechar e integrar diversos sistemas, tanto de SAP como de terceros. (Enrich Cardona, 2013)

SAP ERP, implementa múltiples funcionalidades y cuenta con un módulo llamado SAP AM (Gestión de Activos). En este módulo se realiza el seguimiento y control de los activos de la organización, además de que controla y gestiona las compras y ventas de estos y realiza un análisis de la depreciación de los activos.



## Capítulo 1: Fundamentación teórica de los sistemas de información relacionados con la gestión de útiles y herramientas

Se caracteriza por ser un sistema totalmente preparado para trabajar mediante la web. Implementado en .NET, funciona sobre el sistema operativo Microsoft Windows y presenta soporte para bases de datos Oracle.

**Observaciones:** SAP, como aplicación informática da la posibilidad de ejecutar todos los procesos de negocios haciéndolos más ágiles; incluida la gestión de activos la cual ofrece un seguimiento y control sobre sus costos y la depreciación de los mismos. Inconvenientes fundamentales: Software privativo y no implementa lo establecido en la Norma 40 del 2007 del MES. La figura 1.2 muestra una de las interfaces del sistema SAP ERP.

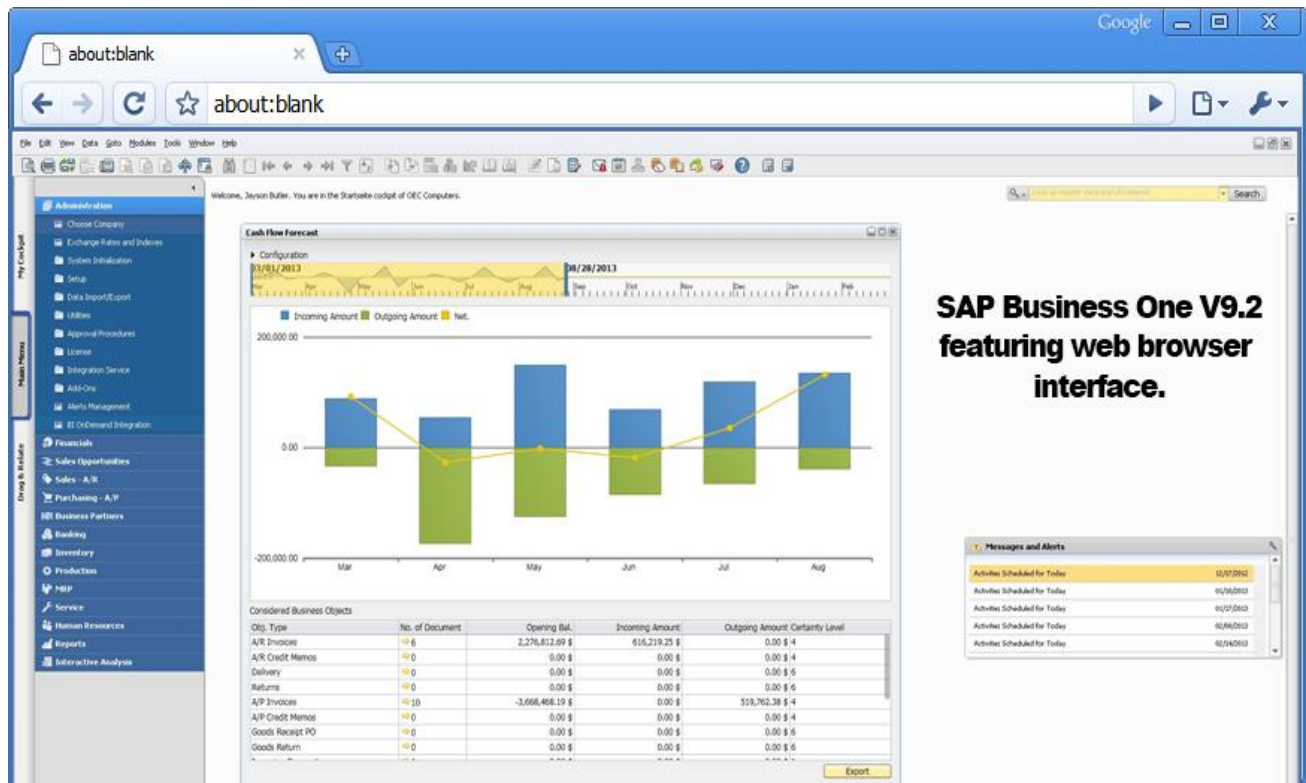


Figura 1.2: Interfaz de SAP ERP. (Remiro, 2018)

### ASSETS

Es un sistema comercializado bajo la firma panameña D'Marcos S.A. y distribuido en Cuba por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios del MES. ASSETS es un sistema de gestión integral de escritorio, estándar y parametrizado que se caracteriza por utilizar una arquitectura cliente-servidor, programado en Visual Basic 6.0 y Microsoft SQL Server 2000. La gestión de información en este sistema se enmarca sobre la gestión comercial y contable, permitiendo obtener los estados financieros, resúmenes de compras y ventas, etcétera a distintos niveles. (Amendola, 2011)

ASSETS permite controlar los útiles y herramientas de las entidades por centros de costos, áreas de responsabilidad material y empleados. También permite controlar los procesos de compra, altas, bajas y traslados integrándose toda esta información con los módulos de Contabilidad General e Inventario. (ASSETS: Sistema de Gestión Integral, 2019)

**Observaciones:** ASSETS como sistema de gestión integral posibilita un amplio manejo y control de todos los procesos asociados al entorno empresarial, incluyendo la gestión de útiles y herramientas, para la cual cuenta con un módulo específico y orientado a una completa integración con los demás procesos de la entidad. Inconvenientes principales: Software privativo y no implementa lo establecido en la Norma 40 del 2007 del MES. La figura 1.3 muestra una de las interfaces del sistema ASSETS.

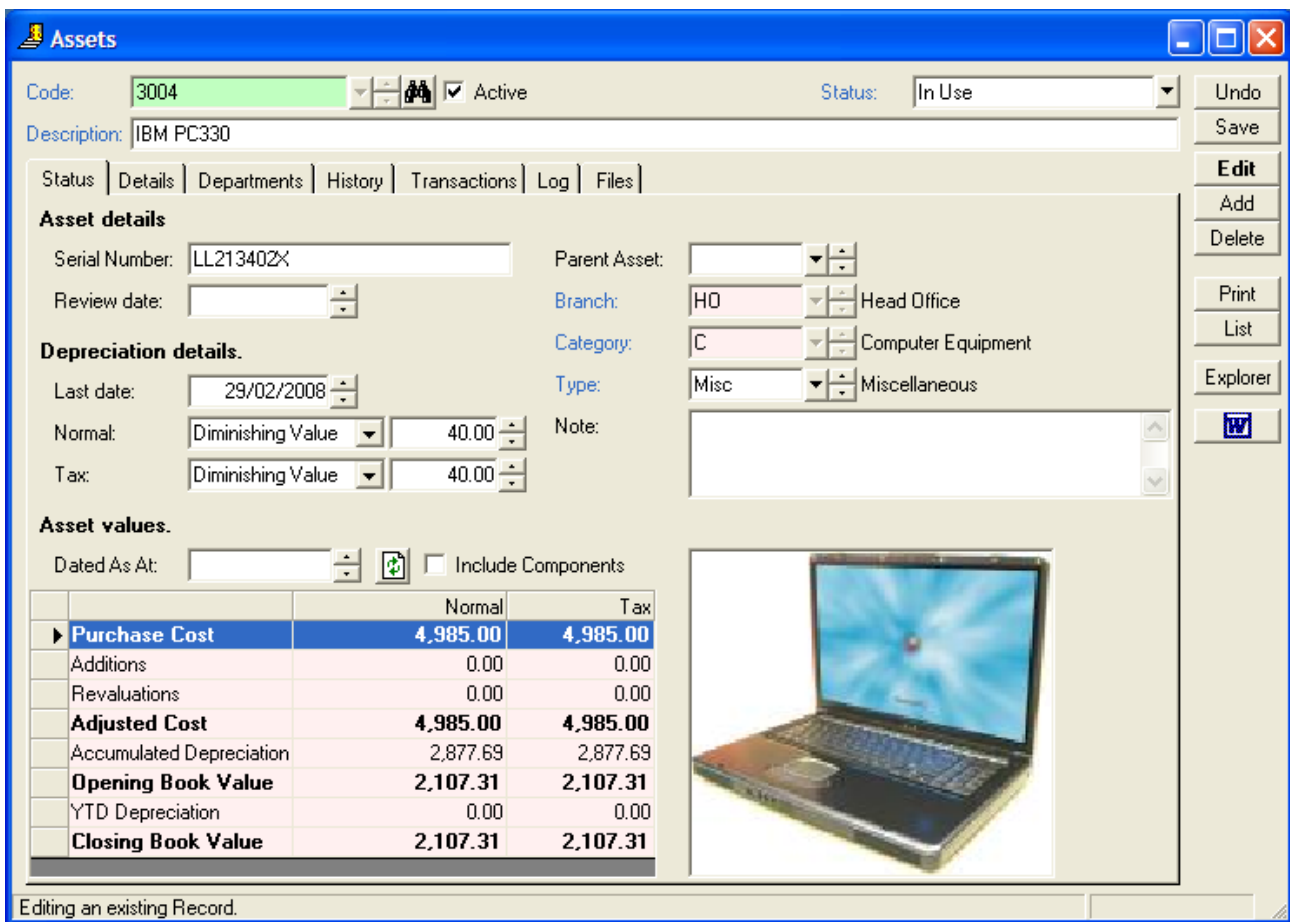


Figura 1.3: Interfaz de ASSETS. (D'Marco S.A., 2018)

### 1.2.2 Versat-Sarasola

Versat Sarasola, se caracteriza por ser una aplicación de escritorio. Concebido sobre una plataforma de trabajo Cliente-Servidor lo que permite además su instalación en red por las posibilidades que esta

tecnología facilita para el trabajo en un entorno multiusuario. Implementado en Delphi. Trabaja sobre el sistema operativo Microsoft Windows y presenta como soporte para bases de datos Microsoft SQL Server 2000. (Romero, 2009)

Permite llevar un registro contable de Activos Fijos (AF) e incluye actividades como: (Hernández Cárdenas, 2013)

- Está provisto de dos métodos para la depreciación de AF.
- Provee una vista donde es posible consultar la información básica de los AF de una empresa o unidad contable y su localización.
- Es posible realizar determinadas modificaciones sobre los AF a través de un árbol de áreas de responsabilidad y por cada una de ellas se desglosan todos los AF que le pertenecen.
- Permite hacer búsquedas de activos para conocer su ubicación.
- Es posible imprimir el submayor de un AF específico, sus modificaciones, así como los de un área específica o unidad.

**Observaciones:** Versat Sarasola es unas de las soluciones informáticas más utilizadas actualmente en el país. En sus últimas versiones tiene en cuenta la dualidad de monedas y la multientidad; define detalladamente todos y cada uno de los procesos que comprende la gestión de activos fijos en las entidades, de forma automática y eficiente. Inconveniente principal: No implementa la Norma 40 del MES del 2007. La figura 1.4 muestra una de las interfaces del sistema Versat Sarasola.

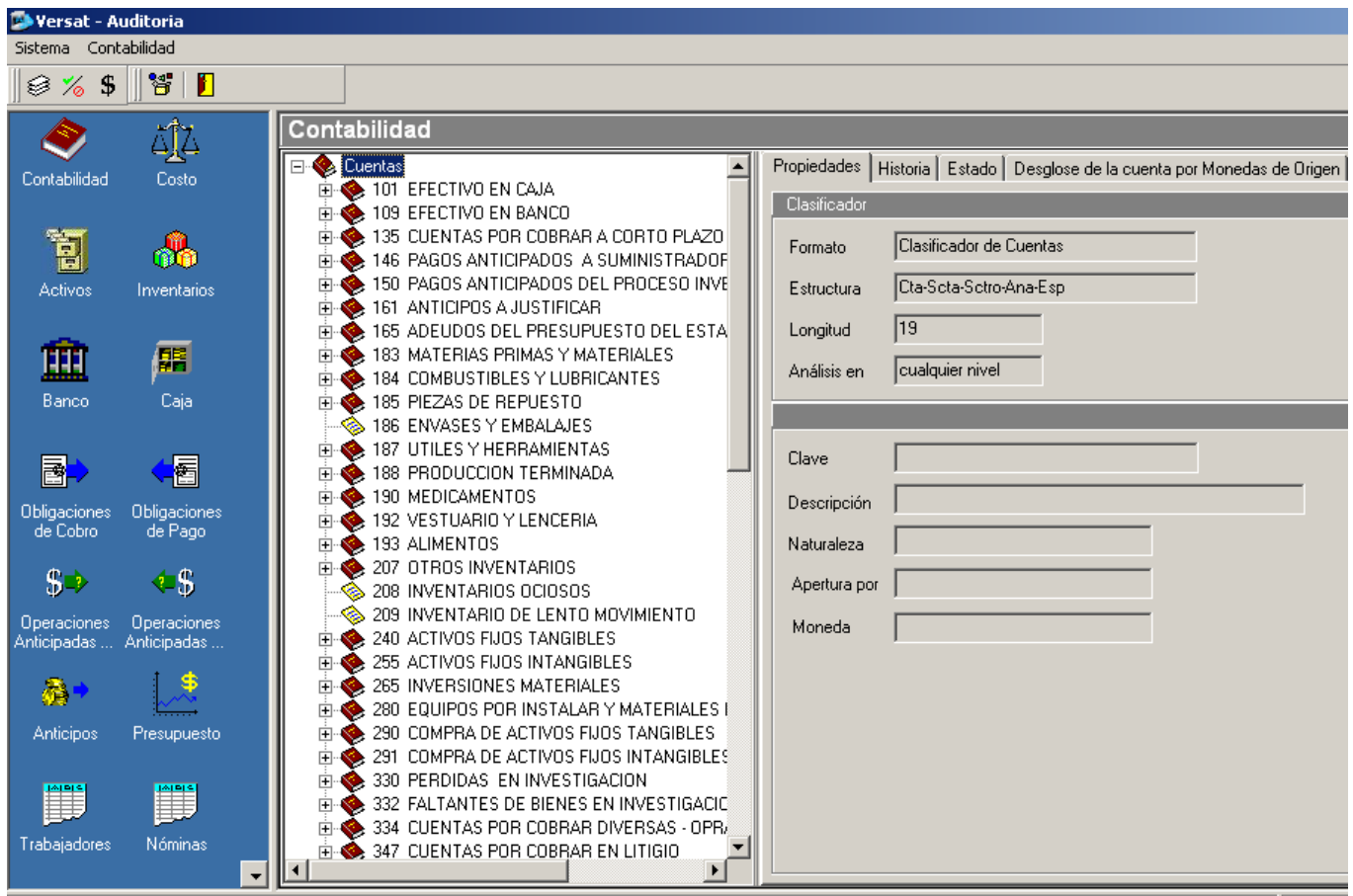


Figura 1.4: Interfaz de Versat Sarasola. (Romero, 2009)

## SISCONT5

SISCONT, es un software contable, multiempresa, doble moneda, que cuenta con módulos de tesorería, créditos y cobranzas, caja chica, unidades de negocio (centro de costos) y presupuestos. Siscont agiliza y facilita el ingreso de documentos, cuenta con asistentes de una sola pantalla y un motor de base de datos para recibir documentos en línea desde cualquier sistema. (Rodríguez Álvarez, 2018)

Permite definir al usuario hasta dos formatos de trabajo con sus activos fijos, visualizar las depreciaciones mensuales acumuladas y codificar los activos fijos. Este módulo no ofrece una gestión integral de los AF, solo es un módulo de extensión para los formatos contables. Presenta como soporte para bases de datos SQL Server 2000. Trabaja sobre el sistema operativo Windows. (Caneiro y Véliz, 2015)

**Observaciones:** Esta aplicación es bastante completa desde el punto de vista del control de la economía en las entidades, no siendo así en el control y gestión de los activos fijos, ya que el módulo que se encarga de estos no ofrece una gestión integral, sino que simplemente es una extensión para los formatos contables. Principales inconvenientes: Utilización de tecnologías privativas y no implementa lo

establecido en la Norma 40 del 2007 del MES. La figura 1.5 muestra una de las interfaces del sistema SISCONT.

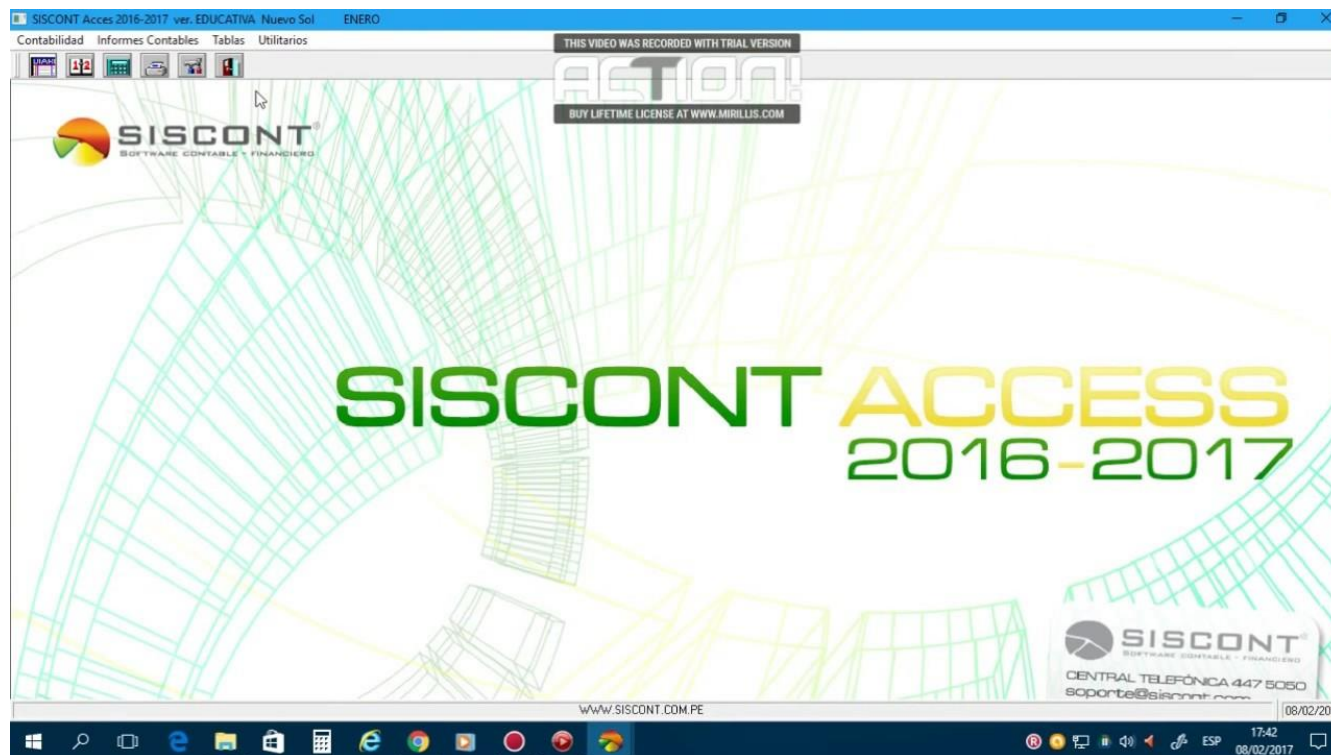


Figura 1.5: Interfaz de SISTCONT5. (Álvarez, 2018)

### **Sistema de Gestión Administrativa (XABAL-KAINOS)**

Actualmente existe y está disponible el Sistema de Gestión Administrativa (XABAL-KAINOS), desarrollado por la Dirección de Informatización de la UCI el cual fue desplegado con el objetivo principal de gestionar y controlar los medios materiales de las facultades, centros de desarrollo y demás áreas de la universidad. El módulo correspondiente a la gestión de útiles y herramientas de dicho sistema no se encuentra disponible, ya que la base de datos del sistema ASSETS que es por la cual XABAL-KAINOS actualiza sus datos se encuentra vacía. La inhabilitación del módulo de gestión de útiles y herramientas de dicho sistema se produjo por resolución del MES, que establece que la gestión de útiles y herramientas debe realizarse por áreas. (García Calderón, 2018)

**Observaciones:** XABAL-KAINOS es una herramienta que actualmente facilita la gestión de activos fijos y otros recursos en la UCI, aunque no se encuentra funcional el módulo de gestión de útiles y herramientas. En entrevista realizada al jefe del departamento de desarrollo del proyecto XABAL-KAINOS de la dirección de informatización, se comprobó que en la actualidad no existen perspectivas de solución para los problemas que presenta dicho software en su módulo de gestión útiles y herramientas, ya que

la gestión y el control de estos pasan a ser responsabilidad de cada una de las áreas de la universidad, como estableció el MES. Además, tampoco está disponible un proyecto para la actualización del software, con el objetivo de cumplir con las prácticas establecidas en la Norma 40 del 2007 del MES. La figura 1.6 muestra una de las interfaces del sistema XABAL-KAINOS.

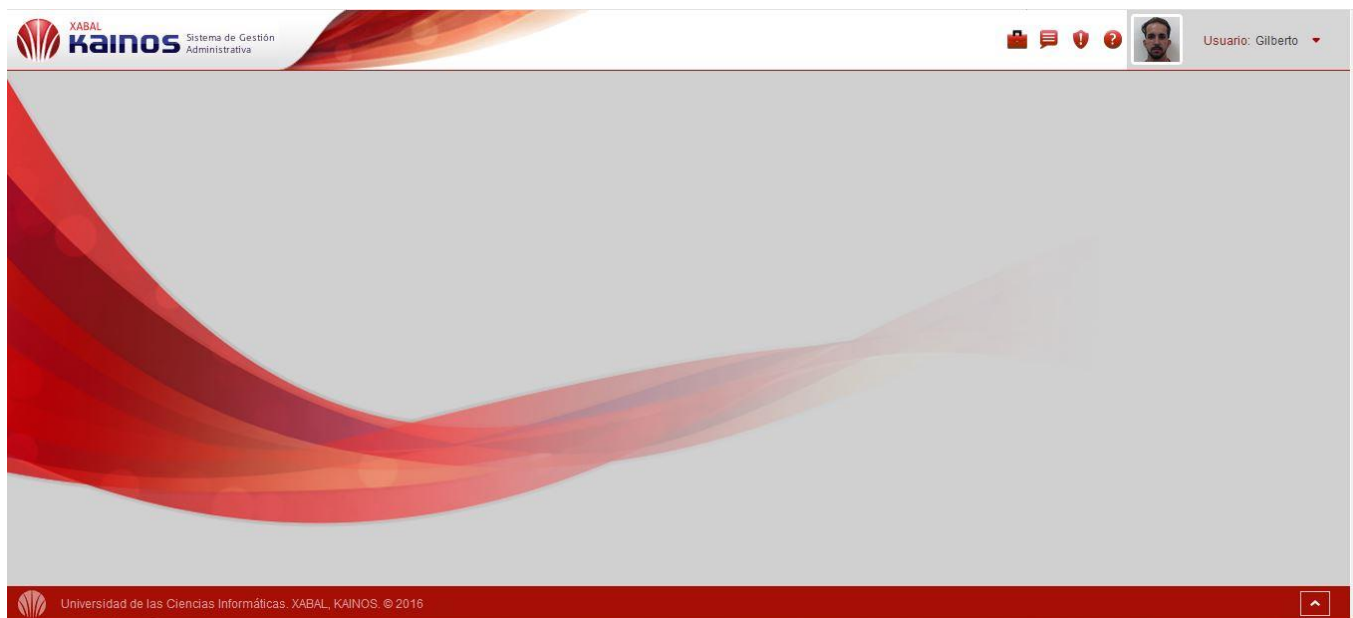


Figura 1.6: Interfaz de XABAL-KAINOS. (Universidad de las Ciencias Informáticas (UCI), 2019)

## **Conclusiones del análisis de los sistemas de información aplicados al proceso de gestión de útiles y herramientas**

A partir del estudio de los sistemas anteriores, se seleccionan las funcionalidades más comunes presentes en los módulos de gestión de activos fijos y/o útiles y herramientas para ser incluidas en la propuesta de solución. Los sistemas analizados presentan diferentes inconvenientes, ya sea por las licencias de sus productos, las herramientas sobre las que están soportados, o porque no son capaces de cumplir con los requisitos establecidos en la Norma 40 del 2007 del MES, por lo que se concluye no utilizar ninguno de ellos. A raíz de lo estipulado anteriormente se dispone implementar un sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2 y que cumpla con la Norma 40 del 2007 del MES y agilice su ejecución.

### **1.3 Selección de la metodología de desarrollo de software y las tecnologías a emplear en la solución**

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos

aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán.

### **1.3.1 Metodologías de Desarrollo de Software (MDS)**

Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. (Letelier y Penadés, 2012)

La presente investigación es un proyecto pequeño, de poca duración, se tiene un equipo de desarrollo pequeño, y el cliente es parte de la investigación. Por tales motivos el equipo de desarrollo se inclina al uso de una metodología de desarrollo de software ágil. Dentro de este tipo de metodología de desarrollo de software se analizan:

**SCRUM:** Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Letelier, 2006)

**Crystal Methodologies:** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. (Letelier, 2006)

**Programación Extrema (Extreme Programming, XP):** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. (Letelier, 2006)

Entre las metodologías ágiles de desarrollo de software **se selecciona la metodología XP** ya que:



- Es el más destacado de los procesos ágiles de desarrollo de software.
- Es considerada una adopción de las mejores metodologías de desarrollo de software.
- Se define especialmente adecuada para los proyectos con requisitos imprecisos y muy cambiantes.
- Se basa en realimentación continua entre el cliente y el equipo de desarrollo.
- Consta de una alta gama de bibliografía. Y los artefactos que genera presentan una descripción del comportamiento del sistema fácil de entender.

### **1.3.2 Herramientas y lenguajes informáticos**

#### **Lenguaje de Marcado de Hipertexto (HTML) 5**

HTML es un lenguaje informático de composición de documentos que cuenta con instrucciones especiales que son interpretadas por el navegador y como resultado se muestran textos, imágenes, vídeos y enlaces. Utilizando HTML se puede definir la apariencia de las páginas creadas de diferentes formas. (Van Lancker, 2012)

Es un estándar reconocido en todo el mundo, multiplataforma, soportado por muchos navegadores. Con él, se puede obtener un producto atractivo y rápido sin mucho esfuerzo por parte del desarrollador. Su diseño asegura una compatibilidad hacia atrás con todo lo desarrollado hasta el presente, en términos de publicación sobre la Web. Además, HTML5 mantiene el principio de separación entre el contenido y la presentación e incluso se ve reforzado. Las hojas de estilo CSS son elementales e inseparables de HTML5 y unido a la mayor potencia que proporcionan las nuevas hojas de estilo CSS3 aportan un importante marco de trabajo para el manejo, diseño y visualización de las interfaces de usuario en las aplicaciones web. (Auquilla y José, 2018)

Se decide utilizar HTML5 ya que éste presenta mejoras con respecto a las anteriores versiones de HTML, en cuanto a la definición de nuevas etiquetas que eliminan los procesos de validaciones y otras funcionalidades a través de su integración con otros lenguajes. Incluye nuevas etiquetas para dar estructura a un documento HTML, abandonando el uso de tablas y etiquetas div para la maquetación de las páginas. En esta versión, los formularios son mejorados con una redefinición de nuevos tipos de elementos para la etiqueta input, y han sido incorporados al lenguaje nuevos atributos para los elementos de formulario.

#### **Hojas de Estilo en Cascada (CSS3)**

Hojas de estilo en cascada (CSS por sus siglas en inglés) es un lenguaje informático de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y



XHTML que separa los contenidos y su presentación. En la actualidad se utiliza el lenguaje informático CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, así como la posición de cada elemento dentro de la página. (Remon, 2014)

A continuación, se relacionan las principales ventajas que hacen que CSS3 se presente como el lenguaje más usado en la presentación de páginas web:

- Es gratuito: Al igual que HTML no se necesita de ningún software costoso para codificar en CSS.
- Unificación del diseño: Esta es probablemente la principal ventaja de CSS, pues permite vincular a un solo archivo CSS a diversas páginas.
- Optimización del tiempo de carga y el tráfico del servidor: No sólo mejora la productividad, sino que ayuda a mejorar el tiempo de respuesta de las aplicaciones web.
- Flexibilidad: Proporciona una mayor experiencia de usuario según las características de sus dispositivos.

Atendiendo a las características y ventajas descritas anteriormente se escoge CSS3 para la aplicación de estilos a los contenidos que gestiona la aplicación web que se pretende desarrollar en la presente investigación.

### **JavaScript**

Es un lenguaje informático interpretado que permite incluir macros en páginas Web. Estos macros se ejecutan en el ordenador del visitante de las páginas, y no en el servidor. Aspecto interesante porque los servidores Web suelen estar sobrecargados, mientras que los clientes no.

Se selecciona JavaScript para el desarrollo de la presente solución informática porque proporciona los mecanismos para:

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas y simples.
- Evitar depender del servidor web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Optimización de los tiempos de carga y el tráfico del servidor.
- Comprobar los datos que el usuario introduce en determinados formularios antes de enviarlos.
- Comunicarse con el usuario mediante diversos métodos.

El lenguaje JavaScript se inserta en documentos HTML, de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa. Se trata de un lenguaje interpretado puro y sensible a mayúsculas, aunque algunas implementaciones ignoran en parte esta última característica. (Maza, 2012)

### **Python (2.7)**

Python es un lenguaje de programación de propósito general y de alto nivel, con características especiales que hacen de él una herramienta muy poderosa para el desarrollo de aplicaciones web, siendo usado comúnmente como lenguaje de programación del lado del servidor. Python garantiza un desarrollo mucho más ágil y dinámico, pues no es necesario compilar y enlazar el código cada vez que se desean probar los cambios que se han realizado. (Ríos et al., 2016)

#### **Principales Características de Python:**

- **Potencia la productividad de los desarrolladores:** En Python con una cantidad reducida de código se pueden implementar funcionalidades que, con otros lenguajes como Java o C se necesitaría una mayor cantidad de código.
- **Multiplataforma:** Soportado en plataformas GNU/Linux, Windows, Mac OS, Android y otros.
- **Fácilmente Extensible:** Python es un lenguaje extensible a partir del empleo de librerías ordenadas coherentemente en paquetes y módulos que llevan a la máxima expresión los principios de reusabilidad y de no repetición del código.
- **Open Source con Desarrollo Comunitario:** Python está licenciado con una licencia Open Source o de Código Abierto, su desarrollo es muy dinámico y está soportado por una amplia comunidad que lo respalda.

Atendiendo a las características antes mencionadas y las bondades que ofrece Python 2.7 con mejoras considerables en el manejo de cadenas de texto y tratamiento de excepciones en comparación con sus versiones anteriores. El uso de Python unido al servicio web NGINX y la integración y flexibilidad de Django como framework de desarrollo basado en Python aportan un valor agregado a la propuesta de solución de la presente investigación.

#### **Herramienta CASE (Visual Paradigm 8)**

Es una herramienta informática profesional multiplataforma, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una construcción más rápida de aplicaciones con mayor calidad y a un menor coste. Permite dibujar

todos los tipos de diagramas de clases, código inverso y generar código desde sus diagramas. (Génova, Fuentes y Valiente, 2006)

Visual Paradigm para BPMN posibilita un ahorro considerable de tiempo y una calidad óptima en el modelado de procesos del negocio. Se considera muy completa y fácil de usar, con soporte multiplataforma y excelentes facilidades de interoperabilidad con otras aplicaciones. (Molina y Moreno, 2010)

Atendiendo a lo expuesto anteriormente, se decide utilizar esta herramienta informática para el modelado de procesos del negocio.

### **Notación para el Modelado de Procesos de Negocio (BPMN)**

Notación para el Modelado de Procesos de Negocio (BPMN) por sus siglas en inglés, es un estándar mundialmente reconocido, aceptado y aplicado en el mercado, para la diagramación y especificación de procesos de negocio, desde la modelación de procesos conceptuales y lógicos, hasta diseños de procesos. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. (White y Miers, 2009)

Los elementos utilizados para construir los modelos BPMN fueron elegidos para ser distinguibles unos de otros y utilizar las figuras que son familiares a la mayoría de los diseñadores. Por ejemplo, las actividades se representan mediante rectángulos y las decisiones mediante rombos. (Ventura, 2013)

Por las características antes expuestas se decide escoger BPMN como notación para el modelado de esta investigación.

### **Servidor Web (NGINX)**

Es un software de código abierto, originalmente desarrollado por Igor Sysoev, un ingeniero informático kazajo, para soportar el alto tráfico del segundo sitio web más visitado de Rusia, Rambler.ru. (Taday y Alexandra, 2018)

Su gran eficiencia, ha propiciado que numerosas empresas decidieran migrar sus sistemas a NGINX, convirtiéndolo según un estudio de Netcraft en febrero de 2015 como se muestra en la figura 1.7 en el segundo servidor web con mayor cuota de mercado para los sitios web más ocupados del mundo. (Molina Coronado, 2015)

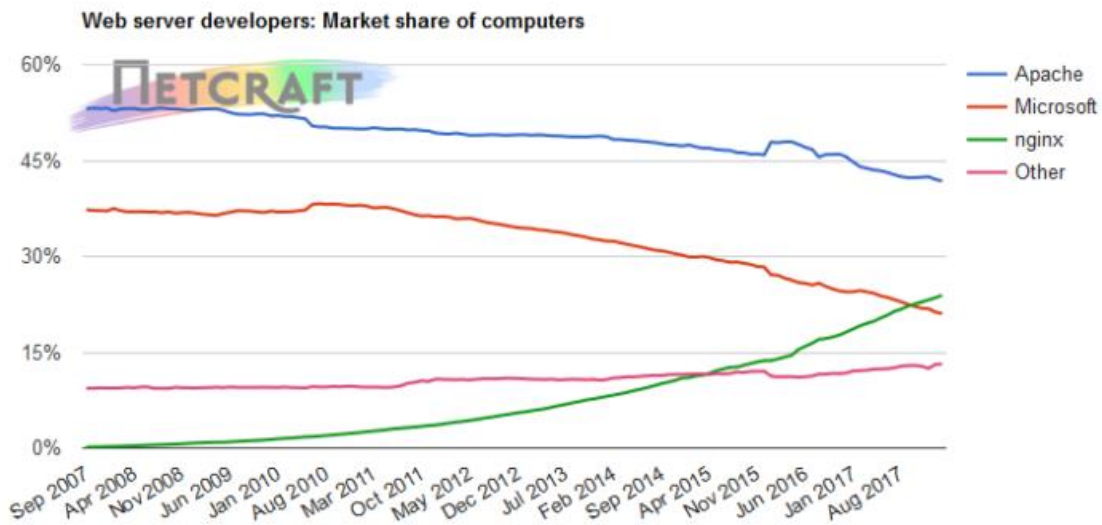


Figura 1.7: Cuota de mercado de NGINX. (Netcraft, 2015)

Se selecciona el servicio web NGINX, debido a su alto nivel de configuración, robustez y estabilidad, lo que hace de éste servidor una herramienta muy confiable gracias a su licencia de código abierto.

### 1.3.3 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. (Juárez y Ivonne, 2018)

#### PostgreSQL 10.5

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Distribución de Software Berkeley) y con código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Se seleccionó PostgreSQL en su versión 10.5 teniendo en cuenta la necesidad de utilizar herramientas informáticas libres para el desarrollo de software. Además de que es un gestor multiplataforma, confiable, estable, con gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad. (Momjian, 2001)

#### PgAdmin III 1.20.0

Se selecciona PgAdmin III en su versión 1.20.0 dado que es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. Está diseñado para responder a las necesidades de todos los

usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. (Ochoa Reyes et al., 2014)

#### **1.3.4 Marco de trabajo (Framework)**

En el desarrollo de software, un framework es un conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software. Un Framework proporciona la arquitectura partiendo el diseño en clases abstractas y definiendo sus responsabilidades y colaboraciones. Un desarrollador realiza una aplicación haciendo subclases y componiendo instancias a partir de las clases definidas por el Framework. (Gutiérrez, 2016)

#### **Django (1.8)**

Es un marco de trabajo web de código abierto escrito en Python que permite construir aplicaciones web más rápido y con menos código. Compatible con la mayor parte de los gestores de bases de datos: SQLite, PostgreSQL, MySQL, Oracle, entre otros. Django además es un framework multiplataforma, de fácil instalación y configuración tanto en GNU/Linux como en Microsoft Windows. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes. (Chun, 2001)

Se elige Django en su versión 1.8 como framework de desarrollo para la implementación de la solución propuesta, porque reúne las mejores prácticas del desarrollo web e integra muchas herramientas informáticas que aportan fortaleza al producto final.

#### **Metronic (5.1)**

Se trata de una plantilla de panel de respuesta basada en el marco de Bootstrap de Twitter. La plantilla es fácil de usar en múltiples tipos de aplicaciones. Provee una serie de controles y componentes HTML, que además de ser visualmente muy atractivos, tienen ya implementadas funcionalidades muy útiles que ahorrarán una cantidad de tiempo enorme a la hora de desarrollar aplicaciones web. (Themes, 2015)

#### **1.3.5 Entorno de Desarrollo Integrado (IDE)**

##### **PyCharm 2018.2.4**

PyCharm es uno de los entornos de desarrollo más populares para Python, que permite una fácil integración con otros lenguajes y frameworks (como JavaScript, HTML 5, CSS 3 y Django). El trabajo con PyCharm ofrece una gran cantidad de ventajas básicas: (Islam, 2015)

- **Editor inteligente:** Permite completar código con algunos atajos de teclado.
- **Refactorizar el código:** En términos generales, significa modificar el código sin comprometer la ejecución del mismo.
- **Interfaz de líneas de comandos:** Permite con el uso del intérprete de Python, un fácil acceso y mapeo de las clases del modelo en la base de datos.
- **Organización de estructura de proyectos:** Permite navegar a través de nuestro código, saltando entre las clases y métodos creados, haciendo el flujo de trabajo mucho más dinámico.

Se selecciona PyCharm como Entorno de Desarrollo Integrado ya que la aplicación web a desarrollarse consta de gran cantidad de archivos, bibliotecas, módulos y clases, lo que se hace fácil el acceso a las mismas con su uso. Además, PyCharm mantiene un control en forma de árbol con carpetas y archivos, lo que facilita la navegación y acceso a la estructura que usa Django con su patrón Modelo, Vista, Plantilla.

### **Conclusiones del capítulo**

El análisis de los sistemas homólogos permitió identificar las principales características y funcionalidades del proceso de gestión de activos fijos y útiles y herramientas. El estudio de las características del equipo y el tiempo disponible para el desarrollo del sistema, permitieron la identificación de la metodología de desarrollo de software XP como la más acertada para el desarrollo de la solución. Para la implementación de las funcionalidades se selecciona Python como lenguaje de programación del lado del servidor, HTML5, CSS3 y JavaScript del lado del cliente. Para el trabajo con los lenguajes seleccionados, se emplearán los marcos de trabajo Django 1.8 del lado del servidor y Metronic 5.1 del lado del cliente.

## CAPÍTULO 2. Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

En el vicedecanato de administración de la Facultad 2 se realizan un conjunto de procesos vinculados a la gestión de información de los útiles y herramientas. Sus descripciones y modelado se relacionan a continuación:

### 2.1 Procesos del vicedecanato de administración de la Facultad 2

#### Entrar útiles y herramientas desde almacén

Inicialmente llega el recurso proveniente del almacén con el vale de entrega correspondiente al mismo. La asistente de control le da entrada al recurso en el Modelo 1, reflejando el número del vale de entrega. En caso de que el recurso clasifique como un útil y herramienta en movimiento, la asistente de control genera 2 copias del Modelo 2 y realiza la entrega del recurso al responsable del útil o herramienta. El responsable firma ambas copias del Modelo 2, archivando una de ellas la secretaria. En caso de que el medio no clasifique como un útil y herramienta en movimiento, el responsable del área de responsabilidad material a la cual va dirigido el recurso genera el Modelo 4, que contiene la entrada del nuevo útil entregado por la secretaria. La figura 2.1 muestra el modelado de este proceso.

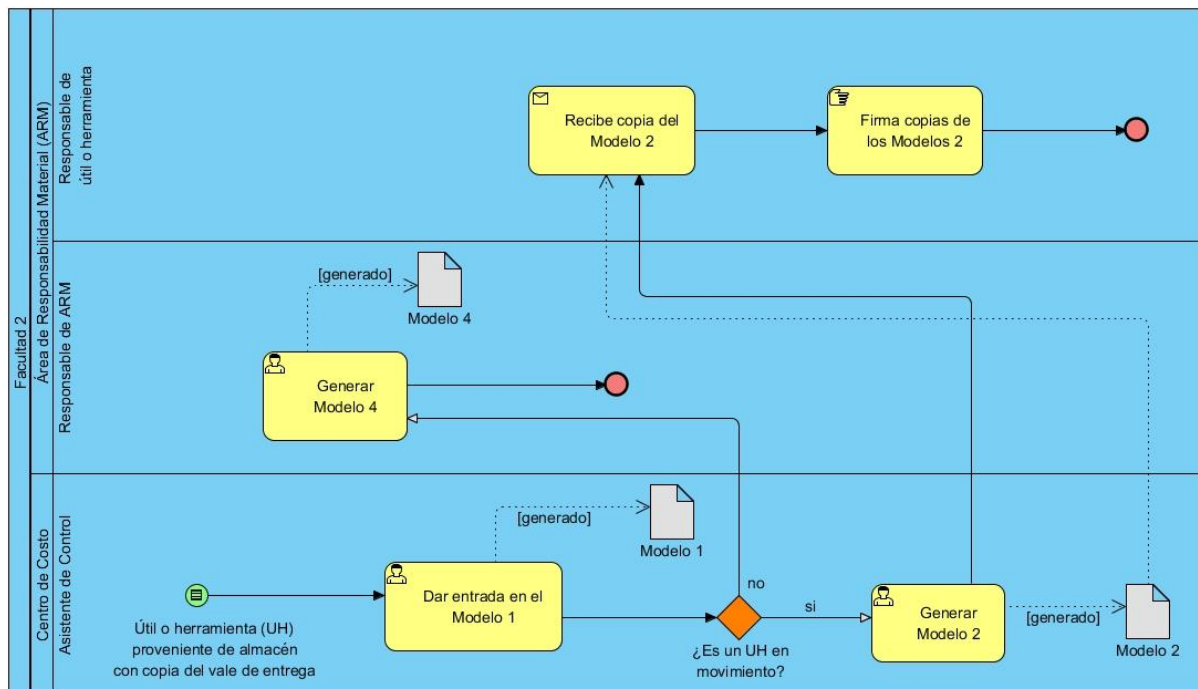


Figura 2.1: Entrar útiles y herramientas desde almacén. (Elaboración propia)

### Verificar inventario de útiles y herramientas

Los responsables de áreas de responsabilidad material como parte del control que llevan sobre los útiles y herramientas, realizan la verificación física de los recursos bajo su custodia. Para ello inicialmente generan el Modelo 3, y a partir del mismo realizan la verificación física de los recursos que se encuentran en el área. Por último, son tramitadas las diferencias identificadas durante el control realizado. La figura 2.2 muestra el modelado de este proceso.

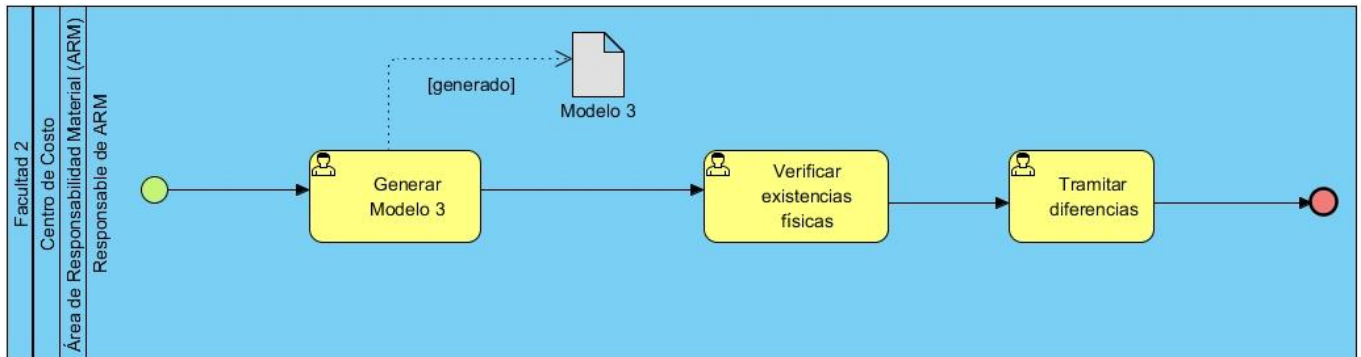


Figura 2.2: Verificar inventario de útiles y herramientas. (Elaboración propia)

### Baja de útiles y herramientas

El proceso de baja de útiles y herramientas inicia cuando el responsable de área de responsabilidad material solicita al vicedecano de administración dar baja a un útil o herramienta, por una determinada causa. Una vez realizada la solicitud, el vicedecano de administración es notificado sobre la solicitud de baja realizada y procede a generar el acta de baja, donde hace efectiva la baja del recurso. Posteriormente el responsable del área de responsabilidad material genera los Modelos 1 y 4, terminando de esta forma el proceso. La figura 2.3 muestra el modelado de este proceso.

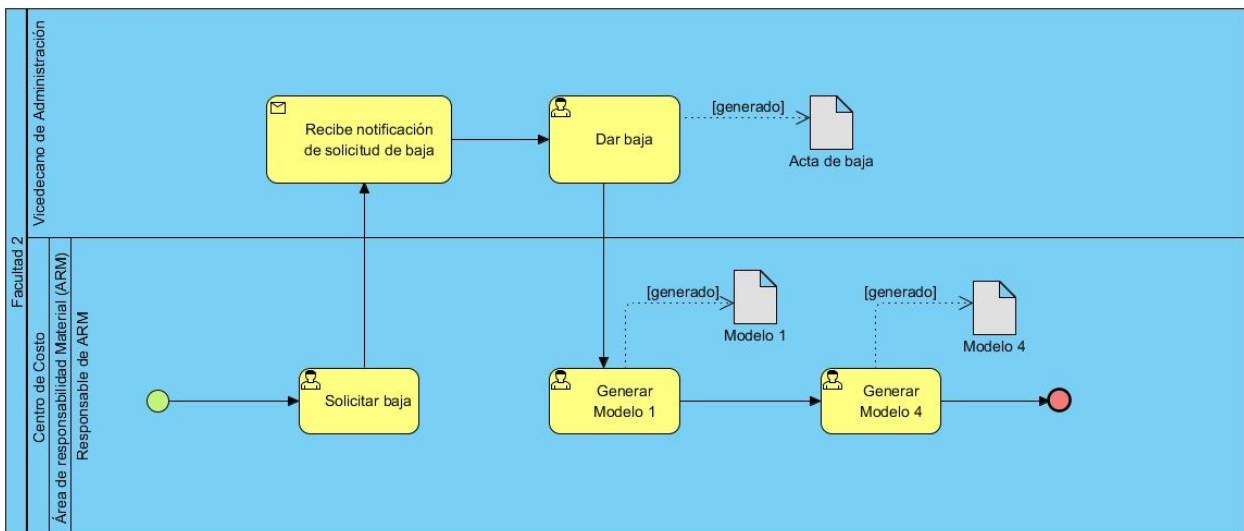


Figura 2.3: Baja de útiles y herramientas. (Elaboración propia)



### Registrar sobrante de útiles y herramientas

Cuando es identificado un sobrante en una determinada área de responsabilidad material, debe ser incorporado al registro de útiles de dicha área. Para ello el jefe del centro de costo registra el sobrante y genera el Anexo I, quedando registrada de esta forma la causa por la que se incorpora el útil o herramienta al registro de útiles del área de responsabilidad material. Seguidamente el responsable del área de responsabilidad material genera los Modelos 1 y 4 correspondientes a su área. De esta forma queda registrada la entrada del nuevo útil o herramienta identificado como sobrante, y concluye el proceso. El modelado de este proceso se muestra en la figura 2.4.

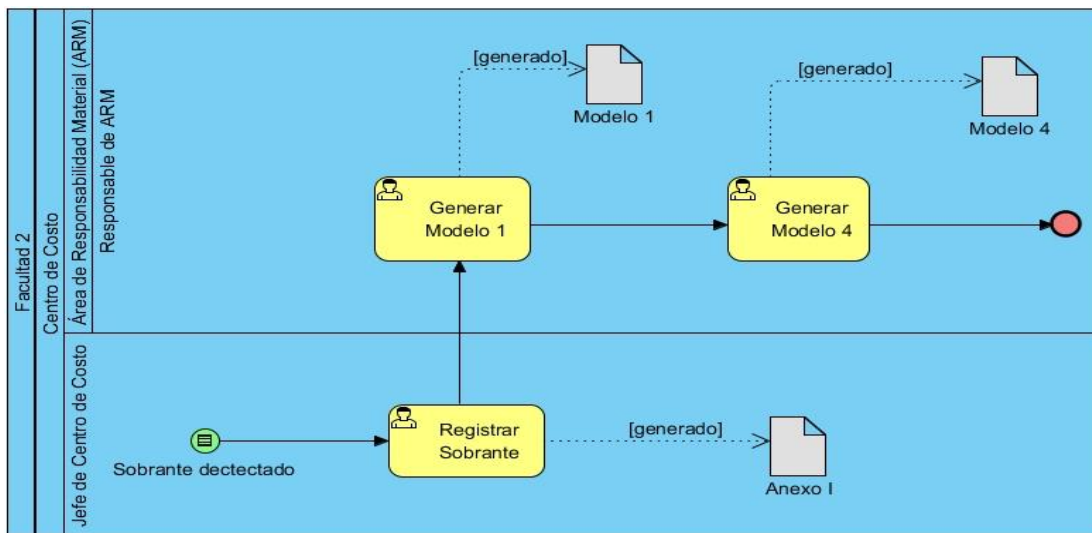


Figura 2.4: Registrar sobrante de útiles y herramientas. (Elaboración propia)

### Movimiento interno de útiles y herramientas

El proceso de realización de un movimiento interno inicia cuando el responsable del área de responsabilidad material (origen) registra el movimiento y genera el Anexo IV. Posteriormente se generan los Modelos 1 y 4 del área de responsabilidad material (origen). El responsable del área de responsabilidad material (destino) recibe el útil o herramienta movido a su área y procede a la generación de los Modelos 1 y 4, terminando de esta forma el proceso. La figura 2.5 muestra el modelado de este proceso.

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

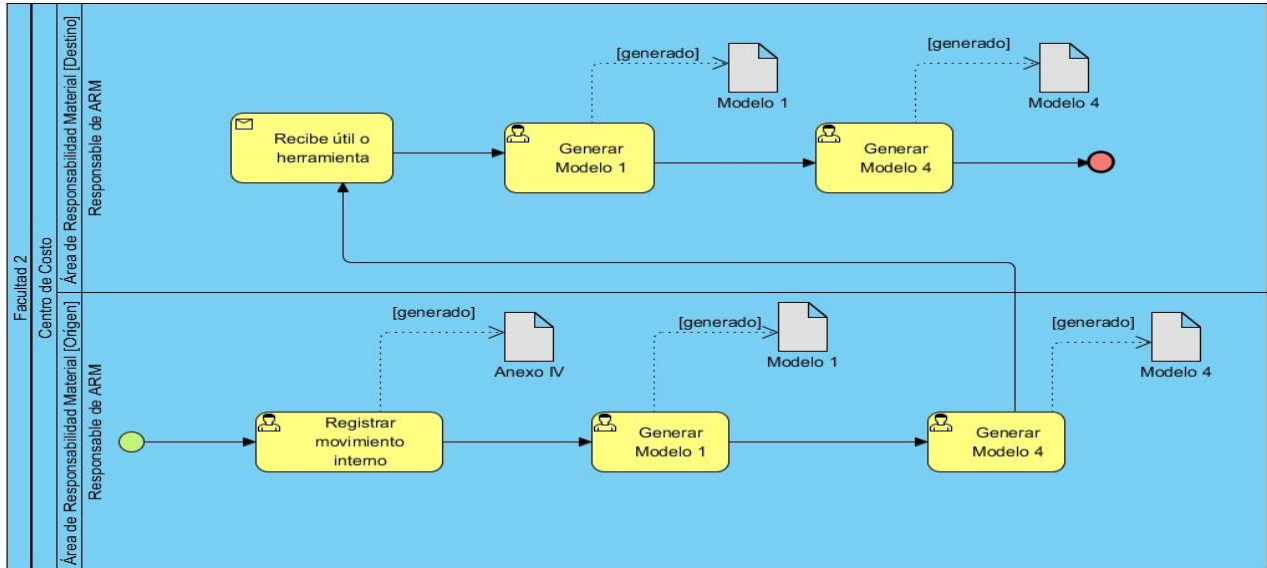


Figura 2.5: Movimiento interno de útiles y herramientas. (Elaboración propia)

**Movimiento externo de útiles y herramientas**

El proceso de realización de un movimiento externo inicia cuando el responsable del área de responsabilidad material (origen) registra el movimiento y genera el Anexo V. Posteriormente se generan los Modelos 1 y 4 del área de responsabilidad material (origen). El responsable del área de responsabilidad material (destino) recibe el útil o herramienta movido a su área y procede a la generación de los Modelos 1 y 4, terminando de esta forma el proceso. La figura 2.6 muestra el modelado de este proceso.

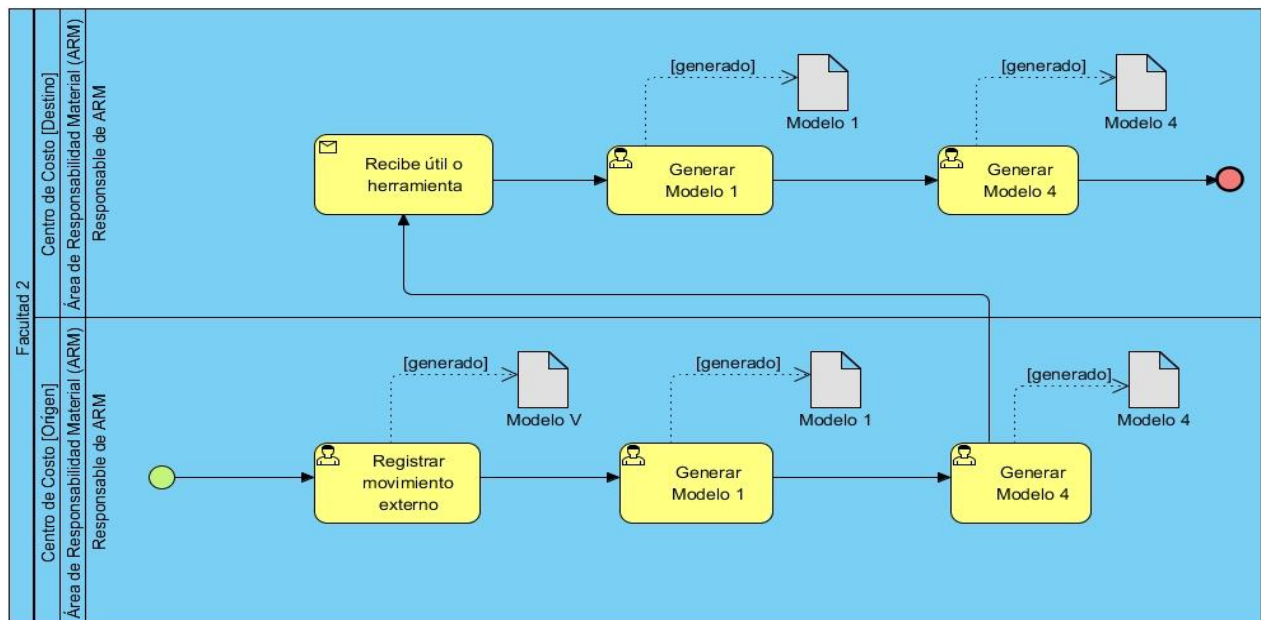


Figura 2.6: Movimiento externo de útiles y herramientas. (Elaboración propia)

## Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

El modelado de los procesos de negocio del área de gestión de útiles y herramientas del vicedecanato de administración de la Facultad 2, permitió identificar un conjunto de personas que interactuarán con el sistema y los requisitos funcionales que serán incluidos en la solución.

Para mejorar la implementación del sistema en lugar de trabajar por módulos, se escogió el desarrollo por roles, como propone y trabaja el framework seleccionado. Entre las características fundamentales que debe tener en cuenta el desarrollo de una aplicación es la especificación de la audiencia a la que está dirigida la misma, delimitando el nivel de interacción entre los diferentes usuarios. Los grupos o roles que estarán disponibles en la aplicación se relacionan a continuación: Vicedecano de Administración (VA1), Jefe de Centro de Costo (JCC1), Asistente de Control (AC1), Responsable de Área de Responsabilidad Material (RARM1) y Responsable de Útil y Herramienta (RUH1).

### **2.2 Definición de requisitos**

El propósito de la definición de requisitos es especificar las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, logrando un entendimiento entre el equipo de desarrollo y el cliente, y especificando las necesidades reales de forma que cumpla sus expectativas.(Rivadeneira Molina, 2012)

#### **2.2.1 Requisitos funcionales**

Estos requisitos se utilizan para determinar que hará el software, definiendo las relaciones de su operación y su implementación, sin olvidar que deben ser explícitos también en lo que el sistema no debe hacer y que validaciones se deben realizar.(Rivadeneira Molina, 2012)

RF 1 Entrada de útil y herramienta

RF 2 Entrada de útil y herramienta en movimiento

RF 3 Movimiento interno de útil y herramienta

RF 4 Movimiento externo de útil y herramienta

RF 5 Solicitar baja de útil y herramienta

RF 6 Baja de útil y herramienta

RF 7 Registrar sobrante de útil y herramienta

RF 8 Generar Modelo 1

RF 9 Generar Modelo 3

RF 10 Generar Modelo 4

## Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

RF 11 Generar reporte de útiles y herramientas en movimiento

RF 12 Crear Centro de Costo

RF 13 Crear Área de Responsabilidad Material

RF 14 Crear unidad de medida

RF 15 Crear causa de baja

RF 16 Autenticar usuario

RF 17 Crear perfil

RF 18 Consultar evidencias

### **2.2.2 Requisitos no funcionales**

Estos requerimientos se basan en las restricciones de los servicios o funciones ofrecidos por el sistema. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste. (Rivadeneira Molina, 2012)

Los requisitos no funcionales identificados para el sistema son:

Requisitos de interfaz: Las interfaces del sistema deben adaptarse a dispositivos móviles con distintas longitudes y resoluciones de pantallas.

Requisitos de rendimiento: El tiempo de respuesta de las funcionalidades de la aplicación debe ser el mínimo posible.

Requisitos de compatibilidad:

- La aplicación debe funcionar en los navegadores más utilizados para acceder a páginas web.

Requisitos de seguridad: Los usuarios solo puedan acceder a las funcionalidades definidas por su rol.

Requisitos de software: El equipo servidor donde se despliegue la aplicación necesitará la instalación de:

- Python en su versión 2.7 y la librería reportlab necesaria para la generación de documentos en formato PDF.
- Django en su versión 1.8.
- Servidor de aplicaciones NGINX.
- Servidor de base de datos PostgreSQL 9.4 o superior.

Requisitos de usabilidad: La navegación, acceso a componentes, menús y funcionalidades del sistema debe resultar intuitiva para los usuarios.

Requisitos de concurrencia: El tiempo de respuesta ante peticiones de un gran número de usuarios concurrentes debe ser el menor posible.

### 2.3 Propuesta del sistema

La propuesta de solución se encuentra enfocada a la implementación de una aplicación web, mediante la que se accederá a través de la red de la universidad y haciendo uso del Protocolo Ligero de Acceso a Directorios (LDAP) de autenticación. En caso que el servicio que implementa dicho protocolo no se encuentre disponible, se podrá acceder a la aplicación comprobando las credenciales de los usuarios de forma local en la base de datos del sistema. Una vez autenticado el usuario, podrá acceder a las funcionalidades disponibles en el sistema según su grupo o rol. La figura 2.7 muestran la estructura de la solución propuesta.

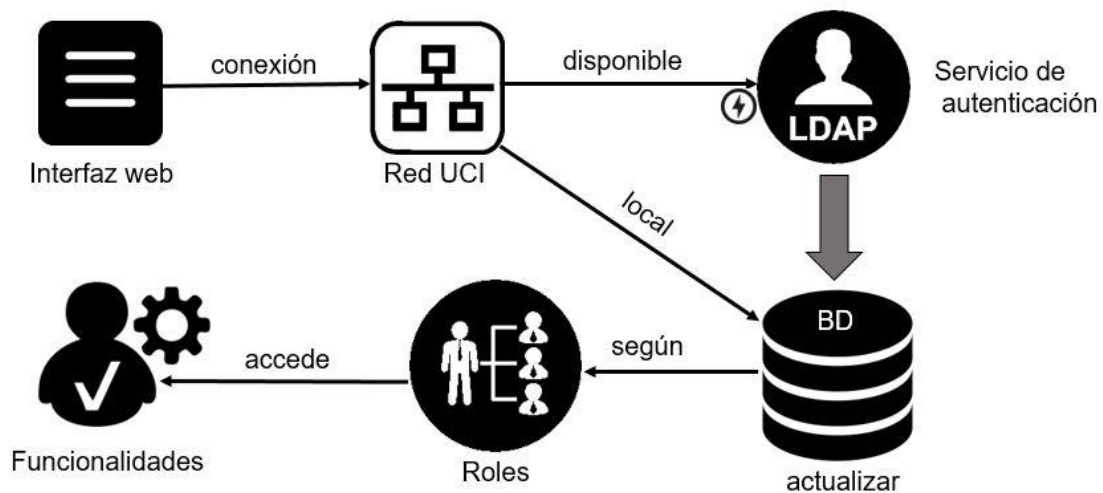


Figura 2.7: Propuesta del sistema. (Elaboración propia)

### 2.4 Exploración de la propuesta de solución

La metodología de desarrollo de software XP comienza con la fase de exploración, en esta fase, los clientes plantean a grandes rasgos las Historias de Usuario (HU) como artefacto que genera e identifica dicha fase. Las HU son porciones del comportamiento deseado de un sistema de software. Son muy utilizadas dentro del marco de desarrollo ágil, y sirven para dividir una gran cantidad de funcionalidades en partes más pequeñas para facilitar la planificación. (Letelier y Penadés, 2012)

Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas o pocos meses,

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2 dependiendo del tamaño del proyecto y familiaridad que tengan los programadores con la tecnología. (Letelier y Penadés, 2012)

### 2.4.1 Historias de usuario (HU)

Las historias de usuario constituyen descripciones cortas y detalladas, escritas por el equipo de desarrollo en conjunto con el cliente para lograr un mayor nivel de especificación de requisitos. Además, a partir de su definición los programadores deben poder ser capaces de realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. (Joskowicz, 2008)

Un punto de estimación equivale a una semana de programación, una semana de programación, corresponde a 40 horas de programación, 8 horas durante 5 días de la semana del calendario normal.

Cuando se hace referencia a Todos como Usuario se refiere a VA1, JCC1, RARM1, AC1 y RUH1.

Las HU del sistema se describen desde la tabla 2.1 a la tabla 2.16:

Tabla 2.1: Historia de usuario # 1

Historia de usuario	
<b>Número:</b> 1	<b>Nombre:</b> Autenticar usuario en el sistema
<b>Usuario:</b> Todos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> Al iniciar la aplicación se solicita nombre de usuario y contraseña. Si el usuario se autentica correctamente y aún no está activo, el sistema muestra una interfaz de bienvenida donde notifica al usuario que su solicitud está siendo atendida por los administradores, y en breve le será asignado un rol y área. En caso contrario el usuario accede a las funcionalidades descritas según su rol en su respectiva área. El sistema detecta y muestra un mensaje cuando hay campos vacíos o el usuario y la contraseña son incorrectos.	
<b>Observaciones:</b> El usuario debe introducir usuario y contraseña del dominio UCI. En caso de ser un usuario local, debe introducir usuario y contraseña locales para el acceso a la aplicación. Un usuario permanece inactivo en el sistema entre tanto el administrador no le asigne un rol y un área de responsabilidad material.	
<b>Prototipo de interfaz:</b> Ver figura 2.8.	

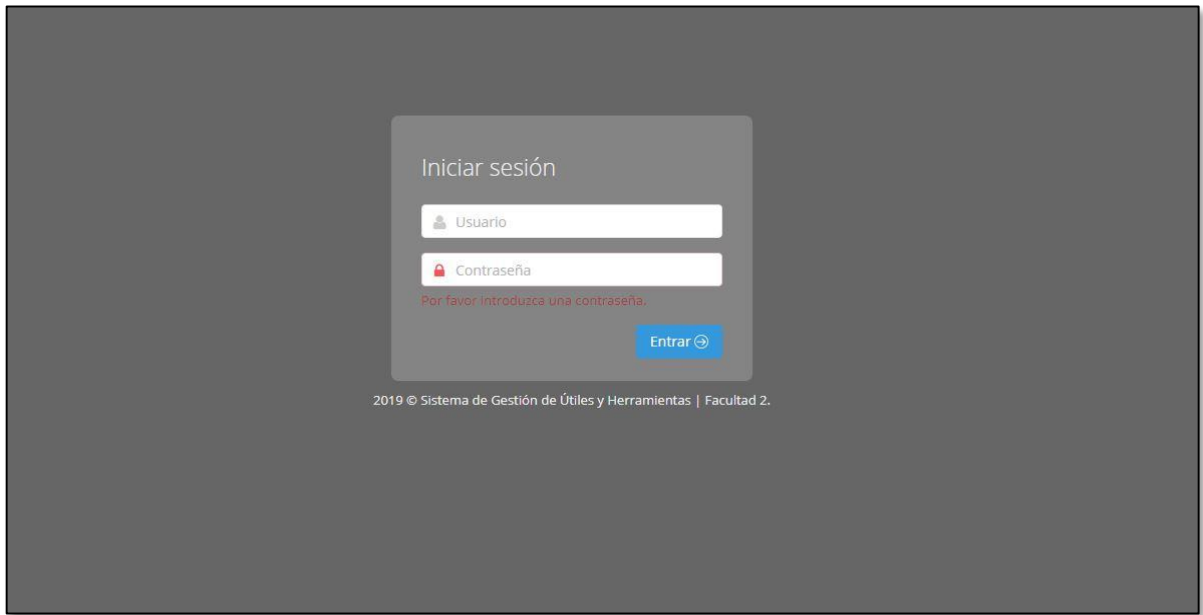


Figura 2.8: Prototipo de interfaz autenticar usuario. (Elaboración propia)

Tabla 2.2: Historia de usuario # 2

Historia de usuario	
<b>Número:</b> 2	<b>Nombre:</b> Crear centro de costo
<b>Usuario:</b> VA1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El VA1 podrá crear un nuevo centro de costo introduciendo los datos en un formulario, una vez creado, el sistema lo añadirá a una lista donde aparecen todos los centros de costo, dicha lista muestra además la cantidad de áreas que tiene cada centro, dando la posibilidad de acceder y generar los modelos 1, 3, 4 y el reporte de útiles y herramientas en movimiento para cada una de ellas. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Al momento de crearse un nuevo centro de costo se activa por defecto, y el valor de su fecha de actualización es igual a la fecha de creación.	
<b>Prototipo de interfaz:</b> Ver figura 2.9.	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

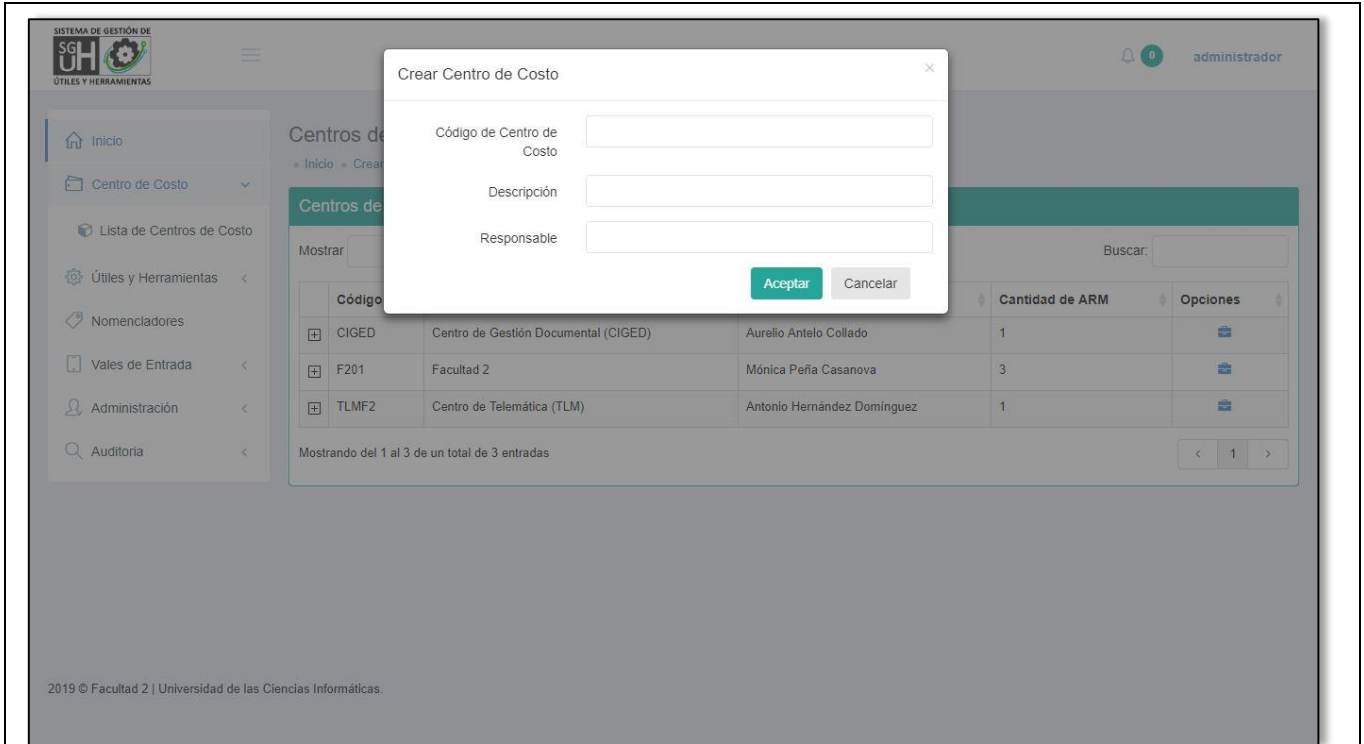


Figura 2.9: Prototipo de interfaz crear centro de costo. (Elaboración propia)

Tabla 2.3: Historia de usuario # 3

Historia de usuario	
<b>Número:</b> 3	<b>Nombre:</b> Crear área de responsabilidad material
<b>Usuario:</b> VA1, JCC1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> Se podrán crear nuevas áreas de responsabilidad material asociadas a un centro de costo, la nueva área creada se añadirá a una lista con información general del área, como su código, responsable y cantidad de útiles y herramientas que esta posee. Para cada área se podrán generar los modelos 1, 3, 4 y el reporte de útiles y herramientas en movimiento. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Las áreas de responsabilidad material pertenecen a un único centro de costo, al momento de su creación están activas.	
<b>Prototipo de interfaz:</b> Ver figura 2.10.	



Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

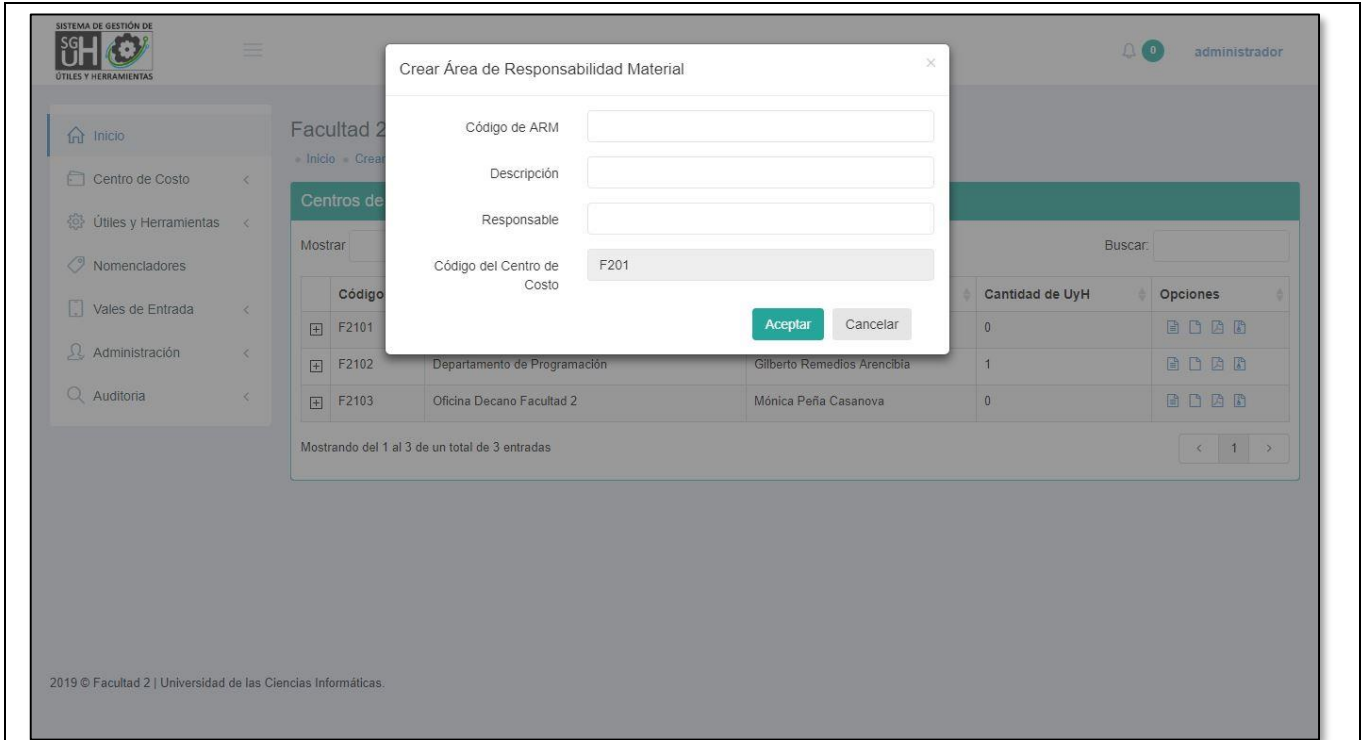


Figura 2.10: Prototipo de interfaz Crear área de responsabilidad material. (Elaboración propia)

Tabla 2.4: Historia de usuario # 4

Historia de usuario	
<b>Número:</b> 4	<b>Nombre:</b> Entrada de útil y herramienta
<b>Usuario:</b> VA1, JCC1, AC1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema debe permitir la entrada de nuevos útiles y herramientas especificando su código, descripción, unidad de medida, cantidad, precio en CUC, precio en CUP, área de responsabilidad a la que pertenecerá, número de vale y fecha de entrada. El nuevo útil se agregará a la lista de útiles y herramientas del área de responsabilidad especificada durante la entrada. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Los valores correspondientes al área de responsabilidad material a la que pertenece el útil y su unidad de medida, estarán definidos antes de realizar la entrada.	

**Prototipo de interfaz:** Ver figura 2.11.

The screenshot shows a web application interface for 'Registrar Entrada' (Register Entry) under the 'Útiles y Herramientas' (Tools and Equipment) section. The form includes the following fields:

- Código \*
- Descripción \*
- Unidad de medida \* (Dropdown menu: Seleccione una unidad)
- Cantidad \*
- Precio CUC \*
- Precio CUP \*
- ARM \* (Dropdown menu: Seleccione un área)
- Vale de entrada \*
- Fecha de entrada \* (Calendar widget showing May 2019, with the 25th selected)

At the bottom of the form are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel). The interface also features a sidebar with navigation options and a top header with the system logo and user name 'asistenteF2'.

Figura 2.11: Prototipo de interfaz Entrada de útil y herramienta. (Elaboración propia)

Tabla 2.5: Historia de usuario # 5

Historia de usuario	
<b>Número:</b> 5	<b>Nombre:</b> Registrar sobrante
<b>Usuario:</b> VA1, JCC1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El registro de un sobrante es una variante de entrada de útiles y herramientas al sistema. El sistema además de permitir el registro de sobrantes, al momento de realizar dicho registro debe generar el Anexo I-Sobrante. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Esta funcionalidad debe incluir la generación del Anexo I-Sobrantes, en formato PDF y tipo de hoja carta.	
<b>Prototipo de interfaz:</b> Ver figura 2.12.	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

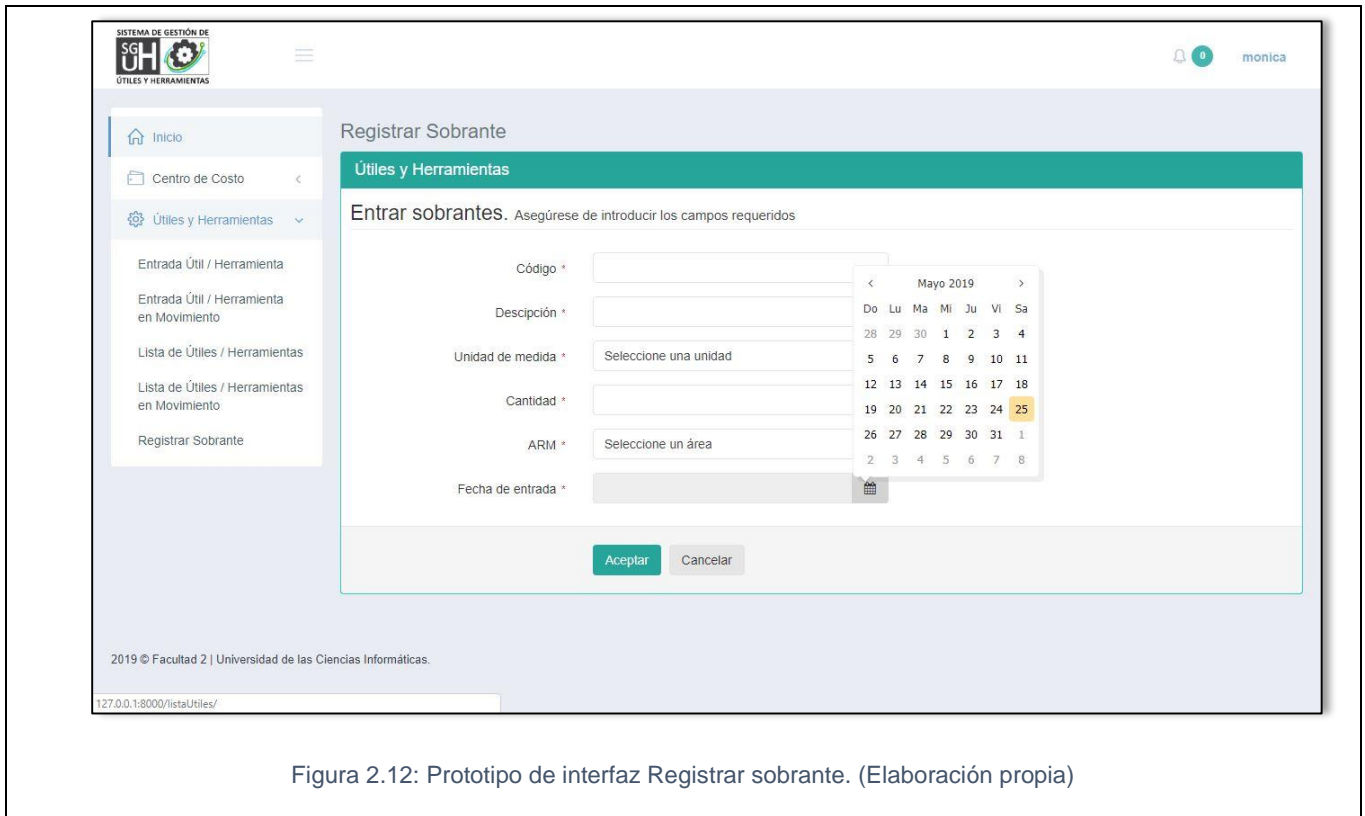


Figura 2.12: Prototipo de interfaz Registrar sobrante. (Elaboración propia)

Tabla 2.6: Historia de usuario # 6

Historia de usuario	
<b>Número:</b> 6	<b>Nombre:</b> Solicitar baja
<b>Usuario:</b> VA1, JCC1, RARM1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema debe permitir realizar solicitudes de baja de útiles y herramientas, especificando para cada caso la cantidad que se desea dar baja y su causa. Cuando se realiza una solicitud de baja, el sistema notifica al VA1. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> La baja queda pendiente a ser tramitada por parte del VA1.	
<b>Prototipo de interfaz:</b> Ver figura 2.13.	

The screenshot shows a web application interface for 'Solicitar Baja' (Request for Withdrawal) under the 'Útiles y Herramientas' (Tools and Instruments) section. The form includes the following fields and values:

- Código \***: 401001
- Descripción \***: SOPORTE DE TV
- Unidad de medida \***: U
- Cantidad \***: (empty field)
- Causa de baja \***: Seleccione una causa (dropdown menu)
- ARM \***: F2102
- Fecha de solicitud \***: 2019-5-25 (fecha actual)

Buttons: Solicitar baja (green), Cancelar (grey).

Footer: 2019 © Facultad 2 | Universidad de las Ciencias Informáticas.

Figura 2.13: Prototipo de interfaz Solicitar baja. (Elaboración propia)

Tabla 2.7: Historia de usuario # 7

Historia de usuario	
<b>Número:</b> 7	<b>Nombre:</b> Movimiento interno
<b>Usuario:</b> VA1, JCC1, RARM1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> Permite a los usuarios realizar movimientos de útiles y herramientas entre áreas de responsabilidad material de un mismo centro de costo, se deben especificar las áreas destino y origen, así como la cantidad de útiles que se desean mover. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Esta funcionalidad debe incluir la generación del Anexo IV-Movimiento interno, en formato PDF y tipo de hoja carta.	
<b>Prototipo de interfaz:</b> Ver figura 2.14.	

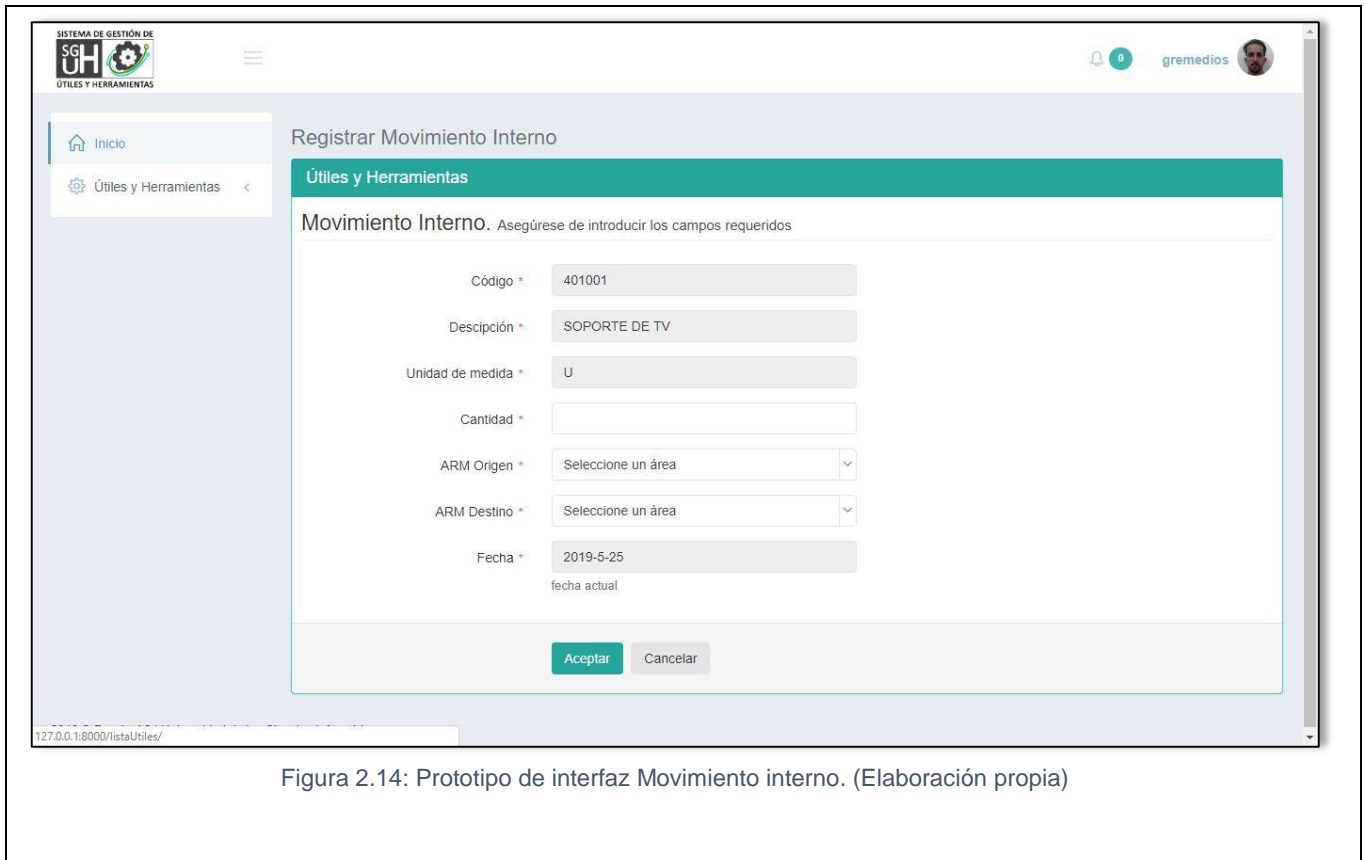


Figura 2.14: Prototipo de interfaz Movimiento interno. (Elaboración propia)

Tabla 2.8: Historia de usuario # 8

Historia de usuario	
<b>Número:</b> 8	<b>Nombre:</b> Movimiento externo
<b>Usuario:</b> VA1, JCC1, RARM1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> Permite a los usuarios realizar movimientos de útiles y herramientas entre áreas de responsabilidad material de pertenecientes a diferentes centros de costo, se deben especificar las áreas destino y origen, así como la cantidad de útiles que se desean mover. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Esta funcionalidad debe incluir la generación del Anexo V-Movimiento externo, en formato PDF y tipo de hoja carta.	
<b>Prototipo de interfaz:</b> Ver figura 2.15.	

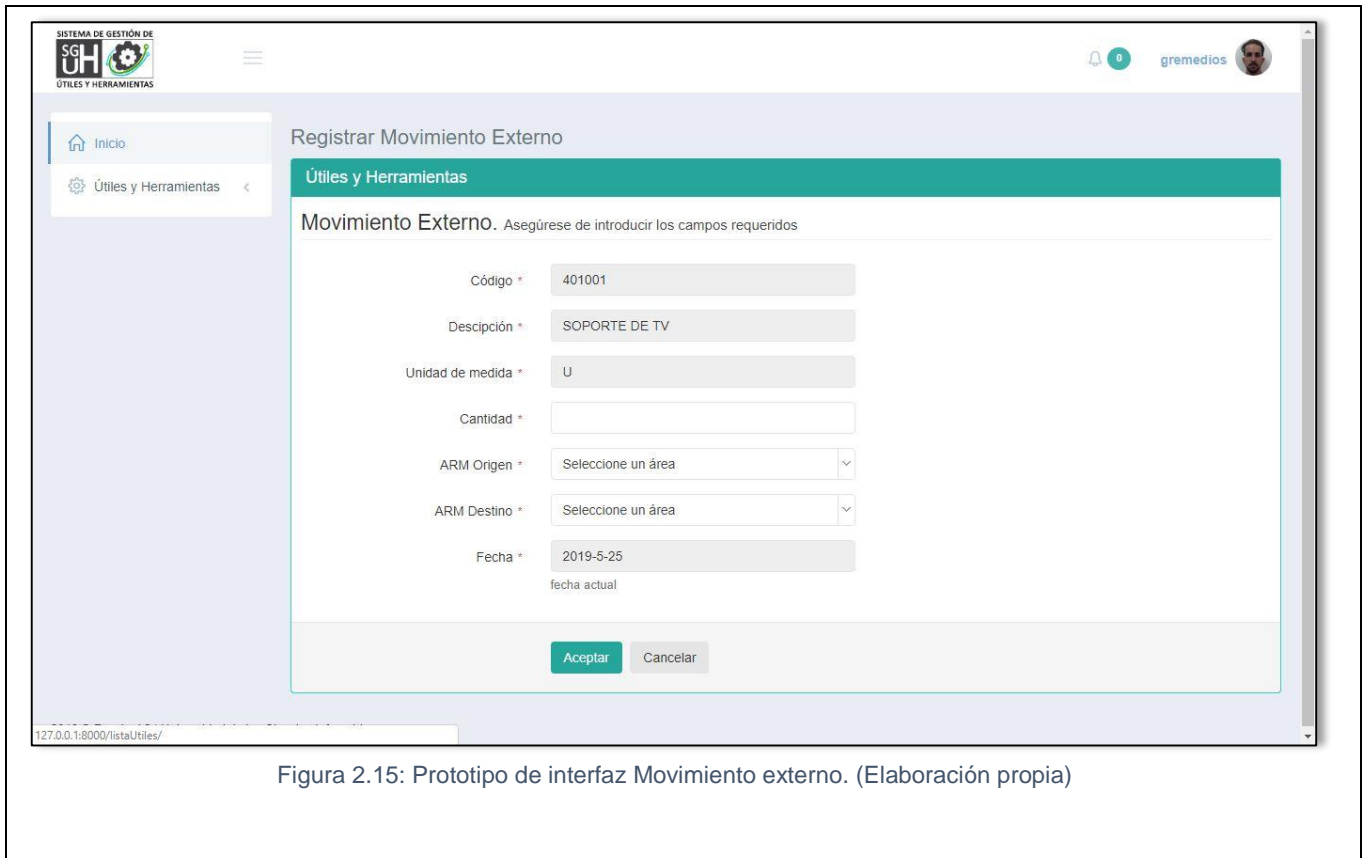


Figura 2.15: Prototipo de interfaz Movimiento externo. (Elaboración propia)

Tabla 2.9: Historia de usuario # 9

Historia de usuario	
<b>Número:</b> 9	<b>Nombre:</b> Crear perfil
<b>Usuario:</b> VA1	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permitirá seleccionar un usuario que no esté activo en el sistema y asignarle un área de responsabilidad material y un determinado rol. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Cada rol tendrá permisos diferentes según sus funciones.	
<b>Prototipo de interfaz:</b> Ver figura 2.16.	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

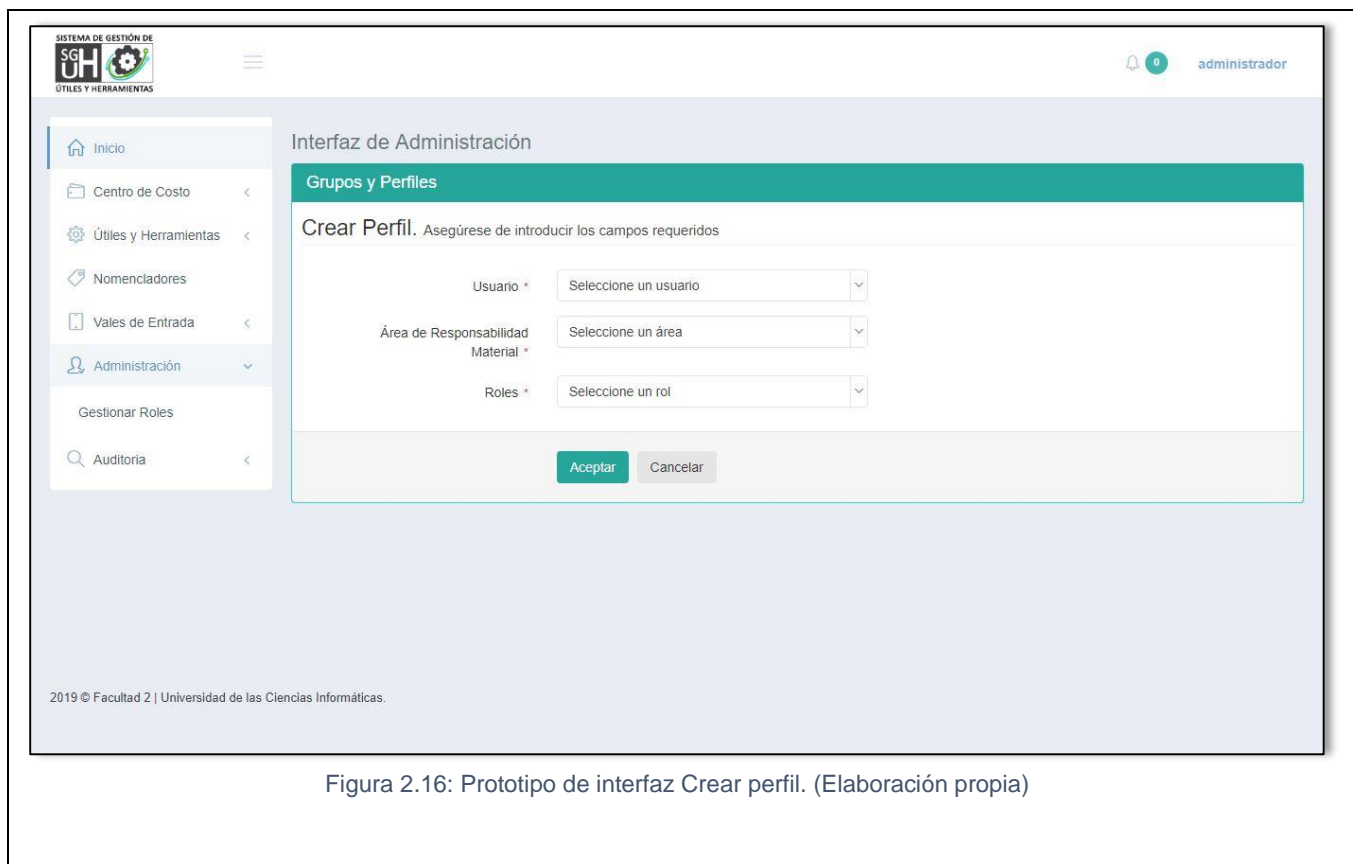


Figura 2.16: Prototipo de interfaz Crear perfil. (Elaboración propia)

Tabla 2.10: Historia de usuario # 10

Historia de usuario	
<b>Número:</b> 10	<b>Nombre:</b> Crear causa de baja
<b>Usuario:</b> VA1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> Se pueden insertar nuevas causas de baja, especificando una descripción. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Al momento de su creación cada causa estará activa en el sistema.	
<b>Prototipo de interfaz:</b> Ver figura 2.17.	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

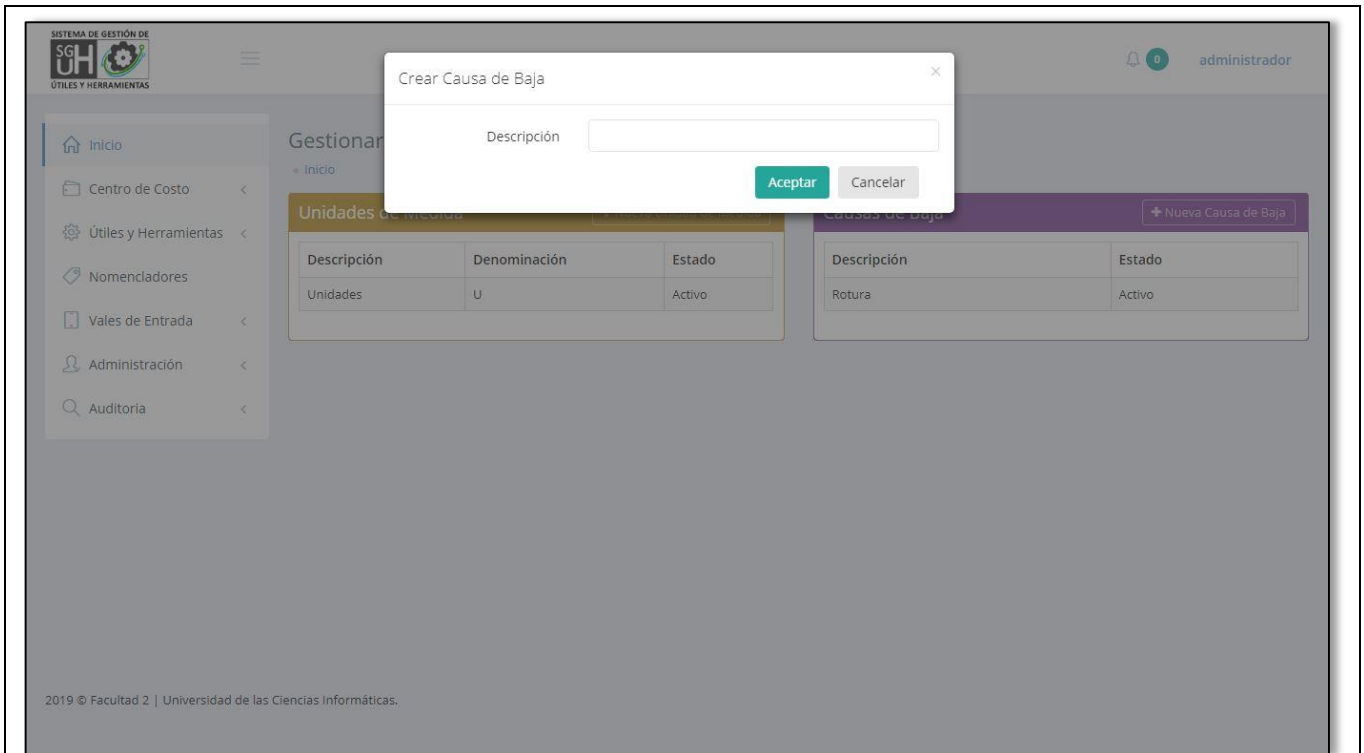


Figura 2.17: Prototipo de interfaz Crear causa de baja. (Elaboración propia)

Tabla 2.11: Historia de usuario # 11

Historia de usuario	
<b>Número:</b> 11	<b>Nombre:</b> Crear unidad de medida
<b>Usuario:</b> VA1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Bajo
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema debe permitir la especificación de nuevas unidades de medidas, de las mismas se deben especificar su descripción y denominación (Ejemplo: Unidades: U). El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Al momento de su creación las unidades de medidas creadas estarán activas en el sistema.	
<b>Prototipo de interfaz:</b> Ver figura 2.18.	



Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

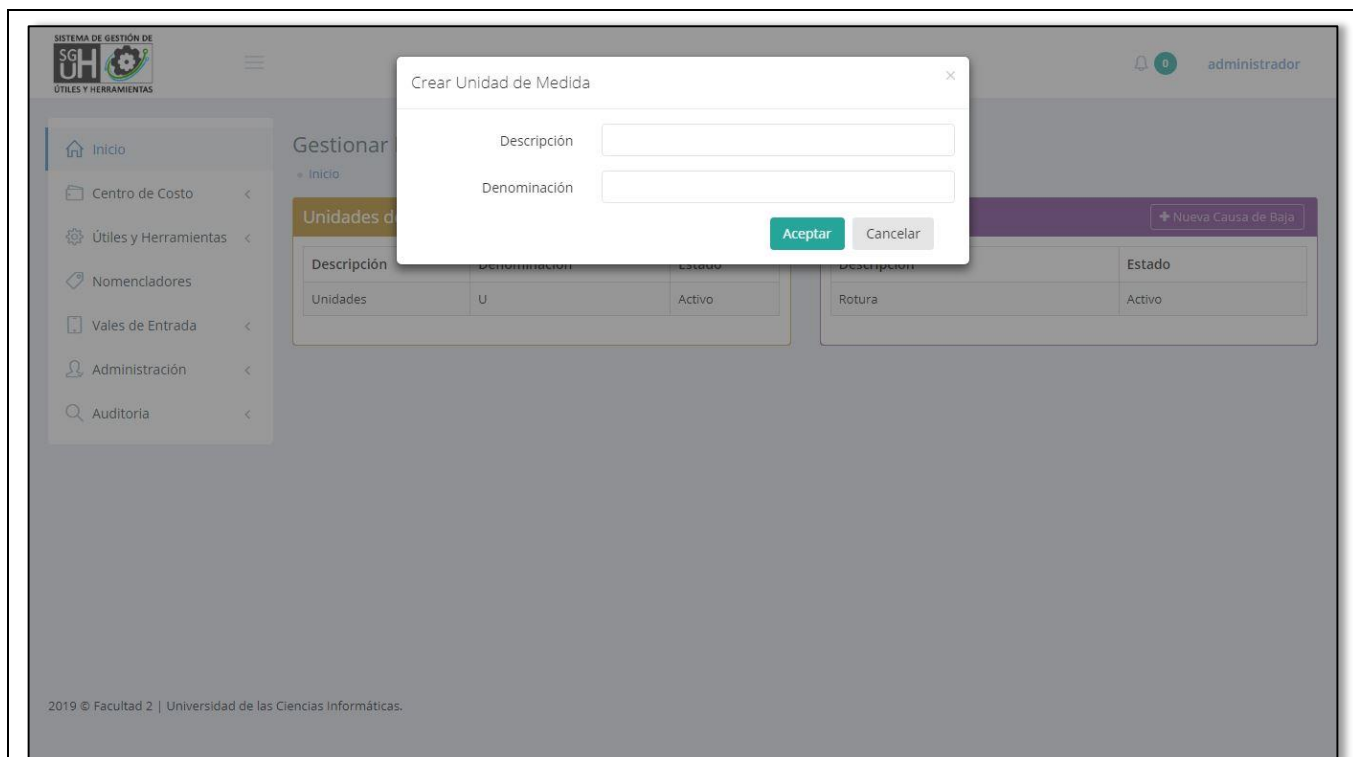


Figura 2.18: Prototipo de interfaz Crear unidad de medida. (Elaboración propia)

Tabla 2.12: Historia de usuario # 12

Historia de usuario	
<b>Número:</b> 12	<b>Nombre:</b> Generar Modelos (1, 3 y 4)
<b>Usuario:</b> VA1, JCC1, AC1, RARM1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema debe permitir generar los modelos 1, 3 y 4 correspondientes a la Norma 40 del 2007 del MES. La generación de dichos modelos se realizará a partir de la lista de útiles y herramientas de una determinada área de responsabilidad material.	
<b>Observaciones:</b> Los modelos deben ser exportados en formato PDF.	

Tabla 2.13: Historia de usuario # 13

Historia de usuario	
<b>Número:</b> 13	<b>Nombre:</b> Tramitar baja
<b>Usuario:</b> VA1	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema debe permitir hacer efectiva las bajas de útiles y herramientas, desagregando sus cantidades en sus respectivas áreas de responsabilidad material. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.	
<b>Observaciones:</b> Al finalizar el proceso de baja, el sistema genera el acta de baja en formato PDF y tipo de hoja carta.	

**Prototipo de interfaz:** Ver figura 2.19.

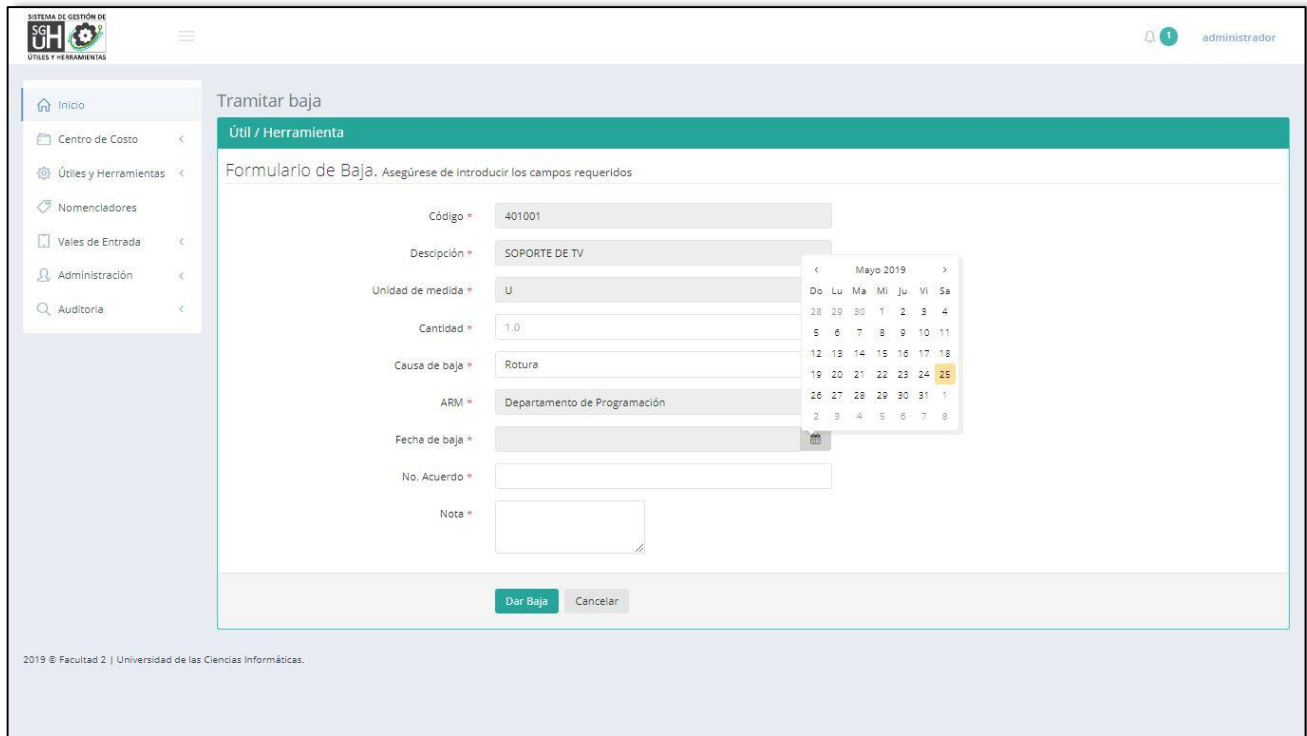


Figura 2.19: Prototipo de interfaz Tramitar baja. (Elaboración propia)

Tabla 2.14: Historia de usuario # 14

Historia de usuario	
<b>Número:</b> 14	<b>Nombre:</b> Entrada de útil y herramienta en movimiento
<b>Usuario:</b> VA1, JCC1, AC1	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo de desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Gilberto Remedios Arencibia	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

**Descripción:** El sistema permitirá la entrada de útiles y herramientas en movimiento. Especificando los datos correspondientes a una entrada de útiles y herramientas, además de un responsable del útil. El sistema detecta y notifica al usuario cuando hay campos vacíos o no válidos.

**Observaciones:** Al concluir la entrada se genera el Modelo 2-Acta de entrega, en formato PDF y tipo de hoja carta.

**Prototipo de interfaz:** Ver figura 2.20

Figura 2.20: Prototipo de interfaz Entrada de útil y herramienta en movimiento. (Elaboración propia)

Tabla 2.15: Historia de usuario # 15

Historia de usuario	
<b>Número:</b> 15	<b>Nombre:</b> Generar reporte de útiles y herramientas en movimiento
<b>Usuario:</b> VA1, JCC1, AC1, RARM1	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Gilberto Remedios Arancibia	
<b>Descripción:</b> El sistema permite la generación de un reporte de útiles y herramientas en movimiento, de una determinada área de responsabilidad material.	
<b>Observaciones:</b> El reporte debe generarse en formato PDF y tipo de hoja carta.	

Tabla 2.16: Historia de usuario # 16

Historia de usuario	
<b>Número:</b> 16	<b>Nombre:</b> Consultar evidencias
<b>Usuario:</b> VA1	
<b>Prioridad en negocio:</b> Media	<b>Riesgo de desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema debe garantizar la consulta por parte de los administradores, de las evidencias almacenadas a partir de la acciones realizadas por los usuarios.	
<b>Observaciones:</b> El sistema debe permitir el filtrado de las evidencias a partir de un usuario seleccionado. El sistema no debe permitir que la información de las evidencias sea modificada.	

**Prototipo de interfaz:** Ver figura 2.21.

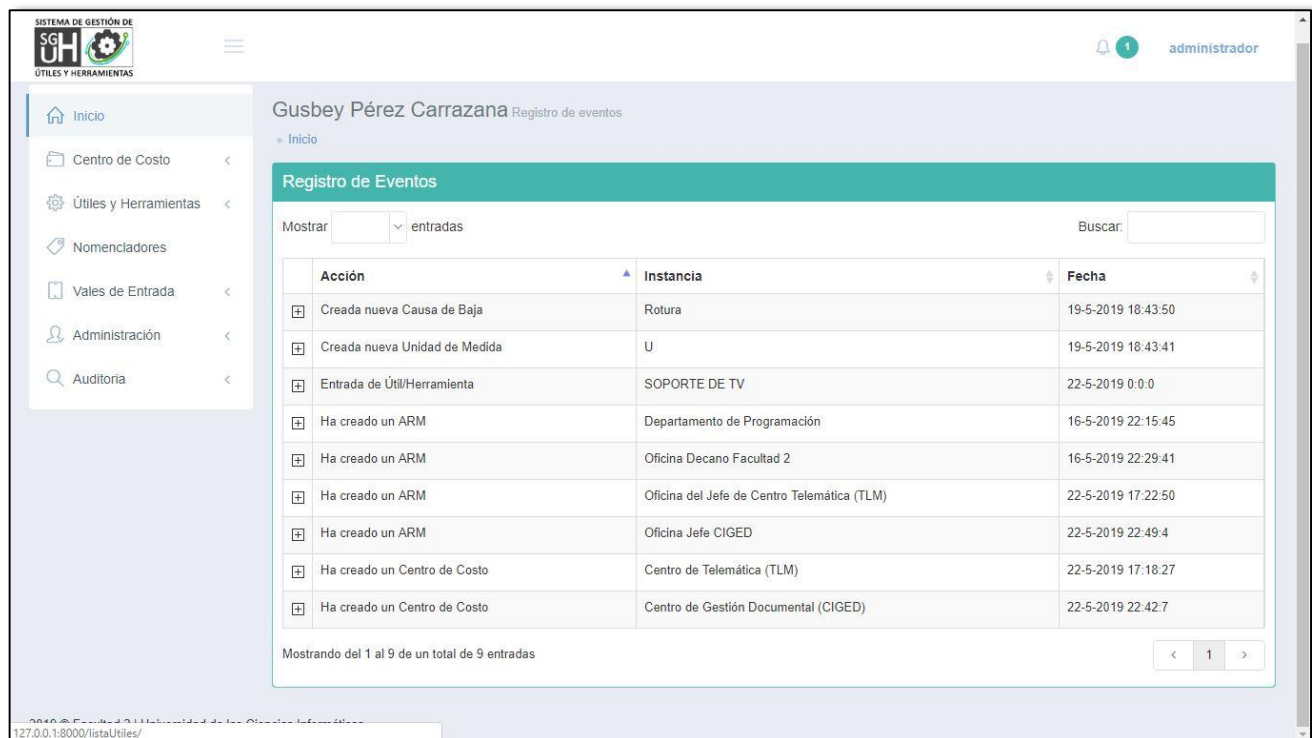


Figura 2.21: Prototipo de interfaz Consultar evidencias. (Elaboración propia)

## 2.5 Planificación

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente y los programadores. El proyecto comienza recopilando “Historias de usuario”, las que sustituyen a los tradicionales “casos de usos”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgo” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión con los distintos actores del proyecto, a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuario”. (Joskowicz, 2008)

### 2.5.1 Estimación de esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen rápidamente los programadores utilizando como medida el punto de estimación. Un punto de estimación equivale a una semana de programación.

Tabla 2.17: Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Autenticar usuarios en el sistema	1
	2	Crear centro de costo	1
	3	Crear áreas de responsabilidad material	1
	10	Crear causa de baja	0.5
	11	Crear unidad de medida	0.5
2	4	Entrada de útil y herramienta	1
	5	Registrar sobrante	1
	6	Solicitar baja	1
	12	Generar Modelos (1, 3 y 4)	2
	13	Tramitar baja	2
3	14	Entrada de útil y herramienta en movimiento	1
	7	Movimiento interno	1
	8	Movimiento externo	1
	9	Crear perfil	2
	15	Generar reporte de útiles y herramientas en movimiento	2
Total	16	Consultar evidencias	2
			<b>20.0</b>

Al realizar una estimación de esfuerzo por HU se concluye que las 16 HU agrupadas en 3 iteraciones poseen un tiempo de demora para su realización de 20 semanas.

### 2.5.2 Plan de iteraciones

El proceso de desarrollo está compuesto por varias iteraciones. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el proyecto. Sin embargo, esto no siempre es posible ya que es el cliente quien decide que HU se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción. (Letelier y Penadés 2012)

#### Iteración 1

En dicha iteración se implementarán las HU que por su impacto en el negocio constituyen la base para implementación de las demás funcionalidades. Además, se obtendrá una primera versión del producto en la que el cliente podrá probar algunas de las funcionalidades como: Autenticar usuarios en el sistema, Crear centro de costo, Crear área de responsabilidad material, Crear causa de baja y Crear unidad de medida, detalladas en las HU 1, 2, 3, 10 y 11 respectivamente.

#### Iteración 2

En esta iteración se implementarán las HU que además de recibir un nivel de prioridad alto por el cliente, dependen de la culminación de la primera iteración. Además, se obtendrá la segunda versión del producto en la que el cliente podrá probar algunas de las funcionalidades como: Entrada de útil y herramienta, Registrar sobrante, Solicitar baja, Generar Modelos (1, 3 y 4), Tramitar baja y Entrada de útiles y herramientas en movimiento, detalladas en las HU 4, 5, 6, 12, 13 y 14 respectivamente.

#### Iteración 3

En esta iteración se implementarán un grupo de HU que les fueron asignado un nivel de prioridad medio por el cliente, las mismas están relacionadas con la tercera etapa del desarrollo de la solución. Además, se obtendrá la tercera y última versión del producto en la que el cliente podrá probar algunas funcionalidades como: Movimiento interno, Movimiento externo, Crear perfil, Generar reporte de útiles y herramientas en movimiento y Consultar evidencias, detalladas en las HU 7, 8, 9, 15 y 16 respectivamente.

Tabla 2.18: Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Autenticar usuarios en el sistema	4
	2	Crear centro de costo	
	3	Crear área de responsabilidad material	
	10	Crear causa de baja	
	11	Crear unidad de medida	
2	4	Entrada de útil y herramienta	8
	5	Registrar sobrante	
	6	Solicitar baja	
	12	Generar Modelos (1, 3 y 4)	

Capítulo 2: Exploración y propuesta de solución del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

	13	Tramitar baja	
	14	Entrada de útiles y herramientas en movimiento	
3	7	Movimiento interno	8
	8	Movimiento externo	
	9	Crear perfil	
	15	Generar reporte de útiles y herramientas en movimiento	
	16	Consultar evidencias	
<b>Total</b>			<b>20.0</b>

### 2.5.3 Plan de Entrega

El plan de entrega, ver en la tabla 2.19, es el resultado de tomar acuerdos con el cliente sobre el contenido de la primera entrega y determinar un cronograma para las demás entregas del producto. Mediante el mismo el equipo de desarrollo mantiene al cliente informado y actualizado del progreso, funcionamiento y calidad del sistema. Gracias al plan de entrega que se confecciona a partir de las iteraciones del sistema tanto el cliente como el equipo de desarrollo se encuentran dirigidos hacia el avance y perfeccionamiento del producto. (Joskowicz, 2008)

A continuación, se refleja la evidencia de lo anteriormente planteado, donde se plasma la fecha en la que se va a realizar cada una de las entregas de las iteraciones del sistema por el equipo de desarrollo al cliente.

Tabla 2.19: Plan de Entregas

Iteración	Fecha de entrega
Iteración 1	26 de noviembre del 2018
Iteración 2	1 de marzo del 2019
Iteración 3	3 de mayo del 2019

### Conclusiones del capítulo

Con el desarrollo de las fases de la metodología XP implementadas se definieron los requisitos funcionales y no funcionales que debe cumplir el sistema. Las HU definidas establecieron un mayor nivel de especificación de requisitos, su estimación individual permitió a los programadores establecer una estimación poco riesgosa del tiempo que llevará su desarrollo. Después de realizada la estimación de esfuerzo que requiere el proyecto, se obtiene un plan de iteraciones y entregas, que permitirán tener un mayor control en el tiempo que posee el equipo de desarrollo para la entrega del producto al cliente.

## **CAPÍTULO 3. Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2**

En este capítulo se describen la fase de diseño, implementación y prueba. Se identifican la arquitectura y los patrones arquitectónicos empleados. Se obtiene el modelo de datos del sistema como punto de partida para su implementación. Se definirán las tareas de ingeniería necesarias para cada HU y las tarjetas Clase-Responsabilidad-Colaborador (CRC). Además, se evaluará la calidad de la aplicación a través de pruebas funcionales y no funcionales.

### **3.1 Arquitectura del sistema**

La Arquitectura de Software es la estructura de un sistema, compuesta de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.(Marini, 2012)

A continuación, se explica la arquitectura seleccionada para el sistema, cliente-servidor como aplicación web.

#### **3.1.1 Arquitectura Cliente-Servidor**

Se selecciona la arquitectura Cliente - Servidor porque:

Es en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes: una requiere servicios y la otra los brinda. Pero que, a la vez, pueden realizar actividades, tanto en forma conjunta como independientes.

El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen, cosa que ocurriría si ellos mismos desempeñan las funciones que les son proporcionadas de forma directa y transparente. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debido a la centralización de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el cliente como el servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes. (Marini, 2012)

### **3.2 Patrones de arquitectónicos**

Los patrones arquitectónicos expresan una organización estructural fundamental para un sistema software. Normalmente expresan un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y guías para organizar las relaciones entre ellos. (Reynoso y Kicillof, 2004)



### 3.2.1 Modelo Vista Plantilla

Para el desarrollo de la aplicación se selecciona el patrón de arquitectura Model Template View (MTV por sus siglas en inglés) ya que permite separar los datos, la interfaz y la lógica del sistema. Django es un marco de trabajo Modelo-Vista-Controlador (MVC). En Django, las tres capas básicas son el modelo, la vista, y la plantilla. Para entender MTV se debe analizar la analogía con la arquitectura MVC. El modelo en Django sigue siendo modelo, la vista se llama Plantilla (*Template*) y el controlador se llama Vista.

El patrón de arquitectura MTV cumple perfectamente con el fin particular de cualquier framework: una estructura bien definida que da soporte a un proyecto web, también ayuda a que el proyecto sea organizado y bien desarrollado. Para comprender mejor el funcionamiento de este patrón ver figura 3.1.

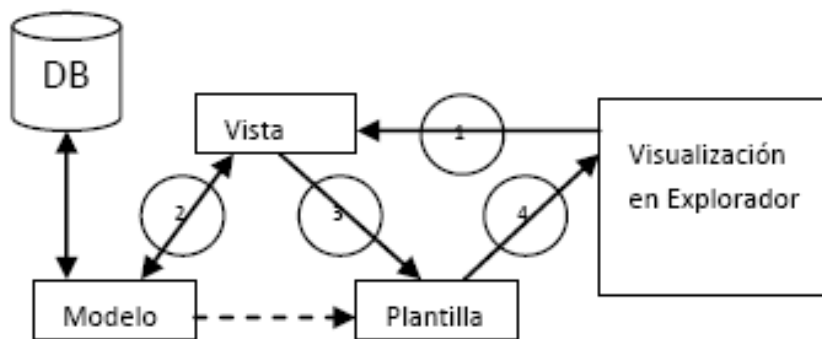


Figura 3.1: Patrón MTV. (Ayala, 2012)

A continuación, se definen cada una de las capas del patrón MTV:

El Modelo: Define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros y posee métodos. Todo esto permite indicar y controlar el comportamiento de los datos. (Ríos et al., 2016)

En la aplicación las clases del modelo se encuentran definidas en el archivo *models.py*. Ejemplo: *class UtilHerramienta*.

La Plantilla: Recibe los datos de la vista y luego los organiza para la presentación al navegador web. Las etiquetas que Django usa para las plantillas permiten que sea flexible para los diseñadores del *frontend*, incluso tiene estructuras de datos como *if*, por si es necesaria una presentación lógica de los datos, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código Python. Esto permite que la lógica del sistema siga permaneciendo en la vista. (Ríos et al., 2016)

En la aplicación las plantillas se agrupan dentro del directorio *templates* por un sistema de carpetas definidas por el desarrollador para una mejor organización del proyecto. Ejemplo: *templates/login/login.html*

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

La Vista: Se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados. El Mapeo Objeto Relacional (ORM) por sus siglas en inglés de Django, permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. Se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. (Ríos et al., 2016)

En la aplicación las vistas se encuentran definidas en el fichero *views.py*, el cual agrupa todas las vistas del sistema. Ejemplo: *CentroCostoView*

### 3.3 Patrones de diseño

En la programación orientada a objetos un patrón de diseño es una descripción de un problema y la solución, a la que se da un nombre, y se puede aplicar a nuevos contextos, idealmente proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. (Larman, 2003)

#### 3.3.1 Patrones de asignación de responsabilidades

Patrones **GRASP**, Patrones de Principios Generales para Asignar Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. (Larman, 2003)

**Experto en Información**: Se utiliza en la asignación de responsabilidades a las clases que poseen la información, expresa que los objetos hacen las cosas relacionadas con la información que poseen. (Larman, 2003)

El sistema de gestión de útiles y herramientas de la Facultad 2 (SGUH) hace uso de este patrón y se evidencia cuando se desean mostrar todos los útiles y herramientas almacenados en el sistema, ya que la única clase con la responsabilidad para conocer esta información es *UtilHerramienta*. De la misma manera sucede con las demás entidades del SGUH. Solo la clase *User* conoce quienes son los usuarios del sistema.

**Creador**: Asignarle a una clase la responsabilidad de crear una instancia de otra. Dentro del sistema este patrón se evidencia en las acciones de las vistas, las cuales crean objetos del modelo o los formularios que representan las entidades.

**Alta Cohesión**: En cuanto al diseño de objetos, la cohesión es una medida de la fuerza con la que se relacionan y el grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión. (Larman, 2003)

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Un ejemplo de utilización de dicho patrón se evidencia cuando se desea generar el Modelo 4. La generación del modelo es responsabilidad de la vista Modelo4View, pero quien tiene todos los datos que va a mostrar el modelo es la clase UtilHerramienta.

**Bajo Acoplamiento:** El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. (Larman, 2003)

Este patrón es usado por el framework Django, y por consiguiente en el sistema, al no asociar las clases del modelo con las de las plantillas o las vistas, la dependencia entre las clases, en este caso se mantiene baja.

**Controlador:** Es el encargado de asignar la responsabilidad de recibir y manejar un evento del sistema. Un Controlador define el método para la operación del sistema. (Larman, 2003)

Se evidencia el uso de este patrón en el SGUH, ya que para cada petición o evento que se genera en el mismo, existe una vista con la responsabilidad de obtenerla y devolver una respuesta. La respuesta puede ser mostrar una plantilla (*template*), generar un documento, devolver un mensaje, etcétera.

### 3.4 Diseño del sistema

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. El diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos. (Letelier y Penadés, 2012)

#### 3.4.1 Modelo de datos

El modelo de datos representa una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. (Silberschatz et al., 2002)

El modelo de datos, ver en la figura 3.2, en esta investigación fue el punto de partida en la implementación de un sistema de gestión de información en el área de útiles y herramientas que solucionara las necesidades específicas del vicedecanato de administración de la Facultad 2. Con el modelo de datos realizado, la lógica del negocio está representada a través de la relación entre las tablas y sus dependencias.

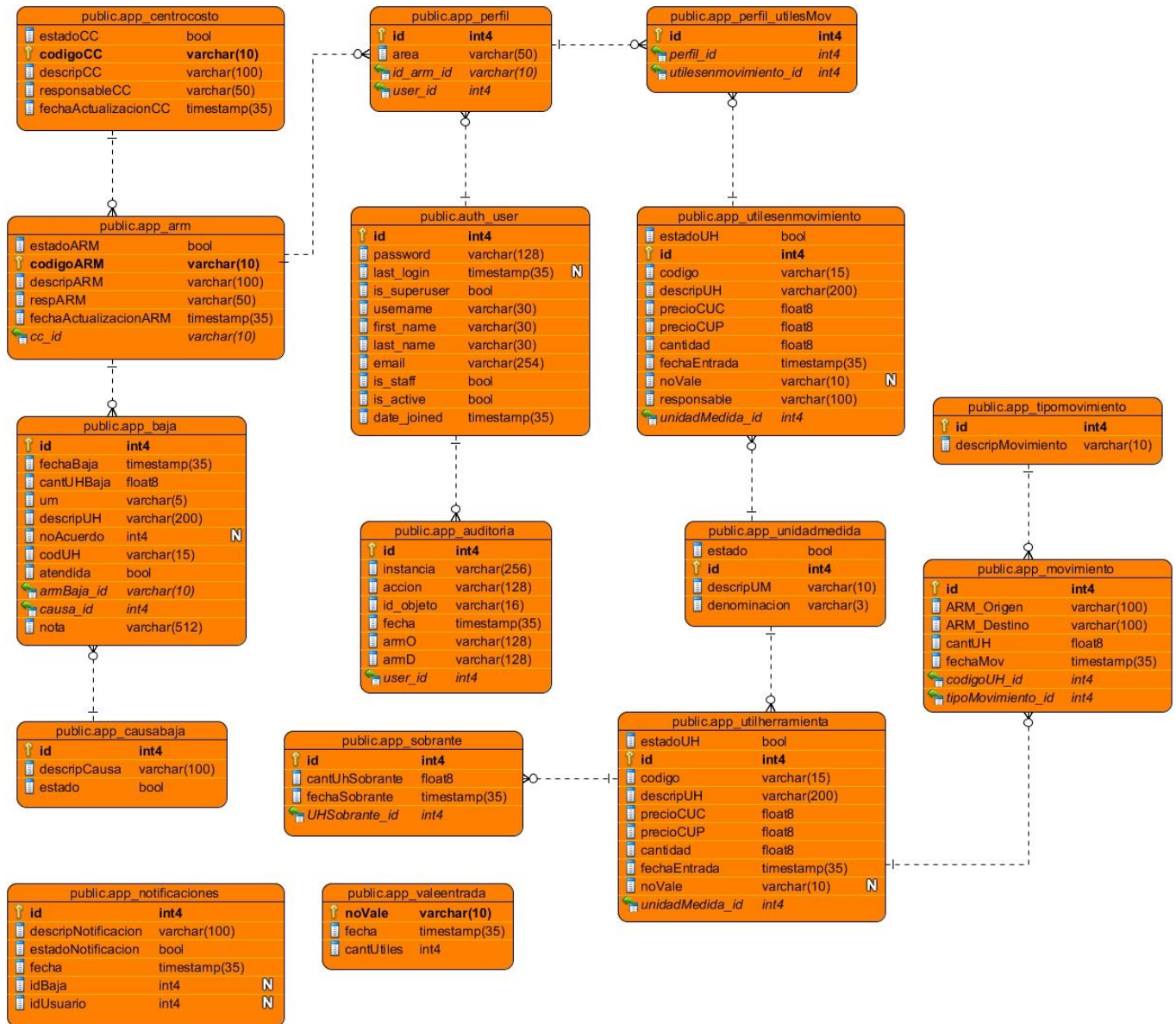


Figura 3.2: Diagrama Entidad Relación. (Elaboración propia)

### 3.4.2 Tarjetas Clase Responsabilidad Colaboración (CRC)

Las tarjetas CRC en la metodología XP se usan para el diseño de software orientado a objetos. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades. (Fernández y Luis, 2018)

A continuación, se describen las tarjetas CRC desde la tabla 3.1 a la tabla 3.10:

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Tabla 3.1: Tarjeta CRC # 1

Tarjeta CRC	
<b>Clase:</b> CentroCostoView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase se encarga de la creación de nuevos centros de costos.	CentroCosto

Tabla 3.2: Tarjeta CRC # 2

Tarjeta CRC	
<b>Clase:</b> ARMView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase se encarga de la creación de las nuevas áreas de responsabilidad material.	ARM CentroCosto

Tabla 3.3: Tarjeta CRC # 3

Tarjeta CRC	
<b>Clase:</b> UHView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de la gestión de los útiles y herramientas.	UtilHerramienta ARM UnidadMedida

Tabla 3.4: Tarjeta CRC # 4

Tarjeta CRC	
<b>Clase:</b> UHenMovimientoView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de la entrada en el sistema de útiles y herramientas en movimiento. Al finalizar la entrada, tiene la responsabilidad de exportar el Modelo 2 en formato PDF.	UtilHerramienta UtilesMovimiento ARM User UnidadMedida

Tabla 3.5: Tarjeta CRC # 5

Tarjeta CRC	
<b>Clase:</b> SobranteView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase se encarga del registro en el sistema de los útiles y herramientas que son detectados como sobrantes en una determinada área de responsabilidad material. Al finalizar el proceso de registro de un	Sobrante UtilHerramienta ARM UnidadMedida

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

sobrante, tiene la responsabilidad de exportar el Anexo I en formato PDF.	
---	--

Tabla 3.6: Tarjeta CRC # 6

Tarjeta CRC	
<b>Clase:</b> MovimientoInternoView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase se encarga de realizar movimientos internos de útiles y herramientas. Al finalizar el registro del movimiento, tiene la responsabilidad de exportar el Anexo IV en formato PDF.	Movimiento UtilHerramienta ARM CentroCosto TipoMovimiento

Tabla 3.7: Tarjeta CRC # 7

Tarjeta CRC	
<b>Clase:</b> MovimientoExternoView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase se encarga de realizar movimientos externos de útiles y herramientas. Al finalizar el registro del movimiento, tiene la responsabilidad de exportar el Anexo V en formato PDF.	Movimiento UtilHerramienta ARM CentroCosto TipoMovimiento

Tabla 3.8: Tarjeta CRC # 8

Tarjeta CRC	
<b>Clase:</b> UsuarioView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase se encarga de crear un perfil para los usuarios del sistema.	User auth_group ARM Perfil

Tabla 3.9: Tarjeta CRC # 9

Tarjeta CRC	
<b>Clase:</b> Modelo1View	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase tiene la responsabilidad de generar el Modelo 1 en formato PDF.	UtilHerramienta ARM

Tabla 3.10: Tarjeta CRC # 10

Tarjeta CRC	
<b>Clase:</b> Modelo3View	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase tiene la responsabilidad de generar el Modelo 3 en formato PDF.	UtilHerramienta ARM

El resto de las tarjetas CRC se encuentran en el Anexo 2.

### 3.5 Implementación

Para describir las tareas llevadas a cabo en la fase de implementación, se emplea un lenguaje técnico, el cual no necesariamente debe ser entendible por el cliente. Dichas tareas son asignadas al equipo o programador responsable. Esta labor se lleva a cabo con el objetivo de detallar mejor las historias de usuario, lo cual facilita el proceso de implementación. Cada historia de usuario puede contener una o más tareas de ingeniería, explicando de forma general las acciones que se realizan en la misma.

#### 3.5.1 Tareas de ingeniería

Las tareas de la ingeniería se realizan para especificar las acciones llevadas a cabo por los programadores en cada HU, haciendo mayor énfasis en los detalles necesarios para lograr una correcta implementación de las mismas.

A continuación, las tablas 3.11 a la tabla 3.26, muestran las tareas de ingeniería derivadas de cada HU.

Tabla 3.11: Tarea de ingeniería # 1

Tarea	
<b>Número de la tarea:</b> 1	<b>Número de Historia de usuario:</b> 1
<b>Nombre de la tarea:</b> Autenticar usuario en el sistema	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 29 de octubre del 2018	<b>Fecha de fin:</b> 2 de noviembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite que un usuario introduzca sus credenciales para autenticarse en la aplicación. Según sus privilegios, accederá a las funcionalidades y permisos correspondientes con su rol en el sistema. En caso de algún problema con el proceso de autenticación el usuario debe ser notificado, si el usuario se autentica correctamente accederá a la página de inicio del sistema.	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Tabla 3.12: Tarea de ingeniería # 2

Tarea	
<b>Número de la tarea:</b> 2	<b>Número de Historia de usuario:</b> 2
<b>Nombre de la tarea:</b> Crear centro de costo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 5 de noviembre del 2018	<b>Fecha de fin:</b> 9 de noviembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1 registrar los centros de costos con que cuenta la facultad actualmente, dígase centros de costos a cada uno de los centros de desarrollo pertenecientes a la facultad y la propia facultad constituye en sí un centro de costo.	

Tabla 3.13: Tarea de ingeniería # 3

Tarea	
<b>Número de la tarea:</b> 3	<b>Número de Historia de usuario:</b> 3
<b>Nombre de la tarea:</b> Crear área de responsabilidad material	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 12 de noviembre del 2018	<b>Fecha de fin:</b> 16 de noviembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1 (en cualquier centro de costo) y a los JCC1 (sólo en sus respectivos centros de costo) registrar nuevas áreas de responsabilidad material, dígase áreas de responsabilidad material a los departamentos, oficinas, aulas, laboratorios, locales, etcétera con que cuentan cada uno de los centros de costo de la facultad.	

Tabla 3.14: Tarea de ingeniería # 4

Tarea	
<b>Número de la tarea:</b> 4	<b>Número de Historia de usuario:</b> 4
<b>Nombre de la tarea:</b> Entrada de útil y herramienta	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 3 de diciembre del 2018	<b>Fecha de fin:</b> 7 de diciembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1, los JCC1 y las AC1 de cada uno de los centros de costo, la entrada de útiles y herramientas en las áreas de responsabilidad material.	



Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Tabla 3.15: Tarea de ingeniería # 5

Tarea	
<b>Número de la tarea:</b> 5	<b>Número de Historia de usuario:</b> 5
<b>Nombre de la tarea:</b> Registrar sobrante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 10 de diciembre del 2018	<b>Fecha de fin:</b> 14 de diciembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1 y los JCC1, el registro de sobrantes, al finalizar el proceso se debe generar el Anexo I en formato PDF y tipo de hoja carta.	

Tabla 3.16: Tarea de ingeniería # 6

Tarea	
<b>Número de la tarea:</b> 6	<b>Número de Historia de usuario:</b> 6
<b>Nombre de la tarea:</b> Solicitar baja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 14 de enero del 2019	<b>Fecha de fin:</b> 18 de enero del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite a los RARM1 realizar solicitudes de bajas sobre los diferentes útiles y herramientas que serán tramitadas por el VA1. El sistema notifica al VA1 cada vez que se realiza una nueva solicitud de baja.	

Tabla 3.17: Tarea de ingeniería # 7

Tarea	
<b>Número de la tarea:</b> 7	<b>Número de Historia de usuario:</b> 7
<b>Nombre de la tarea:</b> Movimiento interno	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 4 de marzo del 2019	<b>Fecha de fin:</b> 8 de marzo del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite a los RARM1, realizar movimientos internos de útiles y herramientas, generando para ello el Anexo IV en formato PDF y tipo de hoja carta.	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Tabla 3.18: Tarea de ingeniería # 8

Tarea	
<b>Número de la tarea:</b> 8	<b>Número de Historia de usuario:</b> 8
<b>Nombre de la tarea:</b> Movimiento externo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 11 de marzo del 2019	<b>Fecha de fin:</b> 15 de marzo del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite a los RARM1, realizar movimientos externos de útiles y herramientas, generando para ello el Anexo V en formato PDF y tipo de hoja carta.	

Tabla 3.19: Tarea de ingeniería # 9

Tarea	
<b>Número de la tarea:</b> 9	<b>Número de Historia de usuario:</b> 9
<b>Nombre de la tarea:</b> Crear perfil	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 4 de marzo del 2019	<b>Fecha de fin:</b> 8 de marzo del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1 crear un perfil para un nuevo usuario que se registre en la aplicación. Para la creación de un perfil el VA1 debe asignarle al usuario un rol y un área de responsabilidad material. Al finalizar el proceso el usuario queda activo y puede acceder al sistema.	

Tabla 3.20: Tarea de ingeniería # 10

Tarea	
<b>Número de la tarea:</b> 10	<b>Número de Historia de usuario:</b> 10
<b>Nombre de la tarea:</b> Crear causa de baja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha de inicio:</b> 19 de noviembre del 2018	<b>Fecha de fin:</b> 21 de noviembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema le permite al VA1 crear diferentes causas de bajas, utilizadas en los procesos de solicitud y aprobación de bajas que se realizan en la aplicación.	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Tabla 3.21: Tarea de ingeniería # 11

Tarea	
<b>Número de la tarea:</b> 11	<b>Número de Historia de usuario:</b> 11
<b>Nombre de la tarea:</b> Crear unidad de medida	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha de inicio:</b> 21 de noviembre del 2018	<b>Fecha de fin:</b> 23 de noviembre del 2018
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema le permite al VA1 crear diferentes unidades de medidas, utilizadas en los procesos de entrada de útiles y herramientas.	

Tabla 3.22: Tarea de ingeniería # 12

Tarea	
<b>Número de la tarea:</b> 12	<b>Número de Historia de usuario:</b> 12
<b>Nombre de la tarea:</b> Generar Modelos (1, 3 y 4)	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 21 de enero del 2019	<b>Fecha de fin:</b> 1 de febrero del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1, JCC1, AC1 y RARM1, la generación de los modelos 1, 3 y 4 de la Norma 40 del 2007 del MES en formato PDF.	

Tabla 3.23: Tarea de ingeniería # 13

Tarea	
<b>Número de la tarea:</b> 13	<b>Número de Historia de usuario:</b> 13
<b>Nombre de la tarea:</b> Tramitar baja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 4 de febrero del 2019	<b>Fecha de fin:</b> 15 de febrero del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1 tramitar las solicitudes de bajas realizadas por los RARM1, generando en cada caso el Acta de baja en formato PDF y tipo de hoja carta.	

Tabla 3.24: Tarea de ingeniería # 14

Tarea	
<b>Número de la tarea:</b> 14	<b>Número de Historia de usuario:</b> 14

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<b>Nombre de la tarea:</b> Entrada de útil y herramienta en movimiento	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha de inicio:</b> 18 de febrero del 2019	<b>Fecha de fin:</b> 22 de febrero del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema le permite al VA1, JCC1 y AC1 la entrada de útiles y herramientas en movimiento, generando en cada caso el Modelo 2 en formato PDF y tipo de hoja carta.	

Tabla 3.25: Tarea de ingeniería # 15

<b>Tarea</b>	
<b>Número de la tarea:</b> 15	<b>Número de Historia de usuario:</b> 15
<b>Nombre de la tarea:</b> Generar reporte de útiles y herramientas en movimiento	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 1 de marzo del 2019	<b>Fecha de fin:</b> 12 de marzo del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1, JCC1, AC1 y RARM1, la generación de un reporte en formato PDF y tipo de hoja carta de los útiles y herramientas en movimiento de una determinada área de responsabilidad material.	

Tabla 3.26: Tarea de ingeniería # 16

<b>Tarea</b>	
<b>Número de la tarea:</b> 16	<b>Número de Historia de usuario:</b> 16
<b>Nombre de la tarea:</b> Consultar evidencias	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 15 de abril del 2019	<b>Fecha de fin:</b> 26 de abril del 2019
<b>Programador responsable:</b> Gilberto Remedios Arencibia	
<b>Descripción:</b> El sistema permite al VA1 consultar el registro de eventos (trazas) de todos los usuarios registrados en la aplicación.	

### 3.6 Pruebas de validación del sistema

El proceso de producción de software requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. (Mera Paz, 2016)

### 3.6.1 Pruebas unitarias

Las pruebas unitarias se implementan a la vez que se realiza el proceso de producción de código, y son ejecutadas constantemente ante cada modificación del sistema. Esta forma de usar las pruebas unitarias ayuda a priorizar y comprobar la evolución del desarrollo y ofrecen una constante retroalimentación. (Robles y Ferrer, 2006)

En la Figura 3.3 se muestra la interfaz del resultado de la aplicación de pruebas unitarias, ejecutadas haciendo uso del módulo *unittest* integrado como librería estándar de Python. La utilización de dicha estrategia de prueba utilizando el módulo *unittest* se encuentra respaldada por la documentación oficial del framework, ya que constituye la forma preferida de realización de pruebas unitarias en Django, por la utilización de un enfoque basado en clases. Como se puede observar en la figura se realizaron un total de 11 test unitarios de los cuales 11 fueron exitosos.

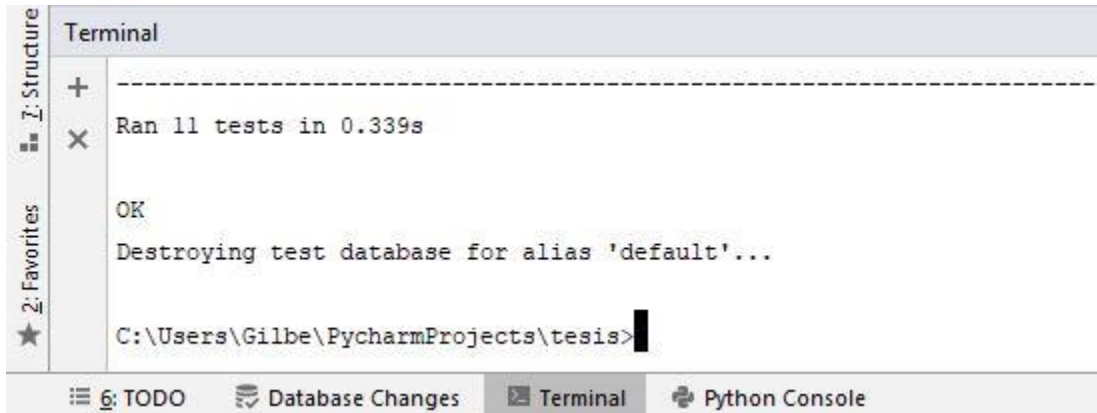


Figura 3.3: Resultados de la ejecución de los test unitarios. (Elaboración propia)

### 3.6.2 Pruebas de aceptación

Las pruebas de aceptación son realizadas por el cliente o por el usuario final para constatar que el sistema hace realmente lo que él quiere. En caso de que existan fallos, debe especificar la prioridad en que deben ser solucionados los diferentes problemas encontrados. Este tipo de pruebas son pruebas de caja negra y se hacen contra las historias de usuarios. (Robles y Ferrer, 2006)

Las pruebas de aceptación se muestran a partir de la tabla 3.27 a la tabla 3.36:

Tabla 3.27: Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre: Autenticar usuario en el sistema.	
Descripción: Prueba para la funcionalidad Autenticar usuario en el sistema.	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<p><b>Condiciones de ejecución:</b>                  El usuario y la contraseña del dominio UCI deben ser válidos.                  El usuario y contraseña de usuarios locales deben ser válidos.</p>
<p><b>Pasos de ejecución:</b>                  Se intenta autenticar un usuario en el sistema con los datos válidos.</p>
<p><b>Resultados esperados:</b> El usuario se autentica correctamente en el sistema.</p>
<p><b>Evaluación de la prueba:</b> Satisfactorio.</p>

Tabla 3.28: Prueba de aceptación # 2

Caso de prueba de aceptación	
<b>Código:</b> HU2_P2	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Crear centro de costo.	
<b>Descripción:</b> Prueba para la funcionalidad Crear centro de costo.	
<p><b>Condiciones de ejecución:</b>                  El VA1 es el único que tiene los permisos para crear nuevos centros de costo.                  El formulario de entrada no puede contener campos vacíos.</p>	
<p><b>Pasos de ejecución:</b>                  El VA1 accede a la creación de los centros de costo.</p>	
<b>Resultados esperados:</b> Los centros de costos son creados correctamente, mostrando sus respectivos datos y posibilitando acceder a sus áreas de responsabilidad material.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 3.29: Prueba de aceptación # 3

Caso de prueba de aceptación	
<b>Código:</b> HU3_P3	<b>Historia de usuario:</b> 3
<b>Nombre:</b> Crear áreas de responsabilidad material.	
<b>Descripción:</b> Prueba para la funcionalidad Crear áreas de responsabilidad material.	
<p><b>Condiciones de ejecución:</b>                  El VA1 y los JCC1 son los únicos que tienen los permisos para crear nuevas áreas de responsabilidad material.                  El formulario de entrada no puede contener campos vacíos.</p>	
<p><b>Pasos de ejecución:</b>                  El VA1 accede a la creación de áreas de responsabilidad material de cualquier centro de costo.                  Los JCC1 acceden a la creación de áreas de responsabilidad material sólo de su respectivo centro de costo.</p>	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<b>Resultados esperados:</b> Las áreas de responsabilidad material son creadas correctamente, mostrando sus respectivos datos y posibilitando la generación de los Modelos 1, 3 y 4.
<b>Evaluación de la prueba:</b> Satisfactorio.

Tabla 3.30: Prueba de aceptación # 4

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU4_P4	<b>Historia de usuario:</b> 4
<b>Nombre:</b> Entrada de útil y herramienta.	
<b>Descripción:</b> Prueba para la funcionalidad Entrada de útil y herramienta.	
<p><b>Condiciones de ejecución:</b></p> <p>El VA1, los JCC1 y las AC1 son los únicos que tienen permisos para crear nuevos útiles y herramientas en el sistema.</p> <p>Para llevar a cabo una entrada, anteriormente debe haberse creado al menos un área de responsabilidad material y una unidad de medida.</p> <p>El formulario de entrada no puede contener campos vacíos.</p>	
<p><b>Pasos de ejecución:</b></p> <p>El VA1 realiza la entrada de útiles y herramientas en cualquiera de las áreas registradas en el sistema.</p> <p>Los JCC1 y las AC1 realizan la entrada de útiles y herramientas sólo en las áreas de sus respectivos centros de costo.</p>	
<b>Resultados esperados:</b> Los útiles y herramientas son creados en el área de responsabilidad material especificada durante el proceso de entrada, mostrando sus respectivos datos y habilitando las diferentes acciones que se pueden realizar sobre los mismos.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 3.31: Prueba de aceptación # 5

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU5_P5	<b>Historia de usuario:</b> 5
<b>Nombre:</b> Registrar sobrante.	
<b>Descripción:</b> Prueba para la funcionalidad Registrar sobrante.	
<p><b>Condiciones de ejecución:</b></p> <p>El VA1 y los JCC1 son los únicos que tienen los permisos para registrar sobrantes en el sistema.</p> <p>Para llevar a cabo el registro exitoso de un sobrante, es necesario que al menos exista un área de responsabilidad material y una unidad de medida registrados en el sistema.</p> <p>El formulario de entrada no puede contener campos vacíos.</p>	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<p><b>Pasos de ejecución:</b>                  El VA1 realiza el registro de sobrantes en cualquiera de las áreas registradas en el sistema.                  Los JCC1 realizan el registro del sobrantes sólo en las áreas de sus respectivos centros de costo.</p>
<p><b>Resultados esperados:</b> Los sobrantes son añadidos al registro de útiles y herramientas del área especificada, mostrando sus respectivos datos y posibilitando las diferentes acciones que se pueden realizar sobre los mismos. Además es generado el Anexo I en formato PDF y tipo de hoja carta.</p>
<p><b>Evaluación de la prueba:</b> Satisfactorio.</p>

Tabla 3.32: Prueba de aceptación # 6

Caso de prueba de aceptación	
<b>Código:</b> HU6_P6	<b>Historia de usuario:</b> 6
<b>Nombre:</b> Solicitar baja.	
<b>Descripción:</b> Prueba para la funcionalidad Solicitar baja.	
<p><b>Condiciones de ejecución:</b>                  Los JCC1 y RARM1 son los únicos que tienen los permisos para solicitar bajas de útiles y herramientas en el sistema.                  La cantidad especificada en el formulario de solicitud de baja, debe ser mayor que cero y menor que la cantidad total del útil en su respectiva área.                  El sistema debe tener registrada al menos una causa de baja.                  El formulario de solicitud no puede contener campos vacíos.</p>	
<p><b>Pasos de ejecución:</b>                  Los JCC1 y RARM1 realizan correctamente la solicitud de baja de útiles y herramientas de sus respectivas áreas.                  El VA1 es notificado sobre la solicitud de baja realizada.</p>	
<b>Resultados esperados:</b> La solicitud de baja es realizada y queda pendiente a ser tramitada por el VA1 .	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 3.33: Prueba de aceptación # 7

Caso de prueba de aceptación	
<b>Código:</b> HU7_P7	<b>Historia de usuario:</b> 7
<b>Nombre:</b> Movimiento interno.	
<b>Descripción:</b> Prueba para la funcionalidad Movimiento interno.	
<p><b>Condiciones de ejecución:</b>                  El VA1, los JCC1 y RARM1 son los únicos que tienen permisos para realizar movimientos internos en el sistema.</p>	



Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<p>La cantidad especificada en el formulario de movimiento interno, debe ser mayor que cero y menor que la cantidad total del útil en su respectiva área.</p> <p>Además, el sistema debe tener registrada al menos un área distinta del área que constituye el destino del movimiento interno.</p> <p>El formulario de movimiento interno no puede contener campos vacíos.</p>
<p><b>Pasos de ejecución:</b></p> <p>Los usuarios autorizados acceden a la realización de los movimientos internos en sus respectivas áreas de responsabilidad material.</p>
<p><b>Resultados esperados:</b> El útil o herramienta involucrado en el movimiento interno disminuye su existencia en el área origen y la aumenta en el área destino, el valor que modifica dichas existencias es el especificando en el campo correspondiente a la cantidad en el formulario de movimientos internos.</p> <p>Además es generado el Anexo IV en formato PDF y tipo de hoja carta.</p>
<p><b>Evaluación de la prueba:</b> Satisfactorio.</p>

Tabla 3.34: Prueba de aceptación # 8

Caso de prueba de aceptación	
<b>Código:</b> HU8_P8	<b>Historia de usuario:</b> 8
<b>Nombre:</b> Movimiento externo.	
<b>Descripción:</b> Prueba para la funcionalidad Movimiento externo.	
<p><b>Condiciones de ejecución:</b></p> <p>El VA1, los JCC1 y RARM1 son los únicos que tienen permisos para realizar movimientos externos en el sistema. La cantidad especificada en el formulario de movimiento externo, debe ser mayor que cero y menor que la cantidad total del útil en su respectiva área.</p> <p>Además, el sistema debe tener registrado al menos un centro de costo y área distinta al centro de costo y área que constituyen el destino del movimiento externo.</p> <p>El formulario de movimiento externo no puede contener campos vacíos.</p>	
<p><b>Pasos de ejecución:</b></p> <p>Los usuarios autorizados acceden a la realización de los movimientos externos en sus respectivas áreas de responsabilidad material.</p>	
<p><b>Resultados esperados:</b> El útil o herramienta involucrado en el movimiento externo disminuye su existencia en el área origen y la aumenta en el área destino, el valor que modifica dichas existencias es el especificando en el campo correspondiente a la cantidad en el formulario de movimientos externos.</p> <p>Además es generado el Anexo V en formato PDF y tipo de hoja carta.</p>	
<p><b>Evaluación de la prueba:</b> Satisfactorio.</p>	

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

Tabla 3.35: Prueba de aceptación # 9

Caso de prueba de aceptación	
<b>Código:</b> HU9_P9	<b>Historia de usuario:</b> 9
<b>Nombre:</b> Crear perfil.	
<b>Descripción:</b> Prueba para la funcionalidad Crear perfil.	
<b>Condiciones de ejecución:</b> El VA1 es el único que tiene los permisos para crear un perfil. El sistema debe tener registrado al menos un área de responsabilidad material. El formulario crear perfil no puede contener campos vacíos.	
<b>Pasos de ejecución:</b> El VA1 accede a la creación de un perfil para un determinado usuario.	
<b>Resultados esperados:</b> El nuevo usuario es activado en el sistema y a partir de ese momento puede acceder a su respectiva área y ejecutar las funcionalidades definidas según su rol.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 3.36: Prueba de aceptación # 10

Caso de prueba de aceptación	
<b>Código:</b> HU10_P10	<b>Historia de usuario:</b> 10
<b>Nombre:</b> Crear causa de baja.	
<b>Descripción:</b> Prueba para la funcionalidad Crear causa de baja.	
<b>Condiciones de ejecución:</b> El VA1 es el único que tiene los permisos para crear una nueva causa de baja. El formulario crear causa de baja no puede contener campos vacíos.	
<b>Pasos de ejecución:</b> El VA1 accede a la creación de una nueva causa de baja.	
<b>Resultados esperados:</b> La nueva causa de baja es creada y comienza a estar disponible para ser seleccionada en los proceso de solicitudes y aprobación de bajas de útiles y herramientas.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

El resto de las pruebas de aceptación se encuentran en el Anexo 3.

A partir de la aplicación de los casos de pruebas se identificaron una serie de no conformidades, lo que significó que algunas funcionalidades no se cumplieron satisfactoriamente. En la primera iteración se obtuvo un total de 4 no conformidades de las cuales 3 fueron resueltas y una no procedía, por lo que quedó pendiente para la próxima iteración. Las no conformidades encontradas se clasifican en:

Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

**Funcionalidad:** Al intentar autenticar un usuario en el sistema con sus credenciales correctas, la respuesta no es la esperada, pues no realiza la comprobación a través del servicio de autenticación de la UCI.

**Seguridad:** Una vez autenticado el usuario correctamente y comprobado el almacenamiento de la información en la base de datos del sistema, se identifica que el mismo almacena la contraseña del usuario en texto plano.

**Validación:**

1. Al intentar crear nuevos centros de costo y áreas de responsabilidad material, durante el proceso de entrada de datos, el sistema permitió introducir campos no válidos.
2. Al intentar crear nuevas unidades de medidas y causas de bajas, durante el proceso de entrada de datos, el sistema permitió introducir campos no válidos.

En la segunda iteración se obtuvo 1 no conformidad nueva y 1 pendiente de la iteración anterior, las mismas fueron resultas, clasificándose en:

**Validación:** Al intentar crear nuevas unidades de medidas y causas de bajas, durante el proceso de entrada de datos, el sistema permitió introducir campos no válidos.

**Funcionalidad:** Al intentar generar los Modelos (1, 3 y 4) en formato PDF, la respuesta no es la esperada, pues el texto y las tablas contenidas en los modelos antes mencionados, se superponen afectado el formato del documento.

La tercera iteración arrojó resultados satisfactorios, ya que se resolvieron las no conformidades identificadas en las iteraciones anteriores y no se encontraron nuevas. La figura 3.4 muestra un gráfico con los resultados obtenidos durante la aplicación de las pruebas de aceptación:

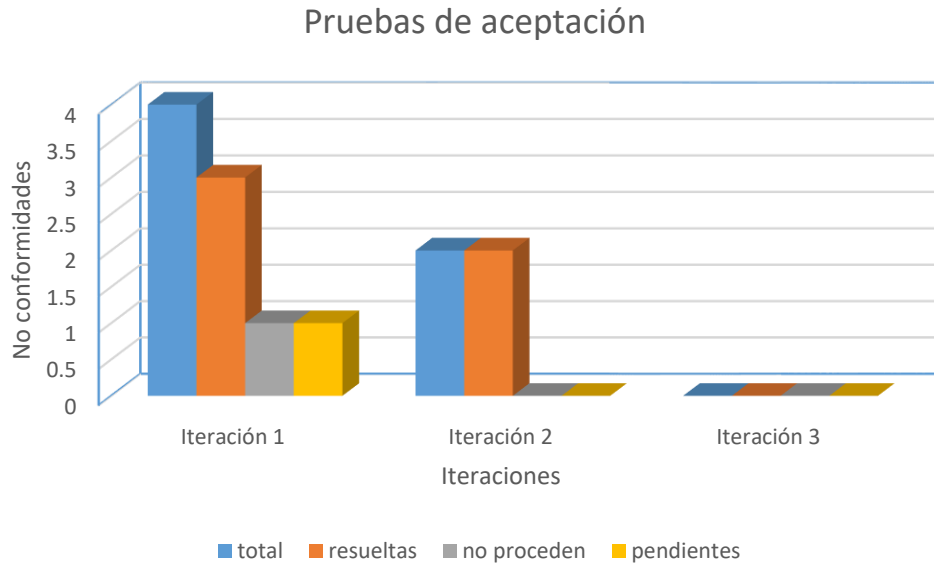


Figura 3.4: Pruebas de aceptación. (Elaboración propia)

### 3.6.3 Pruebas no funcionales

Las pruebas no funcionales son aquellas cuyo objetivo es la verificación de un requisito no funcional que especifica criterios que pueden usarse para juzgar el comportamiento del sistema. Podemos clasificar las pruebas no funcionales según el tipo de requisito no funcional que abarcan. (Mera Paz, 2016)

**Pruebas de interfaz:** La tabla 3.37 muestra un caso de prueba no funcional para un requisito de interfaz definido para la aplicación.

Tabla 3.37: Caso de prueba no funcional de interfaz.

Caso de prueba no funcional
<b>Tipo de prueba:</b> Prueba de interfaz
<b>Requisito no funcional:</b> Las interfaces del sistema deben adaptarse a dispositivos móviles con distintas longitudes y resoluciones de pantallas.
<b>Escenario de prueba:</b> El acceso a la aplicación se realizó mediante diferentes dispositivos móviles.
<b>Resultados:</b> Las interfaces de la aplicación se adaptan a las diferentes resoluciones de pantallas de los dispositivos móviles desde los que se accedió: teléfono móvil inteligente con pantalla de 5 pulgadas y resolución 1280x720, tableta inteligente con pantalla de 7.9 pulgadas y resolución 2048 X 1536. Interfaz vista desde un teléfono móvil ver figura 3.5.

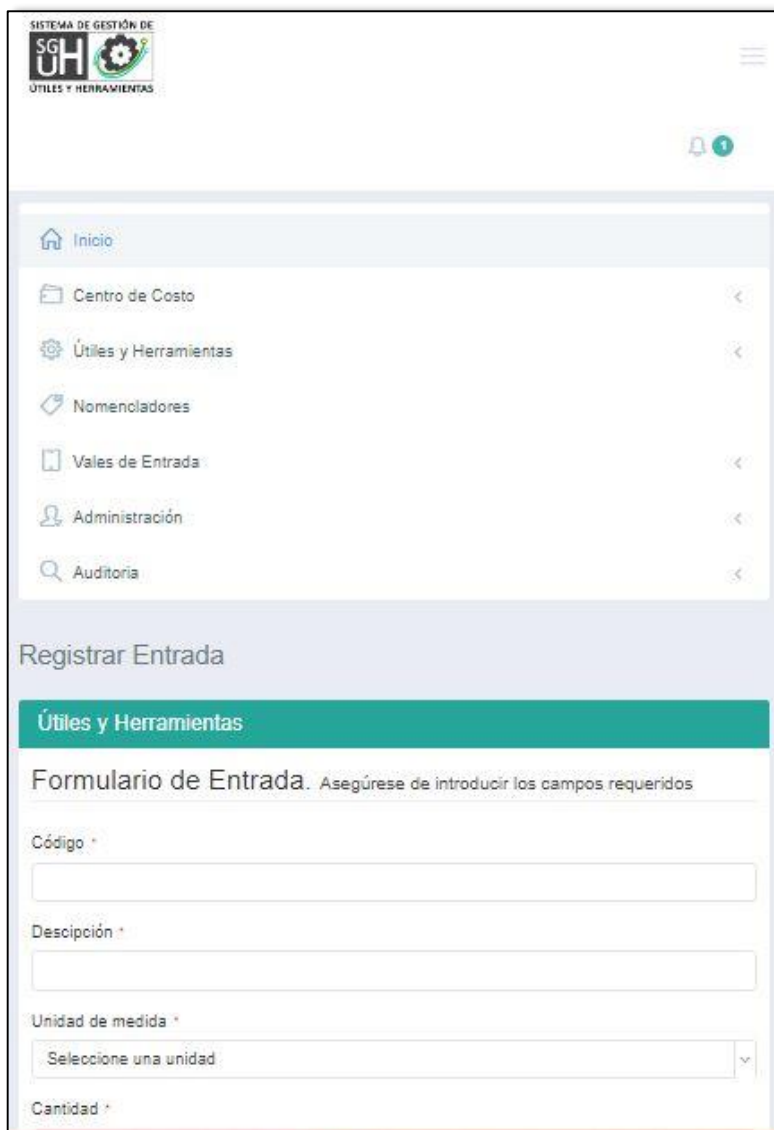


Figura 3.5: Interfaz de la aplicación desde un teléfono móvil. (Elaboración propia)

**Evaluación:** Satisfactoria.

**Prueba de rendimiento:** Ver caso de prueba en la tabla 3.38.

Tabla 3.38: Caso de prueba de rendimiento.

<b>Caso de prueba no funcional</b>
<b>Tipo de prueba:</b> Prueba de rendimiento
<b>Requisito no funcional:</b> El tiempo de respuesta de las funcionalidades de la aplicación debe ser el mínimo posible.

<b>Escenario de prueba:</b> Durante la aplicación de las pruebas unitarias se implementaron 11 casos de pruebas funcionales, las cuales se ejecutaron con diferentes juegos de datos.
<b>Resultados:</b> La aplicación ejecutó las funcionalidades programadas durante las pruebas unitarias en un tiempo de 0.339 segundos. Ver figura 3.3.
<b>Evaluación:</b> Satisfactoria.

### Pruebas de compatibilidad

Para el requisito no funcional referido al funcionamiento de la aplicación en los navegadores más usados para acceder a la web, se describen los resultados de las pruebas realizadas en la tabla 3.39.

Tabla 3.39: Compatibilidad con navegadores web.

	Google Chrome	Mozilla Firefox	Internet Explorer	Opera	Safari
<b>Versiones</b>	>=45	>=38	>=10	>=30	>=9

### Pruebas de usabilidad

La usabilidad de la aplicación se encuentra validada por la utilización de Metronic 5.1 como plantilla de panel de respuesta basada en el marco de Bootstrap de Twitter. El diseño de interfaces y acceso a las funcionalidades cumplen con las cinco reglas definidas para la usabilidad de sitios web establecidas por la norma internacional ISO 9241-1 referente a la “Guía de usabilidad”:

1. Utilidad: Cada elemento de la web debe ser útil, se deben evitar animaciones e imágenes innecesarias.
2. Facilidad de uso: Para llegar a cualquier sección el usuario debe hacer un máximo de 2 clicks.
3. Rapidez: Evitar fotos gigantes, animaciones y vídeos, todo ello puede retrasar el tiempo de carga de la web.
4. Intuitivo: Que el usuario sepa dónde debe hacer click, el menú debe ser claro y conciso.
5. Eficiencia: Que cada botón o enlace cumpla su función. Se debe prestar especial atención a los enlaces rotos (links a páginas que ya no funcionan).

### Pruebas de seguridad

La tabla 3.40 describe el caso de prueba de seguridad realizada a la aplicación:

Tabla 3.40: Caso de prueba de seguridad.

<b>Caso de prueba no funcional</b>
<b>Tipo de prueba:</b> Prueba de seguridad

### Capítulo 3: Diseño, implementación y pruebas del sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2

<b>Requisito no funcional:</b> Los usuarios solo pueden acceder a las funcionalidades definidas por su rol.
<b>Escenario de prueba:</b> El menú para un usuario RARM1 del sistema no muestra funcionalidades como crear centro de costo, crear área o consultar evidencias, funcionalidades disponibles solo para el VA1. Un usuario introduce directamente en el navegador la dirección o ruta correspondiente a una funcionalidad no definida por su rol. Usuario: gremedios, rol: RARM1, rutas a la que intenta acceder: /crearCC/, /crearARM/, /registros/, /evidencias/.
<b>Resultados:</b> En todos los casos la aplicación muestra una interfaz personalizada de Error 403 con un mensaje notificando al usuario que no tiene permisos para acceder a la funcionalidad.
<b>Evaluación:</b> Satisfactoria.

#### Pruebas de concurrencia

Las pruebas de concurrencia se realizaron utilizando la herramienta apache-jmeter-2.10 para la simulación de múltiples peticiones y usuarios concurrentes a aplicaciones web. La tabla 3.41 muestra el caso de prueba de concurrencia realizado.

Tabla 3.41: Caso de prueba de concurrencia.

Caso de prueba no funcional
<b>Tipo de prueba:</b> Prueba de concurrencia
<b>Requisito no funcional:</b> El tiempo de respuesta ante peticiones de un gran número de usuarios concurrentes debe ser el menor posible.
<b>Escenario de prueba:</b> La imagen muestra la configuración de la herramienta apache-jmeter-2.10 para simular la conexión de 50 usuarios (Number of Threads users), realizando un ciclo (Loop Count) de 100 peticiones.

**Thread Group**

Name: Thread Group

Comments:

Action to be taken after a Sampler error

Continue

Thread Properties

Number of Threads (users): 50

Ramp-Up Period (in seconds): 1

Loop Count:  Forever 100

Delay Thread creation until needed

Scheduler

Figura 3.6: Parámetros de configuración de apache-jmeter. (Elaboración propia)

**Resultados:** La aplicación no falla y se mantiene estable durante la aplicación de la prueba, respondiendo en un tiempo de 29 milisegundos. La figura 3.7 muestra el gráfico con los datos registrados por la herramienta apache-jmeter-2.10.



Figura 3.7: Resultado gráfico de prueba de concurrencia. (Elaboración propia)

**Evaluación:** Satisfactoria.



### **Conclusiones del capítulo**

Se selecciona la arquitectura Cliente-Servidor como la más acertada para el trabajo con la aplicación web desarrollada. El framework seleccionado implementa el patrón Modelo Vista Plantilla, por lo que se selecciona el mismo para el desarrollo del sistema. Mediante el diseño del modelo de datos del sistema se establece el punto de partida para la implementación de la aplicación. La elaboración de las tarjetas CRC permitieron identificar, organizar y establecer las relaciones entre las clases del sistema para su implementación. Las tareas de ingeniería identificaron y especificaron las tareas de los programadores para el desarrollo del sistema. La aplicación superó con éxito las pruebas funcionales y no funcionales aplicadas durante el proceso de validación. Por tanto, la culminación del presente capítulo representa la garantía, calidad y satisfacción del cliente y el equipo de desarrollo con el producto final.

## **Conclusiones generales**

Una vez culminado el desarrollo del sistema se arribaron las siguientes conclusiones:

El estudio del sistema de información del proceso de gestión de útiles y herramientas y la normativa que lo regula, permitieron identificar aspectos generales para la implementación del sistema.

El análisis en cuanto al diseño y funcionamiento de los sistemas de información aplicados al proceso de gestión de útiles y herramientas permitió identificar las principales funcionalidades y características de la aplicación desarrollada. Se determinó la necesidad de desarrollar un sistema de gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2, que cumpla con la Norma 40 del 2007 del Ministerio de Educación Superior.

El análisis de las diferentes herramientas y selección de la metodología de desarrollo de software, permitieron identificar la metodología XP, el lenguaje Python y el marco de trabajo Django como los que más se ajustan a las características del proceso de desarrollo.

Como resultado de la planificación, el diseño y las pruebas de validación, se obtuvo un sistema que agiliza la gestión de información en el proceso de gestión de útiles y herramientas de la Facultad 2, que cumple con la Norma 40 del 2007 del Ministerio de Educación Superior.

## Recomendaciones

Los resultados obtenidos en este trabajo sugieren una serie de recomendaciones para seguir perfeccionando la gestión de información para el proceso de gestión de útiles y herramientas de la Facultad 2. Por lo que se recomienda:

- El despliegue del sistema, para que el proceso de gestión de información de útiles y herramientas en el área se agilice y se realice de manera informatizada.
- Analizar la posibilidad de extender el sistema a otras facultades.

## Referencias bibliográficas

- ALARCÓN, V.F., 2006. *Desarrollo de sistemas de información: una metodología basada en el modelado*. S.I.: Univ. Politèc. de Catalunya.
- AMENDOLA, L.J., 2011. *Gestión integral de activos físicos*. S.I.: PMM Institute for Learning.
- ASSETS: Sistema de Gestión Integral. [en línea], 2019. [Consulta: 7 mayo 2019]. Disponible en: <http://www.assets.co.cu/utiles.asp>.
- AUQUILLA, G. y JOSÉ, M., 2018. *Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3*. B.S. thesis. S.I.: Riobamba.
- BLANCO ENCINOSA, L.J., 2002. *Reseña histórica de la informática en cuba hasta el 2002*. [en línea]. 2002. S.I.: s.n. Disponible en: [https://intranet.uci.cu/sites/default/files/historia/historia-doc/Resena\\_historica\\_de\\_la\\_informatica\\_en\\_cuba\\_hasta\\_el\\_2002.pdf](https://intranet.uci.cu/sites/default/files/historia/historia-doc/Resena_historica_de_la_informatica_en_cuba_hasta_el_2002.pdf).
- CANEIRO, K.Z. y VÉLIZ, E.C., 2015. Modelo integral para la gestión de información económica por unidades de negocios en una empresa biotecnológica cubana. *Revista Bioprocesos*, vol. 1, no. 1.
- CHUN, W., 2001. *Core python programming*. S.I.: Prentice Hall Professional.
- DELGADO, N. de las M.G., 2014. Estudio sobre la gestión del costo en las empresas cubanas. *Revista Cubana de Contabilidad y Finanzas. COFIN HABANA*, no. 1, pp. 36–41.
- ENRICH CARDONA, R., 2013. *Implantación de un sistema ERP SAP en una empresa*. B.S. thesis. S.I.: Universitat Politècnica de Catalunya.
- FERNÁNDEZ, A. y LUIS, J., 2018. Desarrollo de un Sistema de Información haciendo uso de la Metodología XP para la Gestión de Ventas, Compras y Almacén de la Empresa Agro Market Peru S.A.C. *Universidad Nacional Pedro Ruiz Gallo* [en línea], [Consulta: 7 mayo 2019]. Disponible en: <http://repositorio.unprg.edu.pe/handle/UNPRG/1948>.
- GARCÍA CALDERÓN, J.A., 2018. *Entrevista al Jefe de Departamento de Desarrollo del Proyecto XABAL-KAINOS*. 11 enero 2018. S.I.: s.n. 837-3166
- GÉNOVA, G., FUENTES, J. y VALIENTE, M., 2006. Evaluación comparativa de herramientas CASE para UML desde el punto de vista notacional. *Novática*, vol. 181, pp. 59–64.
- GUTIÉRREZ, J., 2016. ¿Qué es un framework web? [En línea]. *Consulta en: Octubre*,
- HERNÁNDEZ CÁRDENAS, L., 2013. *Mercado de Datos para el proceso de inventario del Sistema de Gestión Contable VERSAT Sarasola*. PhD Thesis. S.I.: Universidad Central “Marta Abreu” de Las Villas.
- HERNÁNDEZ LEÓN, R.A. y COELLO GONZÁLEZ, Z., 2008. El paradigma cuantitativo de la investigación científica. S.I.: e-libro, Corp.
- ISLAM, Q.N., 2015. *Mastering PyCharm*. S.I.: Packt Publishing Ltd.
- JOSKOWICZ, J., 2008. Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*, vol. 22.

- JUÁREZ, P. y IVONNE, I., 2018. Sistema gestor de base de datos. *REPOSITORIO REMERI*,
- LARMAN, C., 2003. *UML y Patrones*. S.I.: Pearson Educación Madrid.
- LETELIER, P., 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- LETELIER, P. y PENADÉS, M.C., 2012. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- MARINI, E., 2012. El Modelo Cliente/Servidor. *Recuperado el*, vol. 5.
- MAZA, M.A.S., 2012. *Javascript*. S.I.: Innovación Y Cualificación.
- MERA PAZ, J.A., 2016. Análisis del proceso de pruebas de calidad de software.
- MINISTERIO DE EDUCACIÓN SUPERIOR, M., 2007. *NORMA 40 Sobre los procesos de registro y control de los útiles, herramientas, enseres menores y textos en las entidades subordinadas al Ministerio de Educación Superior*. 2007. S.I.: s.n.
- MINISTERIO DE JUSTICIA, 2009. *Decreto Ley No.-281 Del sistema de información del gobierno*. 2009. S.I.: s.n.
- MOLINA CORONADO, B., 2015. *Integración de un proxy inverso NGINX con un panel de control Virtualmin para crear una plataforma de servicios de hospedaje Web*. PhD Thesis. S.I.: s.n.
- MOLINA, J.C. y MORENO, M.E.T., 2010. Análisis de requerimientos usando BPMN. *Revista Colombiana de Computación*, vol. 11, no. 1, pp. 85–97.
- MOMJIAN, B., 2001. *PostgreSQL: Introduction and concepts*. S.I.: Addison-Wesley New York.
- MOSS, G.R., 2013. *Working with openERP*. S.I.: Packt Publishing Ltd.
- OCHOA REYES, A.J., ORELLANA GARCÍA, A., SÁNCHEZ CORALES, Y. y DAVILA HERNÁNDEZ, F., 2014. Componente web para el análisis de información clínica usando la técnica de Minería de Datos por agrupamiento. *Revista Cubana de Informática Médica*, vol. 6, no. 1, pp. 5–16.
- OPENERP SPAIN, O., 2010. *OpenERP Spain: Funcionalidades de los módulos oficiales*. S.I.: URL:<  
<http://www.openerpspain.com/Oficiales/>> [última consulta: 20/1/2011].
- REMON, M.T., 2014. *Diseño web con HTML5 y CSS3*. S.I.: Editorial Macro.
- REYNOSO, C. y KICILLOF, N., 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. S.I.: s.n.
- RÍOS, J.M., MORA, N.L., ORDÓÑEZ, M.Z. y SOJOS, E.L., 2016. Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. *Revista Latinoamericana de Ingeniería de Software*, vol. 4, no. 4, pp. 201–207.
- RIVADENEIRA MOLINA, S.G., 2012. Capítulo 3 Definición de requerimientos - Metodología de Gestión de Requerimientos. [en línea]. [Consulta: 24 mayo 2019]. Disponible en: <https://sites.google.com/site/metodologiareq/capitulo-iii>.

- ROBLES, G. y FERRER, J., 2006. Programación eXtrema y Software Libre. *Universidad Politécnica de Madrid. España*,
- RODRÍGUEZ ÁLVAREZ, A.N., 2018. *Sistema informático para la gestión de la información económica contable y financiera*. PhD Thesis. S.I.: Universidad Tecnológica de la Habana José Antonio Echeverría.
- ROMERO, L.M., 2009. Funcionalidades Y Principales Opciones Del Módulo De Planificación Del Software Integrado Versat Sarasola. *Observatorio de la Economía Latinoamericana*, no. 117.
- SILBERSCHATZ, A., KORTH, H.F., SUDARSHAN, S., PÉREZ, F.S., SANTIAGO, A.I. y SÁNCHEZ, A.V., 2002. Fundamentos de bases de datos.
- TADAY, T. y ALEXANDRA, N., 2018. *Análisis de las plataformas open source: nginx y lighttpd aplicado a la implementación de streaming mediante estándares de calidad en la Universidad Nacional de Chimborazo*. B.S. thesis. S.I.: Universidad Nacional de Chimborazo, 2018.
- THEMES, K., 2015. Documentation - Metronic Admin Dashboard Template. [en línea]. [Consulta: 7 mayo 2019]. Disponible en: [http://filamentbunker.com/v4.1.0/\\_documentation/admin/](http://filamentbunker.com/v4.1.0/_documentation/admin/).
- VAN LANCKER, L., 2012. *HTML5: Los fundamentos del lenguaje*. S.I.: Ediciones ENI.
- VENTURA, T., 2013. *Modelamiento, diseño de procesos bajo tecnología BPMN y propuesta de mejora en la empresa COCEBET SA*. B.S. thesis. S.I.: QUITO: 2013.
- WHITE, S. y MIERS, D., 2009. Guía de referencia y modelado BPMN. Comprendiendo y utilizando BPMN. *Editorial Future Estrategies Inc. Florida USA*.

## Bibliografía

- ALARCÓN, V.F., 2006. *Desarrollo de sistemas de información: una metodología basada en el modelado*. S.I.: Univ. Politèc. de Catalunya.
- ALEXEIS JOEL, O.R., ARTURO, O.G., MSC YOVANNYS, S.C. y FRANK, D.H., 2012. SLD070 COMPONENTE WEB PARA EL ANÁLISIS DE INFORMACIÓN CLÍNICA USANDO LA TÉCNICA DE MINERÍA DE DATOS AGRUPAMIENTO. *Informática Salud 2013*. S.I.: s.n.,
- ALONSO-ARÉVALO, J., 2007. Gestión de la Información, gestión de contenidos y conocimiento. [en línea]. S.I.: s.n., [Consulta: 7 mayo 2019]. Disponible en: <http://eprints.rclis.org/11273/>.
- AMENDOLA, L.J., 2011. *Gestión integral de activos físicos*. S.I.: PMM Institute for Learning.
- ASSETS: Sistema de Gestión Integral. [en línea], 2019a. [Consulta: 7 mayo 2019]. Disponible en: <http://www.assets.co.cu/assets.asp>.
- ASSETS: Sistema de Gestión Integral. [en línea], 2019b. [Consulta: 7 mayo 2019]. Disponible en: <http://www.assets.co.cu/utiles.asp>.
- AUQUILLA, G. y JOSÉ, M., 2018. *Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3*. B.S. thesis. S.I.: Riobamba.
- BANDAR, M.A., [sin fecha]. Zotero: A bibliographic assistant to researcher - ProQuest. [en línea]. [Consulta: 7 mayo 2019]. Disponible en: <https://search.proquest.com/openview/fb0817246568f10cbd6f3057672a6f1b/1?pq-origsite=gscholar&cbl=226473>.
- BECK, K. y GAMMA, E., 2000. *Extreme programming explained: embrace change*. S.I.: addison-wesley professional.
- BLANCO ENCINOSA, L.J., 2002. *Reseña histórica de la informática en cuba hasta el 2002*. [en línea]. 2002. S.I.: s.n. Disponible en: [https://intranet.uci.cu/sites/default/files/historia/historia-doc/Resena\\_historica\\_de\\_la\\_informatica\\_en\\_cuba\\_hasta\\_el\\_2002.pdf](https://intranet.uci.cu/sites/default/files/historia/historia-doc/Resena_historica_de_la_informatica_en_cuba_hasta_el_2002.pdf).
- CANEIRO, K.Z. y VÉLIZ, E.C., 2015. Modelo integral para la gestión de información económica por unidades de negocios en una empresa biotecnológica cubana. *Revista Bioprocesos*, vol. 1, no. 1.
- CHUN, W., 2001. *Core python programming*. S.I.: Prentice Hall Professional.
- CONDORI AYALA, J.L., 2012. Python-Django Framework de desarrollo web para perfeccionistas Basado en el Modelo MTV. *Revista de Información, Tecnología y Sociedad*, pp. 36.

- CRAIG, L., 1999. UML y Patrones. Introducción al análisis y diseño orientado a objetos. *Prentice Hall, Hispanoamérica México*,
- DELGADO, N. de las M.G., 2014. Estudio sobre la gestión del costo en las empresas cubanas. *Revista Cubana de Contabilidad y Finanzas. COFIN HABANA*, no. 1, pp. 36–41.
- DÍAZ, I.M.R., DE CASTRO, E.C.D. y CATALUÑA, F.J.R., 2013. *Gestión de precios*. S.l.: Esic Editorial.
- DIJKMAN, R.M., DUMAS, M. y OUYANG, C., 2008. Semantics and analysis of business process models in BPMN. *Information and Software technology*, vol. 50, no. 12, pp. 1281–1294.
- ELMASRI, R.A. y NAVATHE, S.B., 2007. *Fundamentos de sistemas de bases de datos*. S.l.: Addison Wesley, 004.65.
- ENCINOSA, L.J.B., 2011. *La informática en la dirección de empresas*. S.l.: Editorial Félix Varela.
- ENRICH CARDONA, R., 2013. *Implantación de un sistema ERP SAP en una empresa*. B.S. thesis. S.l.: Universitat Politècnica de Catalunya.
- ESQUIVEL-RAMÍREZ, E.M., 2010. OpenERP y Android, D&T Network Solutions SA.
- February 2019 Web Server Survey | Netcraft. [en línea], 2019. [Consulta: 7 mayo 2019]. Disponible en: <https://news.netcraft.com/archives/2019/02/28/february-2019-web-server-survey.html>.
- FERNÁNDEZ, A. y LUIS, J., 2018. Desarrollo de un Sistema de Información haciendo uso de la Metodología XP para la Gestión de Ventas, Compras y Almacén de la Empresa Agro Market Peru S.A.C. *Universidad Nacional Pedro Ruiz Gallo* [en línea], [Consulta: 7 mayo 2019]. Disponible en: <http://repositorio.unprg.edu.pe/handle/UNPRG/1948>.
- FIGUEROA, R.G., SOLÍS, C.J. y CABRERA, A.A., 2008. Metodologías tradicionales vs. metodologías ágiles. *Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación*,
- FREITAS, A.P. y PEREIRA, J.L.M., 2015. Process simulation support in BPM tools: The case of BPMN. ,
- GARCÍA CALDERÓN, J.A., 2018. *Entrevista al Jefe de Departamento de Desarrollo del Proyecto XABAL-KAINOS*. 11 enero 2018. S.l.: s.n. 837-3166
- GARRIDO, J.C., 2004. Arquitectura y diseño de sistemas web modernos. *InforMAS, Revista de Ingeniería Informática del CIIRM*, no. 1.
- GÉNOVA, G., FUENTES, J. y VALIENTE, M., 2006. Evaluación comparativa de herramientas CASE para UML desde el punto de vista notacional. *Novática*, vol. 181, pp. 59–64.
- GUTIÉRREZ, J., 2016. ¿Qué es un framework web? [En línea]. *Consulta en: Octubre*,



- GUTIÉRREZ, J.J., 1/11/2016. ¿Qué es un Framework web? No available [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf),
- HERNÁNDEZ CÁRDENAS, L., 2013. *Mercado de Datos para el proceso de inventario del Sistema de Gestión Contable VERSAT Sarasola*. PhD Thesis. S.I.: Universidad Central “Marta Abreu” de Las Villas.
- HERNÁNDEZ LEÓN, R.A. y COELLO GONZÁLEZ, Z., 2008. El paradigma cuantitativo de la investigación científica. S.I.: e-libro, Corp.
- ISLAM, Q.N., 2015. *Mastering PyCharm*. S.I.: Packt Publishing Ltd.
- JOSKOWICZ, J., 2008. Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*, vol. 22.
- JUÁREZ, P. y IVONNE, I., 2018. Sistema gestor de base de datos. *REPOSITORIO REMERI*,
- KEENTHEMES, M., 2019. Premium Bootstrap Admin Themes. *Keenthemes* [en línea]. [Consulta: 7 mayo 2019]. Disponible en: <https://keenthemes.com/>.
- KORTH, H.F., SILBERSCHATZ, A., SUDARSHAN, S. y PÉREZ, F.S., 1987. *Fundamentos de bases de datos*. S.I.: McGraw-Hill. QA 76.9. D32. K6718.
- LARMAN, C., 2003. *UML y Patrones*. S.I.: Pearson Educación Madrid.
- LETELIER, P., 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- LETELIER, P. y PENADÉS, M.C., 2012. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- LUTZ, M., 2001. *Programming python*. S.I.: O'Reilly Media, Inc.
- LUTZ, M., 2010. *Programming Python: powerful object-oriented programming*. S.I.: O'Reilly Media, Inc.
- LUTZ, M., 2013. *Learning python: Powerful object-oriented programming*. S.I.: O'Reilly Media, Inc.
- MAGAL, S.R. y WORD, J., 2011. *Integrated business processes with ERP systems*. S.I.: Wiley Publishing.
- MARINI, E., 2012. El Modelo Cliente/Servidor. *Recuperado el*, vol. 5.
- MAZA, M.A.S., 2012. *Javascript*. S.I.: Innovación Y Cualificación.
- MERA PAZ, J.A., 2016. Análisis del proceso de pruebas de calidad de software.
- MINISTERIO DE EDUCACIÓN SUPERIOR, M., 2007. *NORMA 40 Sobre los procesos de registro y control de los útiles, herramientas, enseres menores y textos en las entidades subordinadas al Ministerio de Educación Superior*. 2007. S.I.: s.n.

- Ministerio de Finanzas y Precios de la República de Cuba. *Ministerio de Finanzas y Precios* [en línea], [sin fecha]. [Consulta: 7 mayo 2019]. Disponible en: [http://www.mfp.gob.cu/legislacion/disposiciones\\_publicadas.php](http://www.mfp.gob.cu/legislacion/disposiciones_publicadas.php).
- MINISTERIO DE JUSTICIA, 2009. *Decreto Ley No.-281 Del sistema de información del gobierno*. 2009. S.l.: s.n.
- MOLINA CORONADO, B., 2015. *Integración de un proxy inverso NGINX con un panel de control Virtualmin para crear una plataforma de servicios de hospedaje Web*. PhD Thesis. S.l.: s.n.
- MOLINA, J.C. y MORENO, M.E.T., 2010. Análisis de requerimientos usando BPMN. *Revista Colombiana de Computación*, vol. 11, no. 1, pp. 85–97.
- MOLPECERES, A., 2002. Procesos de desarrollo: RUP, XP y FDD. *Recuperado de [https://uvirtual.unet.edu.ve/pluginfile.php/270296/mod\\_resource/content/1/cualxpfdrrup.PDF](https://uvirtual.unet.edu.ve/pluginfile.php/270296/mod_resource/content/1/cualxpfdrrup.PDF)*,
- MOMJIAN, B., 2001. *PostgreSQL: introduction and concepts*. S.l.: Addison-Wesley New York.
- MORENO-MONTES-DE-OCA, I., RODRÍGUEZ-MORFFI, A., SNOECK, M., MORENO-RODRÍGUEZ, R., CASAS-CARDOSO, G. y GONZÁLEZ-GONZÁLEZ, L., 2014. Directrices prácticas y métricas de calidad en la modelación de procesos de negocio: un caso de estudio. *Revista Cubana de Ciencias Informáticas*, vol. 8, no. 2, pp. 1–18.
- MOSS, G.R., 2013. *Working with openERP*. S.l.: Packt Publishing Ltd.
- OCHOA REYES, A.J., ORELLANA GARCÍA, A., SÁNCHEZ CORALES, Y. y DAVILA HERNÁNDEZ, F., 2014. Componente web para el análisis de información clínica usando la técnica de Minería de Datos por agrupamiento. *Revista Cubana de Informática Médica*, vol. 6, no. 1, pp. 5–16.
- OPENERP SPAIN, O., 2010. *OpenERP Spain: Funcionalidades de los módulos oficiales*. S.l.: URL:< <http://www.openerpspain.com/Oficiales/>> [última consulta: 20/1/2011].
- PÉREZ, O.A., 2011. Cuatro enfoques metodológicos para el desarrollo de Software RUP–MSF–XP–SCRUM. *INVENTUM*, vol. 6, no. 10, pp. 64–78.
- PRESSMAN, R.S., 2005. *Software engineering: a practitioner's approach*. S.l.: Palgrave Macmillan.
- REMIRO, B. y JAVIER, F., 2018. Análisis de requerimientos de una empresa del sector farmacéutico e implantación de mejoras tecnológicas dentro del módulo de ventas y distribución de SAP ERP. ,
- REMON, M.T., 2014. *Diseño web con HTML5 y CSS3*. S.l.: Editorial Macro.

- Resena\_historica\_de\_la\_informatica\_en\_cuba\_hasta\_el\_2002.pdf* [en línea], [sin fecha]. S.l.: s.n. [Consulta: 22 mayo 2019]. Disponible en: [https://intranet.uci.cu/sites/default/files/historia/historia-doc/Resena\\_historica\\_de\\_la\\_informatica\\_en\\_cuba\\_hasta\\_el\\_2002.pdf](https://intranet.uci.cu/sites/default/files/historia/historia-doc/Resena_historica_de_la_informatica_en_cuba_hasta_el_2002.pdf).
- REYNOSO, C. y KICILLOF, N., 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. S.l.: s.n.
- RÍOS, J.M., MORA, N.L., ORDÓÑEZ, M.Z. y SOJOS, E.L., 2016. Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. *Revista Latinoamericana de Ingeniería de Software*, vol. 4, no. 4, pp. 201–207.
- RIVADENEIRA MOLINA, S.G., 2012. Capítulo 3 Definición de requerimientos - Metodología de Gestión de Requerimientos. [en línea]. [Consulta: 24 mayo 2019]. Disponible en: <https://sites.google.com/site/metodologiareq/capitulo-iii>.
- ROBLES, G. y FERRER, J., 2006. Programación eXtrema y Software Libre. *Universidad Politécnica de Madrid. España*,
- RODRÍGUEZ ÁLVAREZ, A.N., 2018. *Sistema informático para la gestión de la información económica contable y financiera*. PhD Thesis. S.l.: Universidad Tecnológica de la Habana José Antonio Echeverría.
- ROMERO, L.M., 2009. Funcionalidades Y Principales Opciones Del Módulo De Planificación Del Software Integrado Versat Sarasola. *Observatorio de la Economía Latinoamericana*, no. 117.
- SÁNCHEZ-RODRÍGUEZ, Á.P., 2010. La gestión de los activos físicos en la función mantenimiento. *Ingeniería Mecánica*, vol. 13, no. 2, pp. 72–78.
- SILBERSCHATZ, A., KORTH, H.F., SUDARSHAN, S., PÉREZ, F.S., SANTIAGO, A.I. y SÁNCHEZ, A.V., 2002. Fundamentos de bases de datos.
- Sistema de Gestión Administrativa. [en línea], [sin fecha]. [Consulta: 14 mayo 2019]. Disponible en: <https://kainos.uci.cu/comun/portada>.
- TADAY, T. y ALEXANDRA, N., 2018. *Análisis de las plataformas open source: nginx y lighttpd aplicado a la implementación de streaming mediante estándares de calidad en la Universidad Nacional de Chimborazo*. B.S. thesis. S.l.: Universidad Nacional de Chimborazo, 2018.
- THEMES, K., 2015. Documentation - Metronic Admin Dashboard Template. [en línea]. [Consulta: 7 mayo 2019]. Disponible en: <http://filamentbunker.com/v4.1.0/documentation/admin/>.
- TORRES, M.A.T., 2005. *Precios unitarios*. S.l.: Univ. J. Autónoma de Tabasco.

VAN LANCKER, L., 2012. *HTML5: Los fundamentos del lenguaje*. S.l.: Ediciones ENI.

VAN ROSSUM, G., 2007. Python Programming Language. *USENIX annual technical conference*. S.l.: s.n., pp. 36.

VENTURA, T., 2013. *Modelamiento, diseño de procesos bajo tecnología BPMN y propuesta de mejora en la empresa COCEBET SA*. B.S. thesis. S.l.: QUITO: 2013.

WHITE, S. y MIERS, D., 2009. Guía de referencia y modelado BPMN. Comprendiendo y utilizando BPMN. *Editorial Future Estrategies Inc. Florida USA*,

WHITE, S.A., 2004. Introduction to BPMN. *Ibm Cooperation*, vol. 2, no. 0, pp. 0.

WHITE, Stephen A. y MIERS, P.-D., 2009. Guía de Referencia y Modelado BPMN. Comprendiendo y utilizando BPMN. Future Strategies Inc. *digital en español, USA*.

## Anexos

### Anexo 1. Modelos y anexos de las Norma 40 del 2007 del Ministerio de Educación Superior (MES).

#### Anexo 1.1 Modelo 4-Acta de Responsabilidad Material

Código	Descripción	U.M.	Existencia actual

Nombre(s) y Apellidos

#### Anexo 1.2 Modelo 2-Acta de entrega

Entidad: Universidad de las Ciencias Informáticas (UCI)

**MODELO 2 - ACTA DE ENTREGA**

Por medio de la presente, estamos entregando al compañero/a \_\_\_\_\_ de la Dependencia- \_\_\_\_\_, las siguientes Herramientas, Utensilios y Similares para su uso y conservación. En caso de causar baja del centro o traslado hacia otra área, debe efectuar la devolución correspondiente al Jefe de Área.

Código	Descripción del útil	Cantidad	Precio CUC	Importe CUC	Precio CUP	Importe CUP

Entrega conforme:

Jefe de Área:

Firma:

Recibe conforme:

Fecha:

Nombre:

Firma:

### Anexo 1.3 Anexo I-Sobrantes

ANEXO I - SOBRANTES			
Número del Centro de Costo:		Nombre del Centro de Costo:	
		Fecha:	

CÓDIGO	DESCRIPCIÓN DEL MEDIO	CANT.	ÁREA DONDE SE ENCUENTRA FISICAMENTE EL MEDIO

Confeccionado por:

Aprobado por:

Notificado a:

### Anexo 1.4 Anexo IV-Movimiento interno

ANEXO IV - MOVIMIENTOS INTERNOS
---------------------------------

Fecha:				
Nombre del Centro de Costo:				
Número del Centro de Costo:				
CÓDIGO	DESCRIPCIÓN DEL MEDIO	CANT	ÁREA DE ORIGEN	ÁREA DE DESTINO

Confeccionado por:

Aprobado por:

Notificado a:

**Anexo 1.5 Anexo V-Movimiento externo**

<b>ANEXO IV - MOVIMIENTOS EXTERNOS</b>				
Fecha:				
Número del Centro de Costo Origen:				
Nombre del Centro de Costo Origen:				
Número del Centro de Costo Destino:				
Nombre del Centro de Costo Destino:				
CÓDIGO	DESCRIPCIÓN DEL MEDIO	CANT	ÁREA DE ORIGEN	ÁREA DE DESTINO

Jefe de Centro de Costo Origen

Nombre y Apellidos:

Firma:

Jefe de Centro de Costo Destino:

Nombre y Apellidos:

Firma:

## Anexo 1.6 Acta de baja

Universidad de las Ciencias Informáticas (UCI)

Fecha:

Nombre del año:

COMISIÓN ENCARGADA DE LA EVALUACIÓN Y PROCESAMIENTO DE BAJAS DE ÚTILES Y HERRAMIENTAS DE LA FACULTAD 2

Acta #:

Asunto a tratar:

Presentes

(Nombres y Apellidos de los presentes)

Siendo las 4 :05 PM del día antes señalado, se reúne la Comisión de Baja de útiles para analizar la posible **baja por Causa de baja** del **Descripción del útil o herramienta**, pertenecientes al Área de Responsabilidad Material **código-nombre de área**, útiles que eran utilizados en - \_\_\_\_\_, para \_\_\_\_\_.

Análisis de las causas y condiciones por las que causa baja el medio, se arriba a la conclusión de que existe o no responsabilidad material sobre la causa de baja del medio por parte de los trabajadores que los utilizaban.

La comisión toma como acuerdo darle **causa de baja** al medio analizado del listado de útiles y herramientas del centro de costo **código-Nombre del centro de costo**. Para que así conste, firman ambos (2) acuerdos todos los miembros de la comisión, expidiendo dos copias de la presente, resguardando una en el Vicedecanato de Administración y Servicios de la Facultad 2.

Acuerdo #: número del acuerdo.

Presentes

Firmas

Siendo la **hora** se da por terminada la reunión de la COMISIÓN ENCARGADA DE LA EVALUACIÓN Y PROCESAMIENTO DE BAJAS DE ÚTILES Y HERRAMIENTAS DE LA FACULTAD 2.



**Anexo 1.7 Modelo 1-Útiles en uso**

Entidad: Universidad de las Cuencas Informáticas (UCI)

MODELO 1: ÚTILES EN USO

Fecha:

Código	Descripción	UM	Entrada	Salida	Existencia	Precio CUC	Importe CUC	Precio CUP	Importe CUP

Área:

Responsable de Área:

Visto bueno:

**Anexo 1.8 Modelo 3-Inventario de Utensilios, Herramientas y Enseres Menores**

Entidad: Universidad de las Cuencas Informáticas (UCI)

MODELO 3: INVENTARIO EN USO DE UTENSILIOS, HERRAMIENTAS Y ENSERES MENORES

Unidad:

Fecha:

Área de Responsabilidad Material:

Responsable:

Anexos

SEGÚN LIBROS							COMPROBACIÓN FÍSICA							
Código	Descripción	Cant	CUC		CUP		Traslados			Bajas por desgastes o roturas	Faltantes	Sobrantes	Existencias	
			Precio	Importe	Precio	Importe	Can	Imp	Cód	Cant	Cant	Cant	Can	Importe

Elaborado por:

Firma:

Aprobado por:

Firma:

Conforme:

Firma:

## Anexo 2. Tarjetas CRC

Tabla A.2.1: Tarjeta CRC # 11

Tarjeta CRC	
<b>Clase:</b> ReporteUHenMovView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la responsable de generar el reporte de útiles y herramientas de útiles en formato PDF.	UtilesEnMovimiento CentroCosto ARM

Tabla A.2.2: Tarjeta CRC # 13

Tarjeta CRC	
<b>Clase:</b> SolicitarBajaView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase tiene la responsabilidad de realizar la solicitud de baja de un útil y herramienta al vicedecano de administración. Al finalizar la solicitud debe enviar una notificación al vicedecano de administración.	Baja CausaBaja Notificaciones User

Tabla A.2.3: Tarjeta CRC # 14

Tarjeta CRC	
<b>Clase:</b> BajaView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de registrar y hacer efectiva la baja de un útil y herramienta. Al finalizar el registro del proceso de baja, tiene la responsabilidad de exportar en formato PDF el Acta de baja.	Baja Notificaciones

Tabla A.2.4: Tarjeta CRC # 15

Tarjeta CRC	
<b>Clase:</b> UnidadMedidaView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la responsable de crear unidades de medida, las unidades de medidas.	UnidadMedida

Tabla A.2.5: Tarjeta CRC # 16

Tarjeta CRC	
<b>Clase:</b> CausaBajaView	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la responsable de crear causas de baja.	CausaBaja

Tabla A.2.6: Tarjeta CRC # 17

Tarjeta CRC	
<b>Clase:</b> Modelo4View	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la responsable de generar el Modelo 4 en formato PDF.	UtilHerramienta ARM

### Anexo 3. Pruebas de aceptación

Tabla A.3.1: Prueba de aceptación # 11

Caso de prueba de aceptación	
<b>Código:</b> HU11_P11	<b>Historia de usuario:</b> 11
<b>Nombre:</b> Crear unidad de medida.	
<b>Descripción:</b> Prueba para la funcionalidad Crear unidad de medida.	
<b>Condiciones de ejecución:</b> El VA1 es el único que tiene los permisos para crear una nueva unidad de medida. El formulario crear unidad de medida no puede contener campos vacíos.	
<b>Pasos de ejecución:</b> El VA1 accede a la creación de una nueva unidad de medida.	
<b>Resultados esperados:</b> La nueva unidad de medida es creada y comienza a estar disponible para ser seleccionada en los proceso de entrada y registro de sobrantes de útiles y herramientas.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla A.3.2: Prueba de aceptación # 12

Caso de prueba de aceptación	
<b>Código:</b> HU12_P12	<b>Historia de usuario:</b> 12
<b>Nombre:</b> Generar Modelos (1, 3 y 4).	
<b>Descripción:</b> Prueba para la funcionalidad Generar Modelos (1, 3 y 4).	
<b>Condiciones de ejecución:</b> El VA1 es el único que tiene los permisos para generar cualquiera de los modelos en todas las áreas registradas en el sistema. Los JCC1 tienen permisos para generar cualquiera de los modelos, en las áreas de sus respectivos centros de costo. Los RARM1 tienen permisos para generar cualquiera de los modelos en sus respectivas áreas.	
<b>Pasos de ejecución:</b> Los usuarios autorizados acceden a la generación de los modelos 1, 3 y 4.	
<b>Resultados esperados:</b> Son generados los modelos 1, 3 y 4 en formato PDF.	

<b>Evaluación de la prueba:</b> Satisfactorio.
--

Tabla A.3.3: Prueba de aceptación # 13

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU13_P13	<b>Historia de usuario:</b> 13
<b>Nombre:</b> Tramitar baja	
<b>Descripción:</b> Prueba para la funcionalidad Tramitar baja	
<b>Condiciones de ejecución:</b> El VA1 es el único que tiene los permisos para tramitar bajas en el sistema. La cantidad del útil o herramienta especificado en el formulario de baja debe ser mayor que cero y menor a la especificada durante el proceso de solicitud de baja. El formulario de baja no puede contener campos vacíos.	
<b>Pasos de ejecución:</b> El VA1 accede a tramitar la baja sobre el útil o herramienta.	
<b>Resultados esperados:</b> El útil o herramienta es dado de baja de su respectiva área y como evidencia del proceso de baja, es generada el Acta de baja en formato PDF y tipo de hoja carta.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla A.3.4: Prueba de aceptación # 14

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU14_P14	<b>Historia de usuario:</b> 14
<b>Nombre:</b> Entrada de útil y herramienta en movimiento.	
<b>Descripción:</b> Prueba para la funcionalidad Entrada de útil y herramienta en movimiento.	
<b>Condiciones de ejecución:</b> El VA1, los JCC1 y las AC1 son los únicos que tienen los permisos para insertar nuevos útiles y herramientas en movimiento. Para llevar a cabo una entrada, anteriormente debe haberse creado al menos un área y una unidad de medida. El formulario de entrada no puede contener campos vacíos.	
<b>Pasos de ejecución:</b> El VA1 realiza la entrada de útiles y herramientas en cualquiera de las áreas registradas en el sistema. Los JCC1 y las AC1 realizan la entrada de útiles y herramientas en movimiento sólo en las áreas de sus respectivos centros de costo.	
<b>Resultados esperados:</b> Los útiles y herramientas son creados en el área especificada durante el proceso de entrada, mostrando sus respectivos. Además, el sistema genera el Modelo 2 en formato PDF y tipo de hoja carta.	

<b>Evaluación de la prueba:</b> Satisfactorio.
--

Tabla A.3.5: Prueba de aceptación # 15

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU15_P15	<b>Historia de usuario:</b> 15
<b>Nombre:</b> Generar reporte de útiles y herramientas en movimiento.	
<b>Descripción:</b> Prueba para la funcionalidad Generar reporte de útiles y herramientas en movimiento.	
<b>Condiciones de ejecución:</b> El VA1 es el único con permiso para generar el reporte de útiles y herramientas en movimiento para todas las áreas del sistema. Los JCC1 sólo pueden generar el reporte de útiles y herramientas en movimiento para las áreas pertenecientes a su respectivo centro de costo. Los RARM1 pueden generar el reporte de útiles y herramientas en movimiento para sus respectivas áreas.	
<b>Pasos de ejecución:</b> Los usuarios autorizados acceden a la generación del reporte de útiles y herramientas en movimiento.	
<b>Resultados esperados:</b> Es generado el reporte de útiles y herramientas en movimiento en formato PDF y tipo de hoja carta.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla A.3.6: Prueba de aceptación # 16

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU16_P16	<b>Historia de usuario:</b> 16
<b>Nombre:</b> Consultar evidencias.	
<b>Descripción:</b> Prueba para la funcionalidad Consultar evidencias.	
<b>Condiciones de ejecución:</b> Los VA1 son los únicos con permisos para consultar las evidencias de los usuarios registrados en el sistema.	
<b>Pasos de ejecución:</b> Los VA1 acceden a la lista de usuarios del sistema y a sus respectivos registros de eventos.	
<b>Resultados esperados:</b> El VA1 consulta el registro de eventos de los usuarios registrados en el sistema, sin poder modificar ningún tipo de información.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

**Anexo 4. Entrevista al Jefe de Departamento de Desarrollo del Proyecto XABAL-KAINOS.**

Tabla A.4.1: Formalización de entrevista

<b>Entrevista al Jefe de Departamento de Desarrollo del Proyecto XABAL-KAINOS</b>	
<b>Entrevistador:</b> Gilberto Remedios Arencibia	<b>Entrevistado:</b> José Alejandro García Calderón
<b>Fecha de la entrevista:</b> 1ro de octubre del 2018.	
<p><b>Cuestionario:</b></p> <p>¿Por qué actualmente no se encuentra en funcionamiento el módulo de gestión de útiles y herramientas del sistema XABAL-KAINOS?</p> <p>¿Cuál es la situación actual en la comunicación de la base de datos del módulo de gestión de útiles y herramientas del sistema XABAL-KAINOS con el sistema Assets?</p> <p>¿El equipo de desarrollo del proyecto tiene conocimiento sobre la Norma 40 del 2007 del MES?</p> <p>¿Actualmente existe alguna línea de proyecto dedicada a solucionar los problemas que presenta el módulo de gestión de útiles y herramientas del sistema XABAL-KAINOS?</p> <p>¿Actualmente existe alguna línea de proyecto dirigida a que el sistema XABAL-KAINOS cumpla con la Norma 40 del 2007 del MES?</p>	