



**Facultad 1**

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

**Título: Sistema de gestión de inventario para la Oficina Nacional de  
Administración Tributaria (ONAT) en Las Tunas.**

**Autor:**

Nelson Alejandro Quiñones Remón.

**Tutor(es):**

Dr.C. Berta Irailis Yanes Watson

Ing. Yilber Agüero Padilla

Ing. Michel Pedrera Suen

La Habana, noviembre de 2021

“Año 63 de la Revolución”

## Índice

Agradecimientos .....	4
Dedicatoria.....	5
Declaración de Autoría .....	7
Resumen .....	8
Palabras Clave .....	8
Introducción .....	9
Capítulo 1. Fundamentos y referentes teórico-metodológicos sobre webs para la gestión de inventarios. .....	14
1.1    Introducción .....	14
1.2    Conceptos fundamentales asociados al dominio del problema .....	14
1.2.1    Gestión Documental y Sistema de gestión documental .....	14
1.2.2    Inventario físico.....	15
1.2.3    Gestión de inventarios y Control de inventario .....	15
1.3    Soluciones Informáticas Similares .....	15
1.3.1    ABC Inventory .....	15
1.3.2    Stockpile de Canvas .....	16
1.3.3    Delivrd .....	16
1.4    Tecnologías de Desarrollo .....	17
1.3.1    Lenguajes de desarrollo.....	17
1.3.2 <i>Frameworks</i> .....	19
1.4    Herramientas de desarrollo.....	20
Visual Paradigm 15.1 .....	20
MySQL 8.0.....	21
NPM 6.0.0.....	21
NetBeans .....	21
Laragon .....	22
1.5    Metodología de Desarrollo de <i>Software</i> .....	22
1.6    Conclusiones parciales del Capítulo .....	23
Capítulo 2. Diseño del sistema de gestión de inventario para la ONAT en Las Tunas. ....	25
2.1    Introducción .....	25

2.2	Modelado del proceso de negocio .....	25
2.2.1	Diagrama de actividad .....	25
2.3	Flujo de Trabajo Requerimientos .....	27
2.3.1	Definición de los Requerimientos Funcionales.....	27
2.3.2	Definición de los Requerimientos no Funcionales.....	29
2.3.3	Actores del Sistema.....	32
2.3.4	Diagrama Entidad-Relación .....	32
2.4	Historias de Usuario.....	33
2.4.1	Planificación .....	34
2.5	Diagramas de Clases de Diseño con Estereotipos Web .....	35
2.6	Diseño Arquitectónico de la aplicación.....	38
2.6.1	Patrón de arquitectura .....	38
2.6.2	Patrones generales de asignación de responsabilidades (GRASP).....	39
2.5	Diagrama de Despliegue .....	39
2.5.1	Arquitectura .....	40
2.6	Conclusiones parciales del Capítulo II. ....	42
Capítulo 3. Implementación y Pruebas del Sistema de gestión de inventario para la Oficina Nacional de Administración Tributaria (ONAT) en Las Tunas.....		43
3.1	Estándares de Codificación .....	43
3.2	Implementación .....	43
3.2.1	Tareas de la Iteración 1 .....	44
3.2.2	Tareas de la Iteración 2 .....	45
3.3	Pruebas de aceptación .....	47
3.4	Conclusiones parciales del Capítulo III. ....	50
Conclusiones .....		51
Recomendaciones .....		52
Bibliografía.....		53
Anexos.....		58

## **Agradecimientos**

Hace tiempo escuché por primera vez la frase: *“Es de bien nacidos ser agradecidos.”* Nunca supe a quién pertenecía, pero sí su significado; saber reconocer y agradecer a quienes te prestaron ayuda, por ese motivo agradezco a:

Emilio Salazar, mi compañero de habitación por 3 largos años y a Juan Pablo León, quienes si no fuera por ellos no estaría aquí hoy.

A mi familia por su apoyo incondicional, sobre todo a mi madre la cual ha velado por mí en todo momento.

Lianet Labrada quien me ayudó cuando no tenía a nadie que aclarara mis dudas.

Mis profesores, aunque su influencia en mi desarrollo como futuro ingeniero no fue la misma tengo que resaltar el apoyo de José Alberto Ponce y Juan Manuel Tur.

A Arturo y Yeline quienes son como si fueran familia debo agradecerles separado porque su ayuda no se puede describir con palabras.

A Serguey por creer en mí y apoyarme en todo momento.

A Niurvis y su secretaria por atender tan bien a mis padres cada vez que ellos preguntaban por mí.

Y por último quienes no pueden faltar y por su ayuda en esta recta final de mi carrera, mis tutores Berta Irailis Yanes Watson, Yilber Agüero Padilla, Michel Pedrera Suen.

A todos ellos y a los que me faltaron, gracias.

## **Dedicatoria**

Este Trabajo de Diploma está dedicado a personas muy especiales para mí que son mi abuela Margarita y mi madre Haydée, a mi querida UCI que cuidó de mi por 6 años y a la cual extrañaré. A aquellas personas que buscarán en este documento una ayuda. Al lector por escogerme entre tantas opciones.

*“Revolución es sentido del momento histórico; es cambiar todo lo que debe ser cambiado; es igualdad y libertad plenas; es ser tratado y tratar a los demás como seres humanos; es emanciparnos por nosotros mismos y con nuestros propios esfuerzos; es desafiar poderosas fuerzas dominantes dentro y fuera del ámbito social y nacional; es defender valores en los que se cree al precio de cualquier sacrificio; es modestia, desinterés, altruismo, solidaridad y heroísmo; es luchar con audacia, inteligencia y realismo; es no mentir jamás ni violar principios éticos; es convicción profunda de que no existe fuerza en el mundo capaz de aplastar la fuerza de la verdad y las ideas. Revolución es unidad, es independencia, es luchar por nuestros sueños de justicia para Cuba y para el mundo, que es la base de nuestro patriotismo, nuestro socialismo y nuestro internacionalismo.”*

Fidel Castro Ruz

1ro de Mayo de 2000

## Declaración de Autoría

El autor del trabajo de diploma con título **“Sistema de gestión de inventario para la Oficina Nacional de Administración Tributaria (ONAT) en Las Tunas”** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Nelson A. Quiñones Remón**

\_\_\_\_\_  
Firma del Autor

**Dr.C. Berta Irailis Yanes  
Watson**

\_\_\_\_\_  
Firma del Tutor

**Ing. Yilber Agüero Padilla**

\_\_\_\_\_  
Firma del Tutor

**Ing. Michel Pedrera Suen**

\_\_\_\_\_  
Firma del Tutor

## **Resumen**

La gestión de inventarios es un punto determinante en el manejo estratégico de toda organización. Los inventarios te permiten conocer la cantidad de mercancía, materia prima, productos almacenados o de bienes que posee tu empresa en un momento determinado el cual pertenece al patrimonio productivo de la misma. Los Sistemas de Gestión de Inventario surgen para acelerar el proceso actual; realizado a mano, de la gestión del inventario el cual es lento y en ocasiones genera costos innecesarios para empresas comerciales; que puede llegar a presentar hasta un 30% de sus ganancias, para la ONAT no resultaría en pérdida monetaria, pero si en tiempo que puede ser utilizado en esferas de más importancia.

## **Palabras Clave**

Sistema, gestión, control, inventario físico.



## Introducción

A través del tiempo el ser humano ha desarrollado diversas Tecnologías de Gestión (TG), las cuales han enfocado su esencia en diferentes aspectos con el único objetivo de hacer más productivas las organizaciones; podría citar, dentro de las primeras TG que se expresaron con tal, al Taylorismo y el Fordismo. Hacia finales del siglo XX y principios de éste, nos encontramos con una amplia variedad de TG; las cuales se han convertido en soporte fundamental de las organizaciones para ser competitivas o diferenciarse en este mercado global en el que vivimos. Si la Tecnología es definida como el conocimiento aplicado y la Gestión es el hacer (éstas son definiciones pragmáticas); entonces podría decir que las TG son el conjunto de conocimientos aplicados para hacer un trabajo dentro de una organización, por ejemplo cuando se utiliza el ciclo PDCA (*plan, do, check, act*) o círculo de Deming, se está aplicando una TG para llevar a cabo las operaciones o proyectos en la organización; esta tecnología se basa en realizar metodológica y secuencialmente acciones de planeación, ejecución, evaluación y mejoramiento. Las TG se convierten en herramientas, basadas en el conocimiento teórico práctico, desarrolladas a través de propuestas probadas (método científico; muchas veces ensayo y error), generando modelos para ejecutar los trabajos en las organizaciones, buscando ser más productivos en el día a día de sus operaciones [1].

Las llamadas Tecnologías de la Información y la Comunicación (TIC) son los recursos y herramientas que se utilizan para el proceso, administración y distribución de la información a través de elementos tecnológicos, como: ordenadores, teléfonos, televisores, etc. A través del paso del tiempo la utilización de este tipo de recursos se ha incrementado y actualmente presta servicios de utilidad como el correo electrónico, la búsqueda y el filtro de la información, descarga de materiales, comercio en línea, entre otras. Su función principal es facilitar el acceso a la información fácil y rápida en cualquier formato, esto es posible a través de la inmaterialidad; es decir de la digitalización de la información para almacenarla en grandes cantidades o tener acceso aún si está en dispositivos lejanos. En segundo lugar, la inmediatez; pues la información tiene la capacidad de ser compartida instantáneamente. Aunque la característica más importante es que permite la comunicación bidireccional entre varias personas, esto es utilizado cuando se trata de foros, mensajería instantánea, videoconferencias. Con el desarrollo de la tecnología y la búsqueda de nuevas formas de comunicación más eficientes, las TIC se han posicionado como uno de los pilares básicos de la sociedad [2].

Aunque las técnicas básicas de la gestión documental ya se llevan aplicando desde hace décadas en el uso del control de documentos en formato de papel, y a lo largo de los años ha ido evolucionando, la exposición se concentrará en la definición de qué debería comprender un sistema de gestión documental avanzado, y cómo se podrá integrar los documentos internos y externos, en formato de papel, con la gestión de documentos electrónicos, y la influencia que tienen las nuevas herramientas y sistemas de gestión documental de última generación en los procesos de negocios relacionados con la creación, administración, control y gestión de documentos [3].

La gestión de inventarios es un punto determinante en el manejo estratégico de toda organización. Las tareas correspondientes a la gestión de un inventario se relacionan con la determinación de los métodos de registro, los puntos de rotación, las formas de clasificación y los modelos de reinventario, determinados por los métodos de control [4]. Los objetivos fundamentales de la gestión de inventarios son: reducir al mínimo "posible" los niveles de existencias y asegurar la disponibilidad de existencias (producto terminado, producto en curso, materia prima, insumo, etc.) en el momento justo. El control de inventario es el proceso por el cual una empresa administra las mercancías que mantiene en almacén. Esto, con el objetivo de recopilar información de la entrada y salida de los productos, buscando además el ahorro de costes [5]. Es decir, el control de inventario permite, entre otros fines, llevar el registro de las existencias de la compañía.

Luego de varias investigaciones relacionadas con el tema de la gestión de inventarios en entidades de diferentes sectores de mercado, tanto productoras como prestadoras de servicios y comercializadoras, incluyendo hospitales, universidades, fábricas, reparación de automóviles y comercio minorista; realizadas por el Laboratorio de Logística y Gestión de la Producción del Instituto Superior Politécnico José Antonio Echeverría (LOGESPRO), en el período comprendido entre el año 2000 y el 2011, se han detectado problemas comunes que las afectan. Varias empresas estudiadas poseen sistemas informáticos ERP (Enterprise Resource Planning), que tienen módulos que permiten manejar el tema de inventarios con un concepto integral y el 100 % de ellas tienen un alto grado de subutilización o inutilización de estos módulos. La causa fundamental que impacta en el inadecuado uso, es el desconocimiento o la falta de preparación de los directivos y el personal que opera los sistemas en los temas de gestión y sus potencialidades, pues, en general, la preparación que reciben está centrada en los aspectos del manejo informático del sistema.

En estos últimos años, la necesidad de informatizar al país se ha hecho un proceso vital para el avance de la sociedad cubana. Hasta ahora el proceso de gestión del inventario solo constaba en formato escrito (en papel) por lo que es un proceso engorroso y lento a la hora de clasificar los distintos tipos de bienes que una entidad posee. Empresas y diversas universidades en el mundo se han abocado a la tarea de desarrollar y fortalecer los conceptos relacionados a la gestión logística con un enfoque de cadena de suministro y de qué manera, si es administrada de forma correcta, puede ayudar a las compañías en el fortalecimiento y mejora de sus procesos, así como la sustitución de viejas y obsoletas formas de realizar el proceso que consume una gran cantidad de tiempo y recursos tanto humanos como capitales.

La Oficina Nacional de Administración Tributaria (ONAT) es una institución pública encargada de la gestión de la recaudación, cobranza y fiscalización de los Tributos. La ONAT es la entidad encargada de velar por la aplicación de la legislación relativa a impuestos y otros ingresos no tributarios; desarrollar la organización para su recaudación en todo el país; organizar y dirigir la auditoría fiscal adoptando las medidas requeridas para contrarrestar la evasión fiscal, así como otras funciones o tareas derivadas de la Ley 113 del Sistema Tributario y otras normas referidas a la recaudación de tributos y aranceles y su control. Tiene entre sus principales objetivos y funciones: cumplir el plan de recaudación, mantener los niveles de disciplina tributaria e incrementar la capacidad de riesgo. Perfeccionar el registro de contribuyentes. Perfeccionar el análisis del control de la recaudación. Fortalecer el modelo de control de obligaciones tributarias. Perfeccionar los mecanismos de control a los contribuyentes e implementar el proyecto de intercambio digital ONAT-Banco. Está organizada en tres niveles ejecutivos: la Oficina Central que constituye el Primer Nivel, es el máximo Órgano Normativo; Ejecutivo y de Coordinación, dentro del Sistema de Administración Tributaria y de Tesorería. Las Oficinas Provinciales que constituyen el Segundo Nivel con funciones de dirección, control y ejecutivas. Y las Oficinas Municipales que constituyen el Tercer Nivel, con Funciones totalmente Ejecutivas.

En las 16 sucursales provinciales que posee la ONAT en el país el control del inventario se realiza de forma trimestral donde se revisan todos los bienes materiales y se actualiza su estado en dependencia si está en uso o almacenado (en deterioro o buen estado), pero en específico el Departamento de Informática es el encargado de mantener el total control de los equipos electrónicos destinados al uso informático de la provincia. Este proceso de gestión, en ocasiones, puede resultar complejo y tomar más tiempo de lo

necesario, debido a que actualmente no se consta con un sistema informático que permita la gestión del inventario en la ONAT en la provincia de Las Tunas.

La **situación problemática** anteriormente planteada condujo a dirigir la investigación a la solución del siguiente **problema científico**: ¿Cómo gestionar el inventario para la ONAT en la dirección provincial de Las Tunas, mediante un sistema informático? Se definió como **objeto de estudio** las webs para la gestión de inventarios y como **campo de acción** la gestión del inventario de la ONAT en la dirección provincial de Las Tunas. Para la solución del problema se estableció como **objetivo general**: Desarrollar un sistema informático que contribuya al proceso de gestión de inventario para la ONAT en la dirección provincial de Las Tunas. Derivado de este se establecieron los siguientes:

### **Objetivos específicos**

1. Fundamentar los conceptos, características y antecedentes de los sistemas para la gestión de inventario.
2. Diseñar el sistema informático que contribuya al proceso de gestión de inventario para la ONAT en Las Tunas.
3. Implementar el sistema informático que contribuya al proceso de gestión de inventario para la ONAT en Las Tunas.
4. Evaluar el sistema informático que contribuya al proceso de gestión de inventario para la ONAT en Las Tunas.

Como métodos de investigación a nivel teórico se utilizaron:

- **Analítico-Sintético**: se empleó en la revisión detallada de las herramientas informáticas, tecnologías web y gestores de base de datos para su futura aplicación en la automatización de un sistema informático para la gestión del inventario especializado.
- **Análisis Histórico-Lógico**: permitió el estudio de los antecedentes de la gestión del inventario de la ONAT en Las Tunas.
- **Inductivo-deductivo**: se empleó en todo el proceso investigativo: en el diagnóstico, la proyección y validación.
- **Modelación**: se utilizó en el proceso del diseño del sistema para la gestión del inventario.

Además, se emplearon métodos a nivel empírico, los cuales fueron:

- **Observación:** para analizar el proceso actual de gestión del inventario.
- **Entrevista:** para recopilar información acerca de los procesos y pasos a seguir.

El desarrollo de la investigación se encuentra dividido en:

**Capítulo 1:** Fundamentos y referentes teórico-metodológicos sobre webs para la gestión de inventarios. Recoge un análisis del tema a tratar. Incluye el estado del tema (a nivel de provincia), de las tendencias, técnicas, tecnologías, metodologías y *software* usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta.

**Capítulo 2:** Diseño del Sistema de gestión de inventario para la Oficina Nacional de Administración Tributaria (ONAT) en Las Tunas. Análisis y diseño de la propuesta de solución a través de herramientas de ingeniería de *software*.

**Capítulo 3:** Implementación y Pruebas del Sistema de gestión de inventario para la Oficina Nacional de Administración Tributaria (ONAT) en Las Tunas. Contiene las pruebas realizadas al sistema y sus resultados.

Al finalizar el capítulo 3 se encuentran las conclusiones, recomendaciones, las referencias bibliográficas y los anexos.

## **Capítulo 1. Fundamentos y referentes teórico-metodológicos sobre webs para la gestión de inventarios.**

### **1.1 Introducción**

En cualquier empresa u organización la gestión y control del inventario es un proceso necesario para mantener la eficiencia y eficacia del negocio además de suponer; si se tiene un buen control, un ahorro monetario a veces de hasta el 30% de los ingresos generales.

En la ONAT como en todas las empresas del país y del mundo la gestión del inventario es controlada por el departamento de economía que es la entidad encargada de esta actividad, pero al mismo tiempo cada departamento debe tener un control de su propio inventario como es el caso del departamento de informática, que debe tener un conteo exacto de todos sus equipos y su estado actual. El jefe del departamento de informática es el encargado de controlar dicho inventario, así como mantener en documentos el estado de cada equipo y los reportes para el departamento de economía.

### **1.2 Conceptos fundamentales asociados al dominio del problema**

#### **1.2.1 Gestión Documental y Sistema de gestión documental**

Pueden existir variaciones a la hora de definir Gestión Documental y Sistema de gestión documental [3], pero todos siguen un mismo significado dado que la esencia de su concepto no varía por lo que en esta investigación se asume el siguiente concepto:

**Gestión Documental:** Es el conjunto de normas que se aplican para gestionar los documentos de todo tipo que se crean y reciben en una organización. Esa gestión de documentos debe facilitar su recuperación, permitir la extracción de información, el expurgo de los que no sean necesarios, la conservación de los que son importantes durante el tiempo que sean útiles, así como la destrucción cuando ya no se necesitan, utilizando para todo esto métodos eficaces y eficientes.

**Sistema de gestión documental:** La administración y el control del enorme volumen de los documentos externos e internos, que forman parte de los procesos de negocios en la administración pública o empresarial, se ha convertido en un reto que ha promovido la adopción de diferentes herramientas que facilitan su gestión, nos brindan mejoras importantes en su control y aportan considerables reducciones de tiempo de búsqueda, manipulación y ahorro de gastos, muy significativos en nuestros procesos administrativos.

### 1.2.2 Inventario físico

Es una lista elaborada que corrobora la existencia real de productos y materias primas almacenadas, verifica su estado y define su estatus. Para ser capaz de llevar un correcto control de todas tus existencias, se recomienda el uso de una herramienta que pueda estipular el total de movimientos de tu almacén de forma actualizada [6].

### 1.2.3 Gestión de inventarios y Control de inventario

Para este trabajo de diploma es necesario ver los conceptos de gestión y control del inventario como uno solo, ya que el inventario a controlar no es una mercancía ni está almacenado en su totalidad por lo que parte de él está actualmente en uso y uno solo de los dos conceptos no abarca la definición de dicho inventario. Para llegar a esta conclusión fue necesario estudiar estos conceptos y compararlos en paralelo con la problemática a investigar:

La **gestión de inventarios** se incluye dentro de la rama de la contabilidad de costes y se define como la administración adecuada del registro, compra y salida de inventario dentro de la empresa [7].

El **control de inventario** permite llevar el registro de las existencias de la compañía. Así, en base a dichos datos pueden tomarse decisiones, por ejemplo, para que las firmas mantengan en lo posible el menor stock inmovilizado, lo cual implica un coste de almacenamiento.

Un **equipo** es un bien material de tipo electrónico, de este modo un grupo de equipos conforman el inventario a gestionar.

Cabe recordar que con **inventario** se suele hacer referencia a la materia prima, los bienes intermedios y los bienes finales que ofrece la firma a sus clientes [5].

## 1.3 Soluciones Informáticas Similares

En este epígrafe se analizarán algunos de los Sistemas de Gestión de Inventario del mercado, enmarcando las ventajas y desventajas de cada uno para ayudar a la realización de la futura aplicación, incluir funcionalidades similares y solucionar las inconformidades de los usuarios que las utilizan.

### 1.3.1 ABC Inventory

ABC Inventory es un subconjunto gratuito del sistema de control Almyta (ACS). Esta aplicación MRP II ha sido diseñada para manejar todos los aspectos de la administración de inventario, brindando la capacidad de rastrear cada paso en el ciclo de vida del inventario desde el momento en que crea una orden de

compra para su proveedor hasta el momento en que envía el producto a su cliente. Al mismo tiempo, la organización modular de ABC Inventory asegura flexibilidad y le permite adaptar el programa a sus necesidades específicas. La versión comercial de ABC Inventory es 100% personalizable.

**Ventajas:**

- Posee funciones para el inventario.
- Fácil de usar para usuarios con experiencia.
- Es gratuita.
- No se limita a gestionar el inventario de una sola empresa.

**Desventajas:**

- Recuentos de inventario inexactos.
- Alta complejidad de uso para usuarios sin experiencia.

**1.3.2 Stockipile de Canvus**

Canvus es un *software* de gestión de inventarios dirigido específicamente a pequeñas empresas. Su sistema de gestión de inventarios llamado Stockipile, ofrece al usuario la posibilidad de aceptar devoluciones, registrar ventas y artículos dañados y supervisar la rotación de productos.

**Ventajas:**

- Fácil de usar.
- Agregar nuevos materiales no presenta ninguna complejidad.
- Datos bursátiles.

**Desventajas:**

- Muy específico a la hora de buscar artículos.
- No posee opciones de impresión.

**1.3.3 Delivrd**

Delivrd es una plataforma de gestión de inventario y cumplimiento de pedidos, diseñada para ayudar a las empresas a agilizar y consolidar los procesos logísticos. Las características incluyen una función de selección, empaque y envío, creación y escaneo de códigos de barras, empaquetado de olas e



integraciones de terceros con Woocommerce y Shopify. Con Delivrd, las empresas pueden usar códigos de barras para identificar y confirmar los productos correctos, los números de pedido y la cantidad requerida, al mismo tiempo que proporcionan datos de seguimiento de pedidos. Ofrece una gestión de inventario regular y dañada para procesar devoluciones defectuosas o no deseadas de los clientes y devoluciones de existencias dañadas a los proveedores. Esto ayuda a mantener un recuento de existencias actualizado dependiendo de si las existencias se envían para reparación, reemplazo o descarte.

Durante la investigación se pudo encontrar muy pocas ventajas o desventajas de este *software*, aunque se sabe que posee una versión gratuita y una interfaz de fácil acceso, a medida que el inventario crezca, se enfrentará a más y más problemas de inventario, como la imposibilidad de realizar un seguimiento de los cambios diarios en la disponibilidad de inventario del remitente. De modo que, tarde o temprano, se terminará eligiendo otro modelo de cumplimiento minorista.

#### **1.4 Tecnologías de Desarrollo**

En este epígrafe se especificarán algunas las tecnologías usadas en la realización de soluciones informáticas, y se definirá el *framework*<sup>1</sup> y los lenguajes de programación y maquetado a usar en el desarrollo de la aplicación.

##### **1.3.1 Lenguajes de desarrollo**

###### **UML**

El **Lenguaje Unificado de Modelado** (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de *software* complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de *software*, p. ej., en el flujo de procesos en la fabricación.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos [8].

---

<sup>1</sup> Un *framework* o marco de trabajo es un conjunto de tecnologías que ofrece una estructura base para elaborar un proyecto específico, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de *software*.

El desarrollo *frontend*<sup>2</sup> está basado en **HTML** y **CSS**, los lenguajes de maquetación que permiten definir la estructura y estilos de una página web, las funcionalidades, animaciones y otros elementos del *frontend* son programados con **JavaScript**, un lenguaje de programación para definir la lógica de nuestra aplicación, recibir las solicitudes de los usuarios y enviárselos al *backend*<sup>3</sup> [9].

### **HTML, CSS y JavaScript:**

- HTML (lenguaje de marcas de hipertexto), es el lenguaje donde se define la información o el contenido del documento, el formato de los archivos es *.html*. [10]
- CSS (cascading style sheets), el lenguaje donde se especifica el diseño del documento, maneja todo lo relacionado con la parte visual, el formato de los archivos es *.css*. [10]
- JavaScript, el lenguaje que hace que todo sea interactivo, es realmente el lenguaje de programación que nos permite crear sitios web, el formato de los archivos es *.js*. [10]

Para la realización de la solución se utilizará HTML en su versión 5, CSS en su versión 3 y JavaScript, todos del lado del cliente.

Algunos lenguajes de programación para *backend* son Python, Node.js, PHP y C#. Y así como en el *frontend*, todos estos lenguajes tienen diferentes *frameworks* que permiten trabajar mejor según el proyecto que se desarrolle, como Django, Laravel, Symphony y ASP.Net.

### **PHP**

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

---

<sup>2</sup> Frontend en el desarrollo web está dedicado a la parte frontal de un sitio, su diseño.

<sup>3</sup>Backend es la parte del desarrollo que se encarga de la lógica. Conjunto de acciones en una web que se realizan de forma no visual. Ej.: comunicación con el servidor.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

Se utilizará PHP en su versión 7.4.25, esto se debe a que es la versión estable más actualizada hasta la fecha.

### 1.3.2 *Frameworks*

Además de los lenguajes de programación que definen cómo estará escrito el código, es importante destacar la importancia de los *frameworks* de desarrollo. Herramientas y librerías de código pre-escrito que facilitan a los desarrolladores las tareas cotidianas o de uso común para evitar ejecutarlas desde cero.

#### **Symfony**

Symfony es un entorno de trabajo estandarizado (*framework* PHP) que se utiliza para el desarrollo de aplicaciones web y es de los más utilizados en el entorno de desarrolladores de apps. En otras palabras, es una herramienta para desarrolladores para crear aplicaciones en PHP. Una de las principales ventajas de Symfony es que posee una licencia MIT, o lo que es lo mismo que una licencia de *software* libre permisivo, que se puede utilizar dentro del *software* del propietario y al no poseer *Copyright*<sup>4</sup> permite su modificación.

Para poder trabajar con esta herramienta es necesario disponer de un servidor web, una consola de comandos del sistema operativo y cualquier versión de PHP5 o PHP7 si se quiere utilizar las últimas versiones. El *framework* de desarrollo de aplicaciones tiene una gran aceptación y popularidad entre programadores de PHP en Europa, lo que hace que tenga una amplia comunidad que ofrece formación, consultorías y desarrollo de proyectos. Es importante destacar que Symfony se puede utilizar en la creación de páginas web y aplicaciones.

Una de las principales preocupaciones de los clientes que contratan el desarrollo de aplicaciones es su escalabilidad, es decir, si una vez creada se puede ampliar y crecer a pesar de tener un código extenso.

---

<sup>4</sup> Copyright se refiere al derecho de autor. Protección legal de una obra de propiedad intelectual.

Efectivamente, la estructura Modelo Vista Controlador del *framework* permite un progreso en función de las necesidades de los clientes.

Además, este *framework* tiene una gran cantidad de plantillas para desarrollar webs y es muy utilizado en la creación de APIs<sup>5</sup>.

Entre sus principales características tenemos:

- Permite crear desde sitios webs sencillos hasta macro proyectos y aplicaciones complejas.
- Al poseer una gran comunidad de desarrolladores, garantiza formación continua, además su creador Fabien Potencier es de los más activos en el proyecto en todo el mundo.
- Sus componentes están altamente testeados, tanto como para que Drupal 8 se haya construido con Symfony [11].

En el desarrollo de la aplicación se usará la versión 5.3.10 del *framework*.

## **Bootstrap**

Bootstrap, el marco de interfaz de usuario más popular creado para diseñar interfaces modernas, receptivas y dinámicas para páginas web de diseño profesional.

Bootstrap 5 alpha se lanzó oficialmente el 16 de junio de 2020 después de varios meses de refinamiento. Con todos los cambios importantes en esta versión, el equipo de desarrollo de Bootstrap 5 informó a los usuarios que la versión actual todavía está en la versión alfa, por lo tanto, los cambios importantes continuarán ocurriendo hasta que se publique la primera versión beta, por lo que es mejor verificar siempre los problemas abiertos. y extraiga solicitudes en su repositorio oficial de GitHub para preguntas abiertas y comentarios [12].

## **1.4 Herramientas de desarrollo**

### **Visual Paradigm 15.1**

Visual Paradigm es una herramienta CASE: Ingeniería de *Software* Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación,

---

<sup>5</sup> Las API son las interfaces de programas de aplicación, por sus siglas en inglés.

pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque Orientado a Objetos [13].

Visual Paradigm será usado para crear todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. Para el desarrollo de la solución se propone utilizar el Visual Paradigm en su versión 15.1.

### **MySQL 8.0**

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL, es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle [14].

### **NPM 6.0.0**

NPM es el administrador de paquetes para la plataforma Node JavaScript. Coloca módulos en su lugar para que el nodo pueda encontrarlos y gestiona los conflictos de dependencia de forma inteligente.

Es extremadamente configurable para admitir una amplia variedad de casos de uso. Por lo general, se utiliza para publicar, descubrir, instalar y desarrollar programas de nodo [15].

### **NetBeans**

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) es un programa que permite a los programadores desarrollar un *software* de forma más rápida, ya que brinda un conjunto de herramientas como editor de código amigable al programador según el lenguaje en que se está programando. Brinda además un depurador y herramientas para la construcción de interfaces con mayor facilidad. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. [16]

Esta es una plataforma de escritorio para el desarrollo de aplicaciones que usa Java mediante un entorno de desarrollo integrado, la cual permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de *software* llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de *software* [17].

Será utilizada la versión 12.3 de este IDE la cual tiene como nombre Apache Netbean.

### **Laragon**

Laragon es un entorno de desarrollo universal portátil, aislado, rápido y potente para PHP, Node.js, Python, Java, Go, Ruby. Es rápido, liviano, fácil de usar y fácil de extender. Es ideal para crear y administrar aplicaciones web modernas. Se centra en el rendimiento, diseñado en torno a la estabilidad, la simplicidad, la flexibilidad y la libertad. Es muy ligero y se mantendrá lo más delgado posible. El binario central en sí tiene menos de 2 MB y utiliza menos de 4 MB de RAM cuando se ejecuta. No utiliza los servicios de Windows. Tiene su propia orquestación de servicios que administra los servicios de forma asincrónica y sin bloqueo, por lo que encontrará que todo funciona de forma rápida y sin problemas con Laragon.

Nota: Laragon está verdaderamente aislado y es portátil. Sin embargo, es posible que deba usar el instalador, ya que detectará e instalará los componentes de tiempo de ejecución que faltan y que son necesarios para ejecutar aplicaciones C ++ creadas con Visual Studio, como PHP, Apache [18].

### **1.5 Metodología de Desarrollo de Software**

Las metodologías de desarrollo de *software* son el marco de trabajo que colecciona un conjunto de pasos y procedimientos que se deben seguir para organizar, controlar y planear el proceso de desarrollo de un *software*. Surge ante la necesidad de trabajar mediante el uso de procedimientos, técnicas, herramientas y documentos durante el desarrollo de *software* [19]. Para la presente investigación se define como metodología de desarrollo de *software* AUP-UCI.

El Proceso Unificado Ágil (AUP, del inglés Agile Unified Process) describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles. Se decide la variación UCI de AUP pues no existe una metodología de *software* universal, toda metodología debe

ser adaptada a las características de cada proyecto. La misma define que para el ciclo de vida de los proyectos de la UCI, de las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición), mantener la fase de Inicio, las 3 fases restantes de AUP se agrupan en una sola, que es Ejecución y se agrega la fase de Cierre.

Este trabajo de investigación se centra en la fase de Ejecución. En esta fase se ejecutan las actividades requeridas para desarrollar el *software*, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

AUP propone siete disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener siete disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en tres disciplinas: Pruebas Internas, de Liberación y Aceptación. Además, incluye las tres disciplinas restantes las cuales son Gestión de la Configuración, Planeación de proyectos y Monitoreo y Control de Proyectos.

De entre estas disciplinas destaca Requisitos por tener cuatro escenarios posibles para modelar el sistema en los proyectos. De ellos para esta investigación se seleccionó el cuarto: Historias de Usuarios (HU). Estas aplican a los proyectos que hayan definido muy bien el negocio. El cliente estará siempre presente durante el desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Es recomendado en proyectos no muy extensos, ya que una HU no debe poseer demasiada información [20].

## **1.6 Conclusiones parciales del Capítulo**

Al haber culminado este capítulo, se han esclarecido los conceptos más importantes sobre la temática y se ha alcanzado un nivel más alto de comprensión en el ámbito de la gestión y el control de inventarios. Se analizaron las soluciones similares y gracias a ello se ha llegado a una idea más clara sobre la estructura del sistema y sus principales funcionalidades. Debido a lo anteriormente expuesto se llegó a la conclusión de:

Serán empleados los *framework* Symfony en su versión 5.3.10 y Bootstrap 5 para el desarrollo de la solución informática.

El lenguaje a utilizar por el lado del backend será php en su versión 7.4.25 y JavaScript, HTML y CSS en el Frontend.

Se utilizará Apache Netbean IDE 12.3 como entorno de desarrollo. El sistema usará como motor de base de datos MySQL 8.0 y Laragon como servidor para las bases de datos.

La metodología a utilizar será AUP en su versión UCI, debido a que es el que más se ajusta a los requerimientos a la hora de desarrollar la aplicación final.



## **Capítulo 2. Diseño del sistema de gestión de inventario para la ONAT en Las Tunas.**

### **2.1 Introducción**

Al diseñar una arquitectura de *software* se crean y representan componentes que interactúan entre sí, con responsabilidades específicas que se organizan de forma tal que se logren los requerimientos establecidos.

En el presente capítulo se presenta los usuarios con sus responsabilidades y la captura de requisitos con la elaboración de las historias de usuario junto a la planificación que plantea la metodología seleccionada. Se parte del modelo de dominio debido a que no fue posible definir los procesos del negocio en cuestión. Además, se presentan las funcionalidades que realiza el sistema obteniéndose los requisitos funcionales y no funcionales presentes para dar solución a los problemas definidos. Se hace alusión al análisis y el diseño del sistema, en los que se incluyen los diagramas correspondientes.

### **2.2 Modelado del proceso de negocio**

El modelado de procesos de negocios es una técnica que implica la creación de una representación visual de un proceso de negocios. Esto se logra típicamente usando herramientas de modelado de procesos de negocios como los diagramas de flujo y la notación universal de procesos de modelado de negocios (también conocido como BPMN) [21].

#### **2.2.1 Diagrama de actividad**

Un diagrama de actividades es un diagrama de flujo que muestra actividades ejecutadas por un sistema. Los diagramas de actividades, junto con los diagramas de casos de uso y los diagramas de máquina de estados, son considerados diagramas de comportamiento porque describen lo que debe suceder en el sistema que se está modelando.

Las partes interesadas tienen muchos asuntos que manejar, por lo que es importante una comunicación clara y concisa. Los diagramas de actividades ayudan a que las personas en las áreas de negocios y desarrollo de una organización se integren para comprender el mismo proceso y comportamiento. Usarás un conjunto de símbolos especializados —incluidos aquellos para pasos de inicio, finalización, fusión y recepción en el flujo— para crear un diagrama de actividades.

En la Figura 1 se muestra el diagrama de actividad Gestionar equipo el cual inicia una vez autenticado el usuario Info\_Admin. En dependencia de si el equipo a gestionar aún no existe en el sistema el usuario seleccionará la opción Crear Equipo y rellenará los campos con la información correspondiente, si el equipo ya existía el usuario podrá editarlo o eliminarlo. Al finalizar la operación Crear Equipo o Editar seleccionará la opción Listar Inventario para cerciorarse que el equipo fue creado o editado con éxito pasando así al historial de acciones para una vez más confirmar la operación, si el equipo fue eliminado de la misma forma que al crear o editar es necesario confirmar la operación en el historial de acciones.

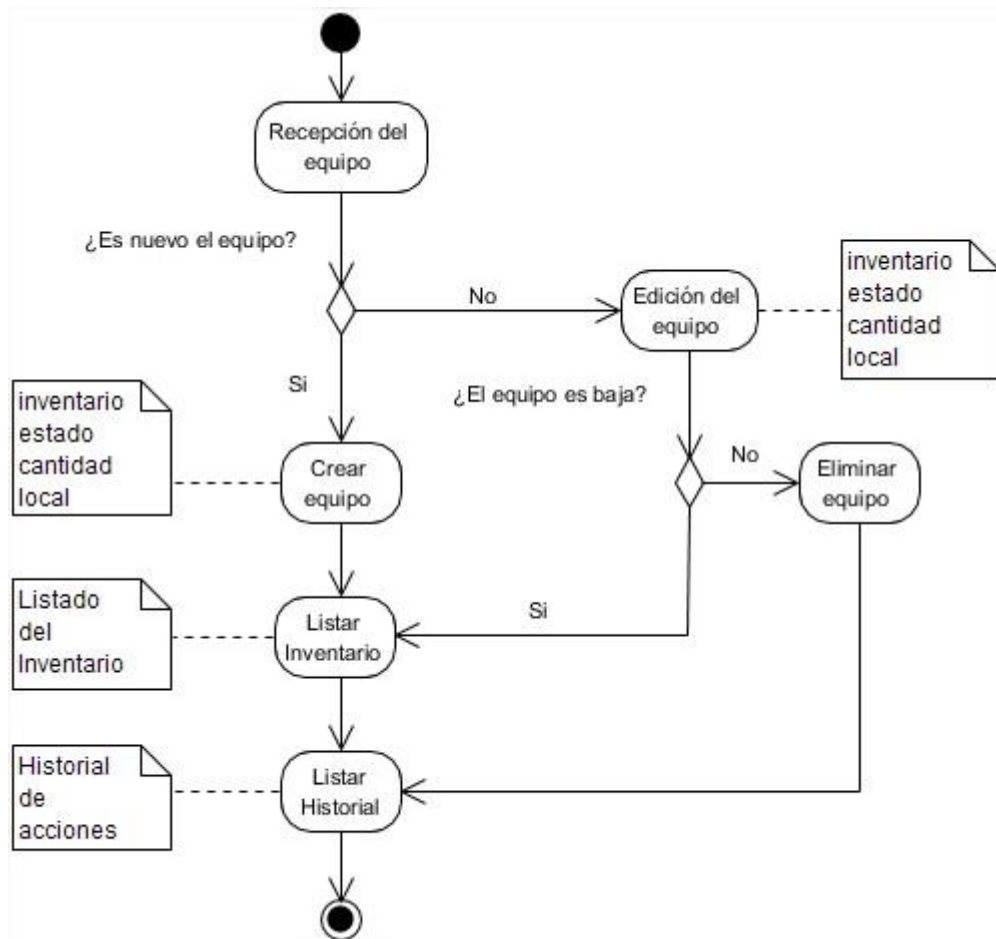


Figura 1 Diagrama de Actividad para gestionar un equipo en el sistema (Elaboración propia).

### **Relación de conceptos importantes.**

- Info\_Admin: es la persona a cargo del Departamento de Informática de la provincia.
- Equipo: medio informático
- Historial de acciones: lista de acciones en orden cronológico realizadas por la persona que accede al sistema.
- Estado: condición en la que se encuentra el equipo (bueno, regular, en proceso de eliminación)
- Local: municipio en el que se encuentra el equipo.
- Inventario: número de serie asignado por el departamento de economía y que es único para cada equipo.

## **2.3 Flujo de Trabajo Requerimientos**

En este flujo de trabajo se define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. Además, son condiciones o capacidades que necesita un usuario para resolver un problema o lograr un objetivo. Los requisitos se pueden clasificar en Funcionales y No Funcionales [22].

### **2.3.1 Definición de los Requerimientos Funcionales**

Son capacidades o condiciones que un sistema debe cumplir para satisfacer las necesidades del cliente. En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización. Estas actividades no son exactamente los requerimientos funcionales, pero sí son el punto de partida para identificar qué debe hacer el sistema. Los requerimientos funcionales se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

### **Requisitos de *Software* [23].**

“Requisitos de *software* consiste en la especificación y validación de las funcionalidades que el sistema a desarrollar debe proporcionar, así como de las restricciones que el sistema debe cumplir. [...]”

### **Los requisitos funcionales definidos son:**

**RF1:** Autenticar Usuario.

**RF2:** Añadir nuevo equipo al inventario.

**RF3:** Editar equipo en el inventario.

**RF4:** Eliminar equipo del inventario.

**RF5:** Listar equipo en el inventario.

**RF6:** Listar historial.

**RF7:** Listar Inventario.

**RF8:** Imprimir inventario.

**RF9:** Adicionar usuarios.

**RF10:** Modificar usuarios.

**RF11:** Eliminar usuarios.

*Tabla 1 Descripción de Requisitos Funcionales*

<b>Nombre</b>	<b>Descripción</b>	<b>P</b>	<b>C</b>
Autenticar Usuario	El sistema debe permitir restringir el acceso a determinadas funcionalidades a usuarios.	A	M
Añadir nuevo equipo al inventario	El sistema debe permitir añadir un nuevo equipo al inventario y se le asigna un número de inventario, local en el que se encontrará, la cantidad y estado.	A	M
Modificar equipo en el inventario	El sistema debe permitir editar las características de un equipo existente en el inventario.	M	M
Eliminar equipo en el inventario	El sistema debe permitir eliminar un equipo del inventario.	A	A
Listar equipo en el inventario	El sistema muestra un equipo específico según su número de inventario.	B	M

Listar historial	El sistema debe mostrar de manera cronológica todas las entradas de los usuarios y las acciones que realizaron.	B	A
Listar Inventario	El sistema debe mostrar una lista de los equipos existentes en el inventario y sus características.	B	A
Imprimir inventario	El sistema debe permitir imprimir una lista de los equipos con su número de inventario, cantidad y estado.	M	A
Adicionar Usuario	El sistema debe permitir adicionar un usuario.	A	M
Modificar Usuario	El sistema debe permitir modificar un usuario.	A	M
Eliminar Usuario	El sistema debe permitir eliminar un usuario.	A	A

### 2.3.2 Definición de los Requerimientos no Funcionales

Los requerimientos no funcionales son los que especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste con los requerimientos funcionales que especifican los comportamientos específicos de las aplicaciones [24].

Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.



**Ian Sommerville. Software Engineering. 9na edición**

*Figura 2 clasificación de requerimientos no funcionales [25]*

**RNF 1:** Restricciones en el diseño e implementación.

- Las páginas no deben exceder los 600 KB.
- No debe contener muchas imágenes para que la recarga sea rápida.
- Se debe usar el marco de trabajo Apache Netbean IDE 12.3.
- Se debe usar el motor de plantillas Bootstrap.

- Se debe usar el NPM 6.0.0.

**RNF 2:** Requerimientos de *software*.

- Sistema gestor de bases de datos MySQL, versión 8.
- Se recomienda usar como navegador Google Chrome o Mozilla Firefox por la velocidad de ejecución de JavaScript, aunque el sistema será probado en varios navegadores.

**RNF 3:** Requerimientos de *Hardware*. Servidor con:

- Memoria RAM 2 GB como mínimo.
- Disco duro 20 GB como mínimo.
- Microprocesador 2.5 GHZ como mínimo.

**RNF 4:** Requerimientos de soporte.

- Se debe realizar una capacitación a los usuarios.

**RNF 5:** Requerimientos de seguridad.

- Validación de datos en el cliente, para evitar estados inconsistentes en la información y posibles ataques al sistema.
- Confidencialidad: sistema de autenticación basado en roles.
- Disponibilidad: la información necesaria y autorizada para cada rol del sistema estará siempre disponible.

**RNF 6:** Requerimientos de interfaz externa.

- El diseño debe ser simple.
- Las opciones deben estar agrupadas de forma tal que el usuario pueda localizarlas fácilmente.
- La interfaz debe ser adaptable a todas las resoluciones de pantalla.
- El sistema estará realizado un conjunto de íconos que reflejan en sí la acción que realizará el usuario.

### 2.3.3 Actores del Sistema

Tabla 2 Actores del Sistema (Elaboración propia)

Usuario	Responsabilidad
Info_Admin	Es el encargado de la gestión de los usuarios, así como la gestión de los equipos. Además del correcto funcionamiento del sistema.
Info_Munic	Usuario encargado de adicionar al sistema los equipos entrantes en su municipio.
Económico	Usuario encargado de hacer revisiones al inventario existente.

### 2.3.4 Diagrama Entidad-Relación

Un diagrama entidad-relación permite representar las entidades más relevantes presentes en la base de datos de un sistema, detallando sus campos, propiedades y relaciones.



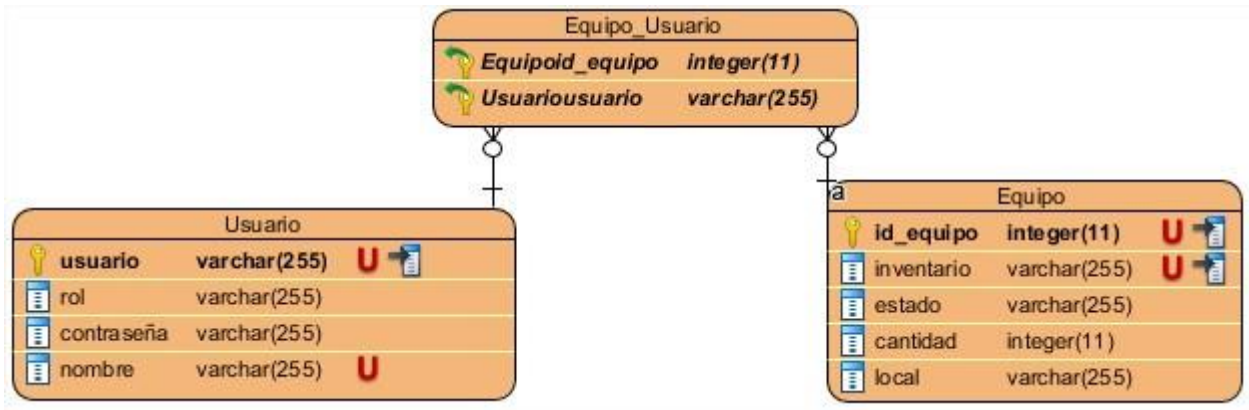


Figura 3 Diagrama Entidad-Relación (Elaboración propia).

Un Usuario es la persona encargada de gestionar los equipos existentes en el sistema, el mismo posee un nombre y usuario que será único y la lleve primaria de la tabla, un rol el cual le permitirá desempeñar distintas funciones en el sistema, además de una contraseña de la cual solo tendrá conocimiento él dado que es la clave secreta de cada usuario para acceder en el sistema y poder desempeñar sus roles. La tabla Usuario posee una relación de muchos a muchos con la tabla Equipo ya que varios usuarios pueden gestionar varios equipos y varios equipos pueden ser gestionados por varios usuarios. Un Equipo es un bien material o medio básico de la empresa que poseerá un `id_equipo` el cual es incremental, la llave primaria de la tabla y no es posible modificar, un número de inventario que es el identificador único de cada equipo, un estado que puede variar en dependencia de la condición en la que se encuentra el equipo, una cantidad en caso de que el equipo a gestionar pueda ser almacenado en cantidades mayores a 1, y el local en el cual se encuentra ubicado el equipo.

## 2.4 Historias de Usuario

Las historias de usuario (HU) o *User Story*, forman parte del enfoque ágil y describen las características y necesidades de un *software* desde la perspectiva de un usuario, ayudando a alinear expectativas y evitar errores críticos en el futuro. Una historia de usuario puede considerarse como una preparación para establecer los requisitos del *software* [26]. Durante la fase Planificación-Definición se establecieron 11 HU, las cuales responden a cada una de las funcionalidades del sistema. A continuación, se describe una HU definida en la propuesta de solución como ejemplo, el resto de HU se presentan en los Anexos.

Tabla 3 HU Añadir nuevo equipo (Elaboración propia)

Historia de Usuario	
<b>Número:</b> 2	<b>Usuario:</b> Info_Admin, Info_Munic
<b>Nombre:</b> Añadir nuevo equipo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b>	
El administrador o informático de municipio podrá adicionar un equipo al sistema con los parámetros inventario, estado, cantidad y local.	
<p style="text-align: center; color: #4F81BD; font-weight: bold; font-size: 1.2em;">Adicionar Equipo</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px auto; width: 80%;"> <p>Inventario  <input type="text" value="108975"/></p> <p>Estado  <input type="text" value="Bueno"/></p> <p>Cantidad  <input type="text" value="1"/></p> <p>Local  <input type="text" value="ONAT Provincial"/></p> <p> <input type="button" value="Cancelar"/> <input type="button" value="Guardar"/> </p> </div>	
<p><i>Figura 4 Prototipo de interfaz Añadir nuevo equipo (Elaboración propia)</i></p>	

## 2.4.1 Planificación

### Plan de Iteraciones

Se planificaron iteraciones de aproximadamente tres semanas de duración para el proyecto. Para cada iteración se define un conjunto de HU que se van a implementar [27].

Tabla 4 Plan de Iteraciones (Elaboración propia)

Historia de Usuario	Mejor Caso	Peor Caso	Resultado
Autenticar Usuario	5 horas	10 horas	6 horas
Gestionar Equipo <ul style="list-style-type: none"> <li>• Añadir nuevo equipo al inventario</li> <li>• Editar equipo en el inventario</li> <li>• Eliminar equipo en el inventario</li> <li>• Listar equipo en el inventario</li> </ul>	2 semanas	3 semanas	2 semanas
Listar inventario	5 horas	10 horas	7 horas
Imprimir inventario	5 horas	9 horas	5 horas
Listar historial	7 horas	12 horas	7 horas
Gestionar Usuario <ul style="list-style-type: none"> <li>• Añadir usuario</li> <li>• Modificar usuario</li> <li>• Eliminar usuario</li> </ul>	1 semana	2 semanas	1 semana

### 2.5 Diagramas de Clases de Diseño con Estereotipos Web

Se han representado los diagramas de clases del diseño correspondiente a Autenticar Usuario, Gestionar Equipo y Gestionar Usuario descritas en el epígrafe 2.4.1. En lugar de usar los estereotipos básicos, se han usado los estereotipos para diagramas UML basados en la web, para una mayor comprensión de los mismos y de la estructura y el funcionamiento de cada HU. En estos se encuentran las clases, relaciones y formularios que contienen representados los requisitos funcionales, en el diagrama Gestionar Equipo serán incluidos los requisitos Listar inventario, Imprimir inventario y Listar historial.

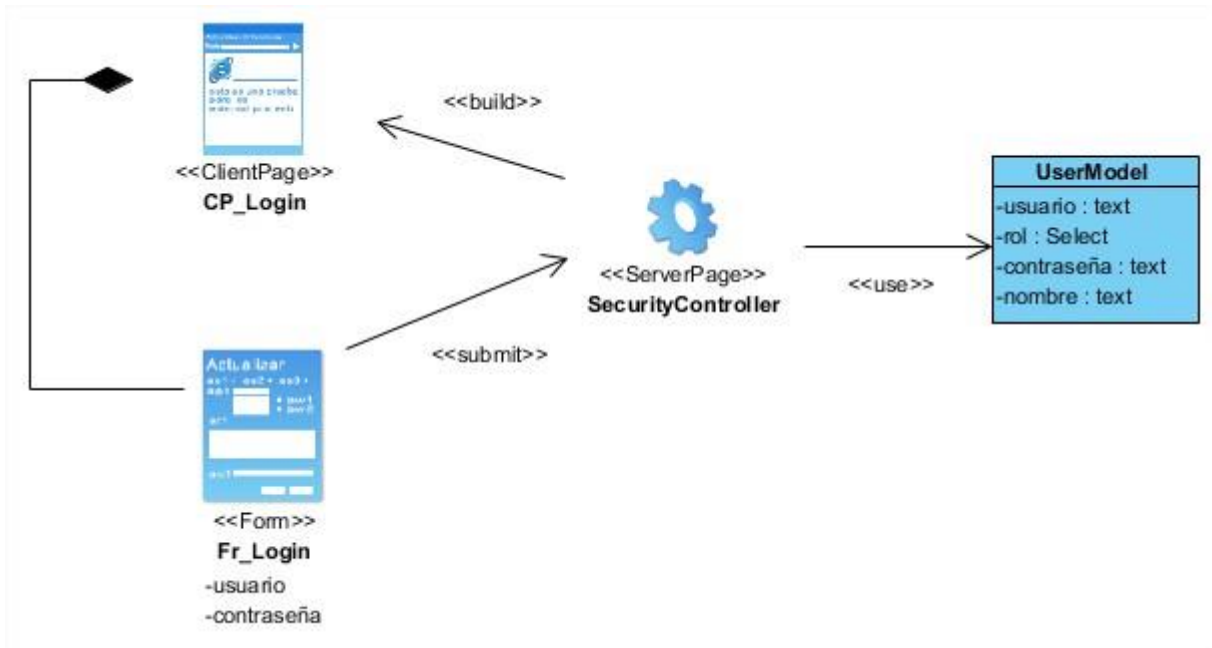


Figura 4 Diagrama de Clases Autenticar Usuario (Elaboración propia).

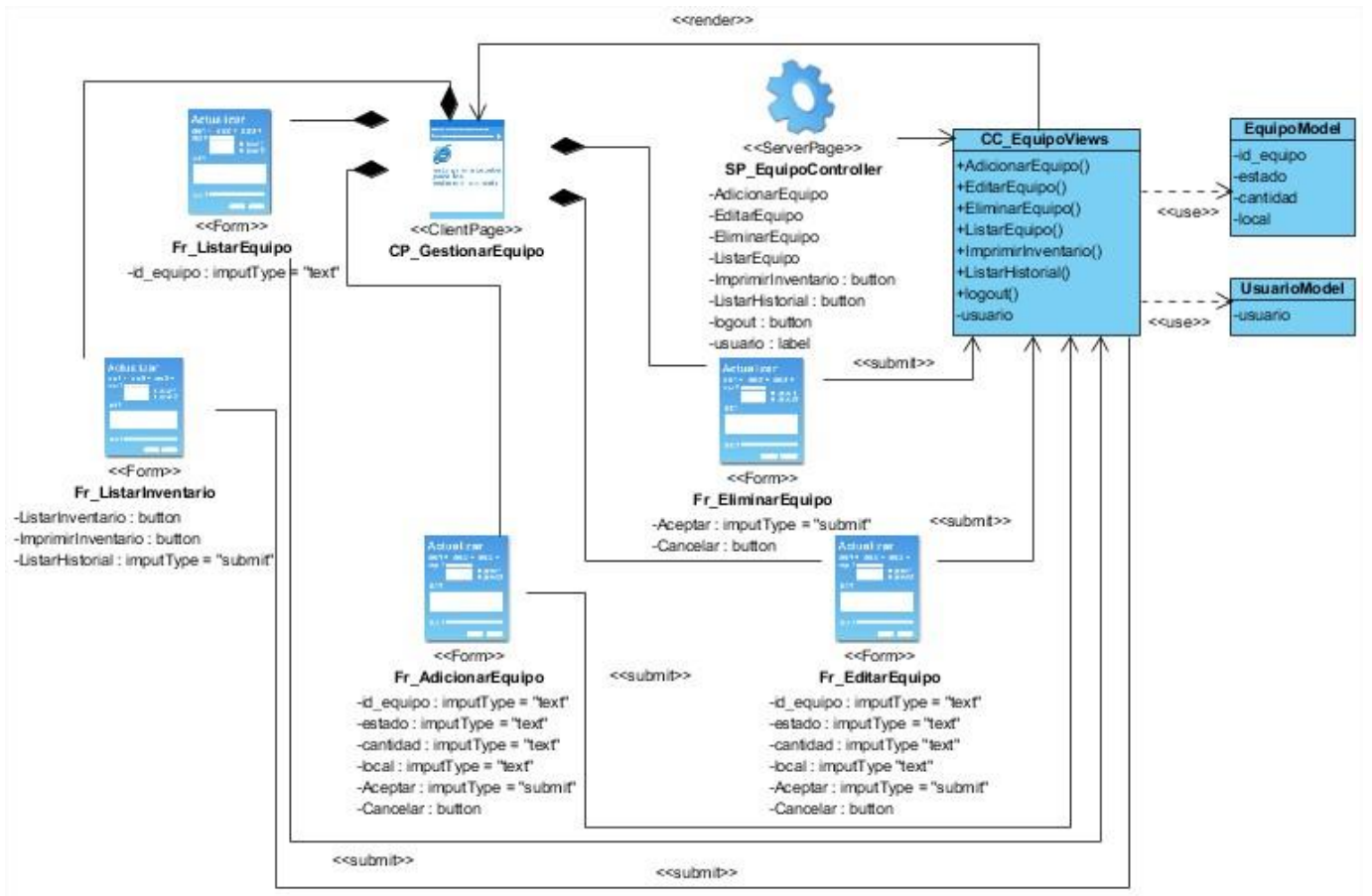


Figura 5 Diagrama de Clases Gestionar Equipo (Elaboración propia).

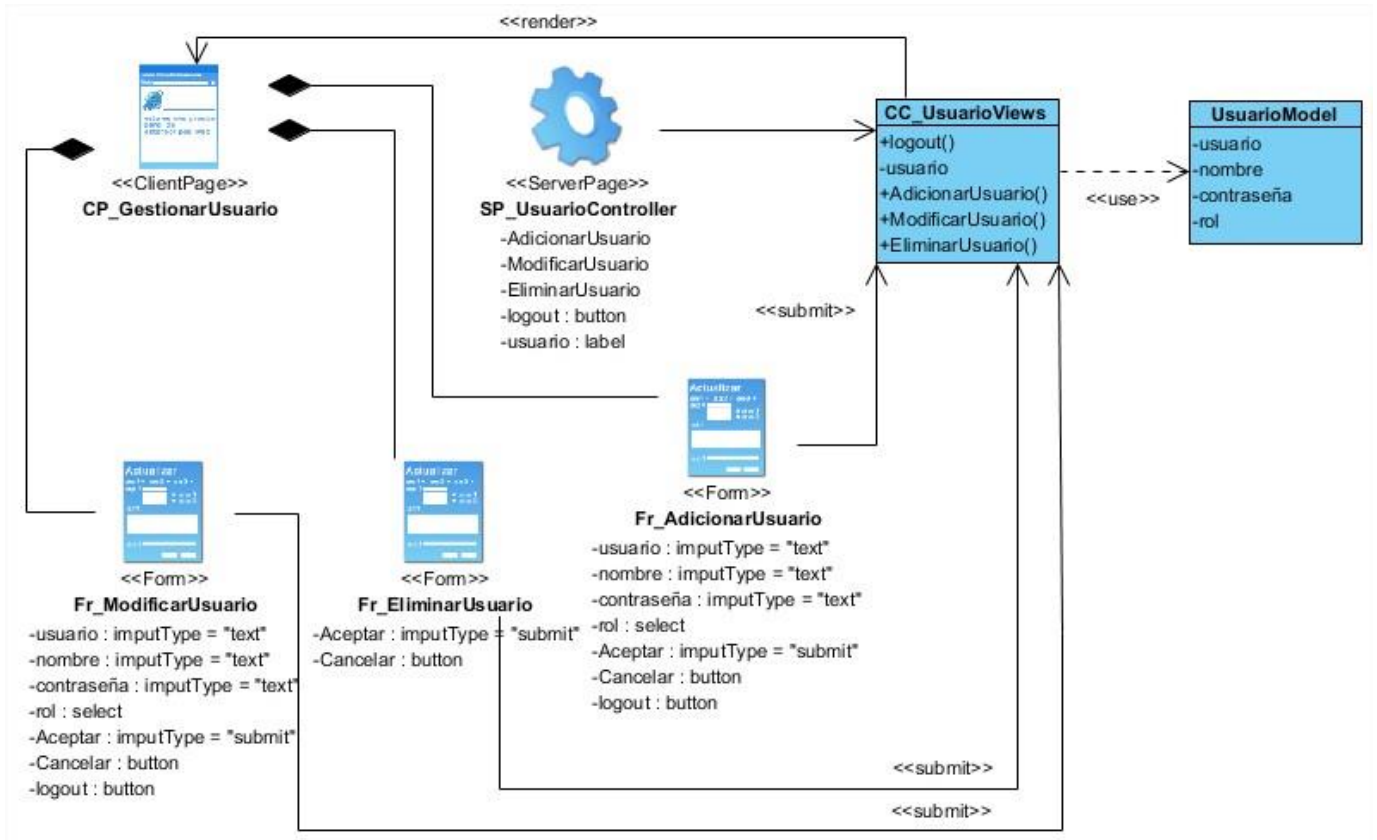


Figura 6 Diagrama de Clases Gestionar Usuario (Elaboración propia).

## 2.6 Diseño Arquitectónico de la aplicación

El diseño arquitectónico nos permite definir cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de este. Es el enlace entre el diseño y la ingeniería de requisitos, que identifica los principales componentes estructurales en un sistema y la relación entre ellos, localizando problemas en el sistema y proporcionando soluciones generales para cada uno.

### 2.6.1 Patrón de arquitectura

El MVC es un estilo de arquitectura de *software* que organiza los componentes de una aplicación, la interfaz de usuario, y la lógica de control en tres paquetes bien definidos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista y procesarlos.

### **Descripción del patrón:**

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y en la vista.

**Ventajas:** Soporte de vistas múltiples. Dado que la vista está separada del modelo y no hay dependencia directa del modelo con respecto a vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.

**Desventajas:** Costo de actualizaciones frecuentes. Desacoplar el modelo de la vista no significa que los sistemas tengan que implementar transformaciones de datos en pasos sucesivos [35].

### **2.6.2 Patrones generales de asignación de responsabilidades (GRASP)**

- **Experto:** básico para asignación de responsabilidades. La clase encargada de crear un objeto es la que conoce todos sus métodos y atributos. De este modo obtendremos mayor cohesión, seguridad y encapsulamiento.
- **Bajo Acoplamiento:** usar la menor interacción o ligamiento entre clases, para crear independencia, así, en caso de modificar alguna clase, esto genera la menor afectación posible en la estructura del sistema.
- **Alta Cohesión:** la información almacenada por cada clase debe ser la correspondiente a esta.

### **2.5 Diagrama de Despliegue**

El modelo de despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. Se utiliza para visualizar la distribución de los componentes de *software* en los nodos físicos. El mismo está compuesto por: Nodos (Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos), Dispositivos (Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela) y Conectores (Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo).

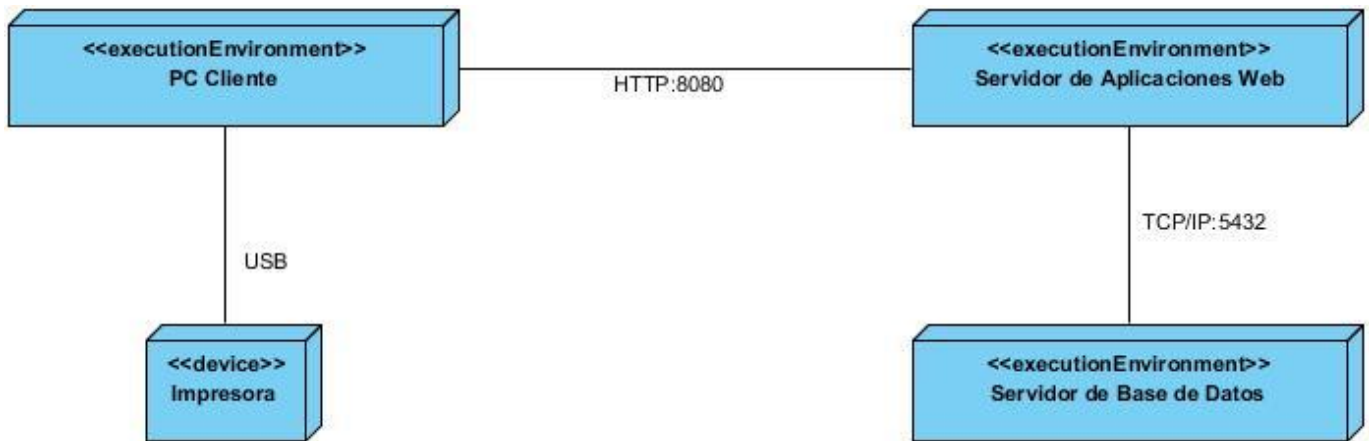


Figura 7 Modelo de Despliegue.

### PC (Cliente)

Es el dispositivo electrónico por el cual el usuario interactúa con la aplicación. El cliente es un ordenador o dispositivo que consume un servicio remoto en otro ordenador conocido como servidor normalmente a través de una red.

### Servidor Web (Apache)

Es el host virtual donde se encuentra el servidor de la aplicación o sistema el cual gestiona la mayor parte de las funciones de acceso a los datos de la aplicación.

### HTTPS - HTTP

Para distinguir una comunicación o página web segura, la *url* debe comenzar con "*https://*" (empleando el puerto 443 por defecto); en tanto la tradicional es "*http://*" (empleando el puerto 80 por defecto). HTTPS fue adoptado como estándar web y opera en la capa más alta del modelo TCP/IP, la capa de Aplicación.

### 2.5.1 Arquitectura

La arquitectura de *software* se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Representa un diseño de alto nivel que tiene dos propósitos primarios: satisfacer los atributos de calidad y servir como guía en el desarrollo [28].



Según Huaman [29]:

“[...]un **patrón arquitectónico** es una solución general y reutilizable a un problema común en la arquitectura de *software* dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de *software*, pero tienen un alcance más amplio.”

Además, constituye una colección de decisiones de diseño arquitectónico, que tiene un nombre específico y son parametrizadas para tener en cuenta diferentes situaciones durante el desarrollo de *software*.

Para el desarrollo del sistema se decide hacer uso del Patrón Arquitectónico Modelo - Vista – Controlador (MVC).

Las principales ventajas del uso del patrón MVC son:

- La separación del Modelo y la Vista, lo cual logra separar los datos, de su representación visual.
- Facilita el manejo de errores.
- Permite que el sistema sea escalable si es requerido.
- Es posible agregar múltiples representaciones de los datos.

La arquitectura Modelo Vista Controlador (Model-View-Controller) es un patrón de diseño de *software* en torno a la interconexión de los tres tipos de componentes principales en un lenguaje de programación, a menudo con un fuerte enfoque en la programación orientada a objetos (POO). Estos tres tipos de componentes son vagamente llamados modelos, vistas y controladores [30].

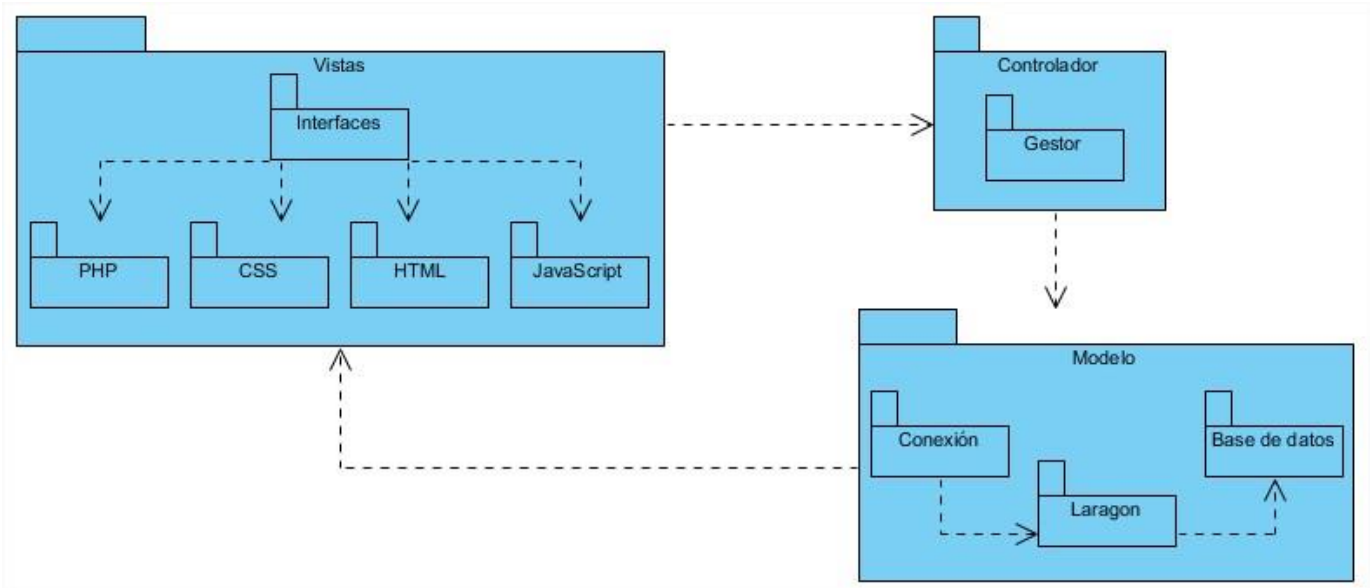


Figura 8 Arquitectura (Elaboración propia).

## 2.6 Conclusiones parciales del Capítulo II.

- En la fase de exploración se realizó la detección de los usuarios a interactuar con el sistema, además se levantaron los requisitos funcionales de los cuales fueron encontrados 11 y no funcionales. Se confeccionaron también las Historias de Usuarios.
- En la fase de planificación se realizó el plan de iteraciones, obteniendo tres iteraciones en el desarrollo del sistema, donde el tiempo máximo de una iteración es de 2 semanas.
- Se describió la arquitectura de *software* MVC, la cual se mantendrá para el presente proyecto.

## Capítulo 3. Implementación y Pruebas del Sistema de gestión de inventario para la Oficina Nacional de Administración Tributaria (ONAT) en Las Tunas.

En el presente capítulo se aborda acerca de los estándares de codificación y las tareas de ingeniería utilizadas para la implementación de la aplicación web. Además de se expone el proceso de prueba al que es sometido el *software*.

### 3.1 Estándares de Codificación

Los estándares de codificación o programación son términos que describen convenciones para escribir código fuente en los lenguajes de programación.

- Utilizar 4 espacios para la sangría.
- Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible.
- La declaración de importación debe escribirse en líneas separadas.
- Poner sufijo *Exception* a todas las excepciones.
- Poner sufijo *View* a las vistas.
- Poner sufijo *Model* a los modelos.
- Poner sufijo *Controller* a las clases controladoras.
- Las variables usaran las letras en minúscula y separando las palabras con un guion bajo.
- Mostrar un mensaje de confirmación antes de eliminar un elemento.
- La contraseña de estar encriptada.
- Usar verbos para los métodos y sustantivos para atributos.
- Escribir comentarios para encontrar código más fácilmente.
- Los estilos van en archivos separados del código HTML.
- Usar nombres de variable significativos como *usuario*, *local* o *inventario* en lugar de *u*, *l* o *i*.

### 3.2 Implementación

Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios.

El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance [31].

### 3.2.1 Tareas de la Iteración 1

El objetivo de esta iteración es crear la aplicación con la base y estructura necesaria para que el administrador pueda crear los usuarios necesarios que operarán el sistema y los mismos puedan autenticarse. Las tareas a realizar son las siguientes:

1. Tarea 1: Realizar sistema de autenticación
2. Tarea 2: Adicionar Usuario
3. Tarea 3: Modificar Usuario
4. Tarea 4: Eliminar Usuario
5. Tarea 5: Listar historial

Tabla 5 Tarea de Ingeniería 2. Adicionar Usuario. (Elaboración propia)

Tarea	
<b>Número de Tarea: 2</b>	<b>Número de la Historia de Usuario: 9</b>
<b>Nombre de la Tarea:</b> Adicionar Usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 16-10-21	<b>Fecha de fin:</b> 16-10-21
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Implementar una vista que permita adicionar un usuario al sistema	

Tabla 6 Tarea de Ingeniería 3. Modificar Usuario. (Elaboración propia)

Tarea	
<b>Número de Tarea: 3</b>	<b>Número de la Historia de Usuario: 10</b>
<b>Nombre de la Tarea:</b> Modificar Usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 17-10-21	<b>Fecha de fin:</b> 17-10-21

<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón
<b>Descripción:</b> Implementar una vista que permita editar un usuario del sistema

Tabla 7 Tarea de Ingeniería 4. Eliminar Usuario. (Elaboración propia)

Tarea	
<b>Número de Tarea:</b> 4	<b>Número de la Historia de Usuario:</b> 11
<b>Nombre de la Tarea:</b> Eliminar Usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 18-10-21	<b>Fecha de fin:</b> 18-10-21
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Implementar una vista que permita eliminar un usuario del sistema	

### 3.2.2 Tareas de la Iteración 2

El objetivo de esta iteración es crear las vistas de la gestión de los equipos. Las tareas trazadas son:

6. Tarea 6: Añadir nuevo equipo al inventario
7. Tarea 7: Editar equipo en el inventario
8. Tarea 8: Eliminar equipo en el inventario
9. Tarea 9: Listar equipo en el inventario
10. Tarea 10: Listar Inventario
11. Tarea 11: Imprimir Inventario

Tabla 8 Tarea de Ingeniería 5. Añadir nuevo equipo al inventario. (Elaboración propia)

Tarea	
<b>Número de Tarea:</b> 6	<b>Número de la Historia de Usuario:</b> 2
<b>Nombre de la Tarea:</b> Añadir nuevo equipo al inventario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 16-10-21	<b>Fecha de fin:</b> 16-10-21
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	

<b>Descripción:</b> Implementar una vista que permita adicionar un nuevo equipo al sistema
--

Tabla 9 Tarea de Ingeniería 6. Editar equipo en el inventario. (Elaboración propia)

Tarea	
<b>Número de Tarea:</b> 7	<b>Número de la Historia de Usuario:</b> 3
<b>Nombre de la Tarea:</b> Editar equipo en el inventario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 17-10-21	<b>Fecha de fin:</b> 17-10-21
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Implementar una vista que permita editar un equipo del sistema	

Tabla 10 Tarea de Ingeniería 7. Eliminar equipo en el inventario. (Elaboración propia)

Tarea	
<b>Número de Tarea:</b> 8	<b>Número de la Historia de Usuario:</b> 4
<b>Nombre de la Tarea:</b> Eliminar equipo en el inventario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día
<b>Fecha de inicio:</b> 18-10-21	<b>Fecha de fin:</b> 18-10-21
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Implementar una vista que permita eliminar un equipo del sistema	

Tabla 11 Tarea de Ingeniería 8. Listar equipo en el inventario. (Elaboración propia)

Tarea	
<b>Número de Tarea:</b> 9	<b>Número de la Historia de Usuario:</b> 5
<b>Nombre de la Tarea:</b> Listar equipo en el inventario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 1 día

<b>Fecha de inicio:</b> 18-10-21	<b>Fecha de fin:</b> 18-10-21
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Implementar una vista que permita listar un equipo según su id.	

### 3.3 Pruebas de aceptación

Las pruebas de aceptación son una parte integral del desarrollo incremental. Todas las historias de usuarios están respaldadas por pruebas de aceptación, que son definidas por el cliente in situ y significaban la satisfacción del mismo con el producto desarrollado. Estas pruebas obligan al cliente a profundizar en el conocimiento de su dominio y declarar con precisión qué debe hacer la aplicación en circunstancias específicas, por esto, el cliente es la persona adecuada para diseñarlas [32]. En efecto, las pruebas de aceptación marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo la dirección a seguir, así como los puntos o funcionalidades en que se debe poner el mayor esfuerzo y atención.

Las pruebas de aceptación tienen una importancia crítica para el éxito de una iteración. Por esta razón, el probador debe diseñar las pruebas de aceptación lo antes posible a partir del comienzo de la iteración. Estas pruebas se presentarán luego al cliente para su aprobación y posteriormente se presentarán al equipo de desarrollo [33].

A continuación, se especifican algunas de las pruebas de aceptación realizadas al sistema:

Pruebas de aceptación para la Iteración 1. Para la iteración 1 se define una prueba de aceptación correspondiente a la HU: Autenticar Usuario la cual se muestra a continuación:

Tabla 12 Prueba de aceptación #1 (Elaboración Propia)

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>Historia de usuario:</b> 1
<b>Nombre:</b> Autenticar usuario	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Autenticar usuario.	
<b>Condiciones de ejecución:</b> El usuario no puede estar autenticado en el sistema.	
<b>Pasos de ejecución:</b>	

1. El usuario debe ubicarse en la pantalla de inicio de sesión.
2. Introducir usuario y contraseña.
3. Acceder al sistema luego de validados los datos.

**Resultados esperados:** Satisfactorio

Pruebas de aceptación para la Iteración 2. Para la iteración 2 se definieron un total de 4 pruebas de aceptación correspondientes a las HU: Añadir nuevo equipo al inventario, Editar equipo en el inventario, Eliminar equipo en el inventario, Listar equipo en el inventario, las cuales se muestran a continuación:

Tabla 13 Prueba de aceptación #2 (Elaboración propia)

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Añadir nuevo equipo al inventario	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Añadir nuevo equipo al inventario.	
<b>Condiciones de ejecución:</b> El usuario tiene que estar autenticado en el sistema con el rol Info_Admin o Info_Munic.	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario debe haber iniciado la sesión con el rol correspondiente.</li> <li>2. Acceder a la sección de gestión de equipo.</li> <li>3. Seleccionar la opción añadir equipo</li> <li>4. Rellenar todos los campos con la información correspondiente.</li> <li>5. Seleccionar la opción Aceptar.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	



Tabla 14 Prueba de aceptación #3 (Elaboración propia)

Caso de prueba de aceptación	
<b>Código:</b> HU3_P1	<b>Historia de usuario:</b> 3
<b>Nombre:</b> Editar equipo en el inventario	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Editar equipo en el inventario.	
<b>Condiciones de ejecución:</b> El usuario tiene que estar autenticado en el sistema con el rol Info_Admin.	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El usuario debe haber iniciado la sesión con el rol correspondiente.</li> <li>2. Acceder a la sección de gestión de equipo.</li> <li>3. Seleccionar un equipo y posteriormente seleccionar la opción Editar.</li> <li>4. Modificar los datos. Seleccionar la opción Aceptar.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	

Tabla 15 Prueba de aceptación #4 (Elaboración propia)

Caso de prueba de aceptación	
<b>Código:</b> HU4_P1	<b>Historia de usuario:</b> 4
<b>Nombre:</b> Eliminar equipo en el inventario	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Eliminar equipo en el inventario.	
<b>Condiciones de ejecución:</b> El usuario tiene que estar autenticado en el sistema con el rol Info_Admin.	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El usuario debe haber iniciado la sesión con el rol correspondiente.</li> <li>2. Acceder a la sección de gestión de equipo.</li> <li>3. Seleccionar un equipo.</li> <li>4. Seleccionar la opción Eliminar.</li> <li>5. Confirmar la eliminación.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	

Tabla 16 Prueba de aceptación #5 (Elaboración propia)

Caso de prueba de aceptación	
<b>Código:</b> HU5_P1	<b>Historia de usuario:</b> 5
<b>Nombre:</b> Listar equipo en el inventario	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Listar equipo en el inventario.	
<b>Condiciones de ejecución:</b> El usuario tiene que estar autenticado en el sistema.	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El usuario debe haber iniciado la sesión.</li> <li>2. Acceder a la sección de gestión de equipo.</li> <li>3. Acceder al botón de búsqueda y rellenar con el número de inventario del equipo deseado a buscar.</li> <li>4. Es mostrado el equipo.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	

Las pruebas de aceptación restantes, correspondientes a **Añadir usuario, Modificar usuario, Eliminar usuario** de la segunda iteración y a la tercera iteración en su totalidad, se pueden consultar en los **Anexos**.

En total se realizaron un total de 8 pruebas de aceptación. Estas fueron realizadas pasando por cada iteración, supervisando de esta manera el cumplimiento de cada tarea de ingeniería correspondiente a estas.

### 3.4 Conclusiones parciales del Capítulo III.

Al culminar el proceso de pruebas por iteración, se detectaron un grupo de no conformidades que se fueron solucionando antes de comenzar la siguiente etapa, permitiendo que al culminar la última iteración de pruebas la plataforma web tuviera un gran por ciento de aceptación ante el cliente. En la primera iteración se detectó una no conformidad relacionada con la interfaz de la página de inicio. Para la segunda iteración se detectaron un total de 3 no conformidades relacionadas con la posición botones, búsqueda de datos y diseño de la interfaz. En la tercera iteración no hubo no conformidades.

## Conclusiones

Luego del estudio realizado se puede concluir que:

- El análisis de los referentes teóricos, los sistemas similares que gestionan y controlan inventarios, y la lentitud y obsoleto proceso de gestión de inventario en la entidad evidenció la necesidad de desarrollar un Sistema que permitiera gestionar el inventario físico existente.
- La utilización del sistema propuesto facilita el manejo de la información relacionada con el inventario físico. Cumpliendo así con las necesidades del cliente. Por lo que se concluye que el producto es sostenible.
- El uso de la metodología AUP-UCI, orientada a equipos de desarrollo pequeños, permitió la obtención de un producto que responde a los objetivos de esta investigación.
- Las tecnologías y herramientas seleccionadas resultaron ser adecuadas para el desarrollo del producto facilitando el proceso de desarrollo y proporcionando mayor mantenibilidad y flexibilidad.
- Con el desarrollo del sistema propuesto se cumplió con el objetivo planteado en la investigación, desarrollándose una herramienta para la gestión del inventario perteneciente a la Oficina Nacional de Administración Tributaria.

## **Recomendaciones**

El resultado obtenido solo una mera versión beta de lo que podría ser la fase final del Sistema, el país también podría beneficiarse de esta investigación e implementar en un único servidor todas las provincias con sus respectivos municipios. Se recomienda:

- Por motivos de integridad y seguridad tener la menos cantidad de usuarios posibles.
- Crear copias de seguridad de la base de datos al menos cada 3 meses luego de cada revisión del inventario para mantener las actualizaciones de los bienes materiales a la par con el departamento de economía.

## Bibliografía

1. Bustos Coral, Holman Darío. 2007. Gestipolis. [En línea] 18 de julio de 2007. [Citado el: 29 de junio de 2021.] <https://www.gestipolis.com/tecnologias-de-gestion-tg>.
2. Universidad Latina de Costa Rica. 2021. Universidad Latina de Costa Rica. Universidad Latina de Costa Rica. [En línea] 2021. [Citado el: 29 de junio de 2021.] <https://www.ulatina.ac.cr/blog/qu3-son-las-tic-y-para-que-sirven.2>.
3. Vega, Mario Cruz. Pixelware. *Pixelware*. [Online] 2021. [Cited: junio 28, 2021.] <https://pixelware.com/sistemas-gestion-documental>.
4. Bastidas Bonilla, Edwin. 2010. *Logística y Abastecimiento*. *logisticayabastecimiento*. [En línea] 2010. [Citado el: 2021 de junio de 25.] <https://logisticayabastecimiento.jimdofree.com/gesti%C3%B3n-de-inventarios/>.
5. Westreicher, Guillermo. 2020. *Control de inventario*. *Economipedia.com*. [En línea] 26 de mayo de 2020. [Citado el: 28 de junio de 2021.] <https://economipedia.com/definiciones/control-de-inventario.html>.
6. Mijangos, Luis Filiberto Lozano. A&T Sorting. *A&T Sorting Company*. [En línea] 18 de marzo de 2021. [Citado el: 10 de noviembre de 2021.] <https://www.atsorting.com/empresa-sorteadora/la-importancia-de-la-gestion-de-inventarios/>.
7. Sepúlveda, Sebastián Avilés. SUMUP. *SUMUP*. [En línea] 2021. [Citado el: 4 de Octubre de 2021.] <https://sumup.es/facturas/glosario/gestion-de-inventarios/>.
8. Dilts, Ben. lucidchart. *lucidchart*. [En línea] Lucid Software Inc., 2021. [Citado el: 13 de noviembre de 2021.] <https://www.lucidchart.com/pages/es/tutorial-diagrama-de-actividades-uml>.
9. Chapaval, Nicole. Platzi. Platzi. [Online] 2017. [Cited: noviembre 4, 2021.] <https://platzi.com/blog/que-es-frontend-y-backend/>.
10. Vega, Freddy. Platzi. Platzi. [Online] 2016. [Cited: noviembre 4, 2021.] <https://platzi.com/clases/1050-programacion-basica/5104-que-es-htmlcssjs/#:~:text=%2F%2F-,HTML%2C%20CSS%20y%20JavaScript%20son%20los%20tres%20lenguajes%20que%20est%C3%A1n,hacer%20interactividad%20con%20esa%20informacion>.
11. Dutta, Dev. Quality devs. *qualitydevs*. [En línea] 5 de agosto de 2019. [Citado el: 10 de noviembre de 2021.] <https://www.qualitydevs.com/2019/08/05/que-es-symfony/>.

12. Norton, Sam. designmodo. designmodo. [Online] junio 17, 2020. [Cited: noviembre 10, 2021.] <https://designmodo.com/bootstrap-5/>.
13. Galeano Luna, Luz Dey and Correa C., Javier Mauricio. Prezi. Prezi. [Online] Universidad Antonio Nariño, 2016. [Cited: noviembre 4, 2021.] <https://prezi.com/j84ywfyzvit/visual-paradigm/>.
14. Robledano, Ángel. OpenWebinars. OpenWebinars S.L. [Online] septiembre 24, 2019. [Cited: noviembre 4, 2021.] <https://openwebinars.net/blog/que-es-mysql/>.
15. Schlueter, Isaac Z. npm Docs. npm Docs. [Online] 2021. [Cited: noviembre 10, 2021.] <https://docs.npmjs.com/cli/v6/commands/npm>.
16. Boix Ruiz, J. 2015. *Desarrollo de una aplicación web para la gestión de un servicio de préstamo de material*. 2015.
17. Trujillo G., B., y otros. *Software para el análisis de la situación de salud en Estomatología. Medicentro Electrónica*. 2016. 2016, Vol. 20(2), págs. 156-158.
18. Khoa, Leo. [Online] marzo 1, 2019. [Cited: noviembre 14, 2021.] <https://laragon.org/docs/index.html>.
19. PRESSMAN, ROGER S. *Software engineering: a practitioner's approach (7th ed.)*. 2010.
20. RODRÍGUEZ S., TAMARA. *Metodología de desarrollo para la actividad productiva de la uci*. 2015.
21. Kay, Summer. ProcessMaker. ProcessMaker. [En línea] 15 de septiembre de 2020. [Citado el: 24 de noviembre de 2021.]
22. Anwar, A. A Review of RUP (Rational Unified Process). 2014. s.l. : *International Journal of Software Engineering (IJSE)*, 2014, Vol. vol. 5.
23. Cortés Bravo, Carlos Alberto, Abud Figueroa, María Antonieta y Romero Torres, Celia. 2016. *Propuesta de un Catálogo de Patrones de Escenario para la Definición de Requisitos*. Guadalajara : s.n., 2016.
24. Bochman, G. Non-Functional Requirements. University of Ottawa. [En línea] 13 de Abril de 2015. [Citado el: 4 de Octubre de 2021.] <https://www.site.uottawa.ca/~bochmann/SEG3101/Notes/SEG3101-ch3-4%20-%20Non-Functional%20Requirements%20-%20Qualities.pdf> Somerville, Ian. 2012. *Ingeniería de software (Spanish Edition)*. s.l. : Universidad Nacional Autónoma de México, 2012. Vol. 9. 9786073206037.
25. Vergara, Sergio. 2021. ITDO. ITDO. [En línea] 10 de Agosto de 2021. [Citado el: 4 de Octubre de 2021.] <https://www.itdo.com/blog/que-son-las-historias-de-usuario/>.

26. Acosta, M. P., & Martínez, C. M. de L. (2015). *Sistema informático de apoyo a los procesos “Evaluación del estudiante” y “Censo, capacidad y matrícula” en la Dirección de la Residencia Estudiantil 2 de la Universidad de las Ciencias Informáticas*. Universidad de las Ciencias Informáticas.
27. ingesoft2. 2014. Ingeniería de software 2. *Ingeniería de software 2-Notas de Ingeniería de software 2*. [En línea] 5 de Agosto de 2014. [Citado el: 4 de Octubre de 2021.] <https://ingenieriasoftware2.wordpress.com/2014/08/05/disenio-de-la-arquitectura/>.
28. huaman, Wilber Ccori. 2018. [En línea] 7 de Septiembre de 2018. [Citado el: 4 de Octubre de 2021.] <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>.
29. Gómez, Rodrigo. 2015. [En línea] 11 de Noviembre de 2015. [Citado el: 4 de Octubre de 2021.] [rodrigogr.com/blog/modelo-vista-controlador/](http://rodrigogr.com/blog/modelo-vista-controlador/).
30. M. Labrador and A. Collazo Garcia, “La programación extrema,” 11 2013.
31. J. J. Gutiérrez, M. J. Escalona, M. Mejías, and J. Torres, “PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA,” p. 12.
32. ELLINGWOOD, Justin, 2017. An Introduction to Continuous Integration, Delivery, and Deployment. *DigitalOcean* (vid. pág. 37).
33. Olanrewaju, Rashidah F. 2015. *An Empirical Study of the Evolution of PHP MVC Framework in Advanced Computer and Communication Engineering Technology*. 2015. págs. 399-410.
34. Labañino Urbina, S., Toledano López, O. G., & Labañino Gainza, C. (2020). Algoritmo de búsqueda fonética para el Sistema de Gestión Integral de Aduanas. *Serie Científica de La Universidad de Las Ciencias Informáticas*, 13(9), 82–96.
35. Mar. “Conoce las 3 metodologías ágiles más usadas.” 2018. [Online]. Available: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>.
36. D. M. Lacal, “Introducción a XP.” Universidad de las Ciencias Informáticas.
37. M.C. Saúl González Campos<sup>1</sup> and M.C. Luis Felipe Fernández Martínez, “Programación Extrema: Prácticas, Aceptación y Controversia,” Universidad Autónoma de Cd. Juárez.
38. Thakur Vaswani, Navin. 2010. *Desarrollo de una aplicación web para la generación de la información curricular*. Madrid : s.n., 2010.
39. Alimadadi, Saba, y otros. *Understanding JavaScript event-based interactions in Proceedings of the 36th International Conference on Software Engineering*. 2014. 31 de mayo de 2014, págs. 367-377.

40. Pressman, R. S. 2005. *Software engineering: a practitioner's approach: Palgrave Macmillan*. 2005.
41. Obe, Regina O. y Hsu, Leo S. 2017. *PostgreSQL: Up and Running: A Practical Introduction to the Advanced Open Source Database*. Segunda. s.l. : O'Reilly Media, Inc., 2017.
42. Nevado Cabello, Maria Antonia. 2013. *Introducción a las Bases de Datos Relacionales*. España : Casa del Libro México, 2013.
43. Guadamillas Herranz, Álvaro. 2014. *Advanced system of planning and organization of classes at the university*. Madrid : s.n., 2014.
44. Maldonado, R. I. Z. *Comparativa sintáctica los lenguajes de programación java y groovy entre*. 2015. s.l.: RECI Revista Electrónica de las Ciencias Computacionales e Informática, 2015, Vol. 4, págs. 76-108.
45. Rodríguez Sánchez, Tamara. Metodología de desarrollo para la Actividad productiva de la UCI. Disponible en: <http://mejoras.prod.uci.cu/>.
46. BELL, DONALD. *UML basics: the component diagram*. 2004 (Disponible en <https://www.fing.edu.uy/inco/cursos/ingsoft/iis/files/uml-component%20diagram.pdf>)
47. DIMITRIJEVIC, SONJA; JOVANOVIC, JELENA; DEVEDZIC, VLADAN. *A comparative study of software tools for user story management*. 2015 (Disponible en <https://www.sciencedirect.com/science/article/abs/pii/S0950584914001293>)
48. ERICKSSON, ULF. *The difference between functional and non-functional requirements*. 2015 (Disponible en <https://reqtest.com/requirements-blog/understanding-the-difference-between-functional-and-non-functional-requirements/>)
49. ESCALONA, MARÍA JOSÉ; KOCH, NORA. *Ingeniería de requisitos en aplicaciones para la web – un estudio comparativo*. 2002 (Disponible en <http://www.lsi.us.es/docs/informes/lsi-2002-4.pdf>)
50. FLANAGAN, DAVID. *JavaScript - The definitive guide (7th Ed.)*. 2020 (Disponible en <https://books.google.com.cu/>)
51. HERNÁNDEZ, E. O. *El Lenguaje Unificado de Modelado (UML)*. 2015 (Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>)
52. ISLAS HERNANDEZ, LUIS. *Análisis y diseño orientado a objetos. Apuntes digitales*. 2014 (Disponible en <http://cidecame.uaeh.edu.mx/lcc/mapa/proyecto/libro10/index.html>)
53. ITURRALDE, OSCAR J. *Introducción a Los Patrones de Diseño: Un Enfoque Práctico*. 2016



54. LAPLANTE, PHILLIP A. *Requirements engineering for software and systems*. 2009 (Disponible en <https://content.taylorfrancis.com/books/download?dac=c2012-0-02503-5&isbn=9781466560826&format=googlepreviewpdf>)
55. LÓPEZ MARTÍNEZ, J. A. *Sistema para la gestión de oportunidades de negocio para el centro de informática industrial*. 2015
56. MACRAE, CALLUM. *Vue.js: Up and Running: Building Accessible and Performant Web Apps*. 2018 (Disponible en <https://books.google.com.cu>)
57. MCGRATH, MIKE. *HTML, CSS & JavaScript in easy steps*. 2020. (Disponible en <https://books.google.com.cu/>)
58. MENENDEZ ALONSO, EVELYN. *Introducción a herramientas case*. 2011.
59. RUMBAUCH JAMES, J. I. y GRADY, B. *El Lenguaje Unificado de Modelado. Manual de Referencia*. 2015

## Anexos

Tabla 17 HU Autenticar Usuario (Elaboración propia)


<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Usuario:</b> Info_Admin, Info_Munic, Economico
<b>Nombre:</b> Autenticar Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> El usuario podrá iniciar sesión para tener acceso a las funcionalidades asociadas al su rol y desempeñar sus funcionalidades. Según su rol se les será mostrada distinta información.	
 <p>El prototipo de interfaz para la historia de usuario 'Autenticar Usuario' muestra un formulario con dos campos de entrada: 'Usuario:' y 'Contraseña:'. Cada campo tiene un recuadro rectangular para el texto. Debajo de los campos, hay un botón rectangular con el texto 'Acceder' en blanco sobre un fondo oscuro.</p>	
<i>Figura 5 Prototipo de interfaz HU Autenticar Usuario (Elaboración propia)</i>	

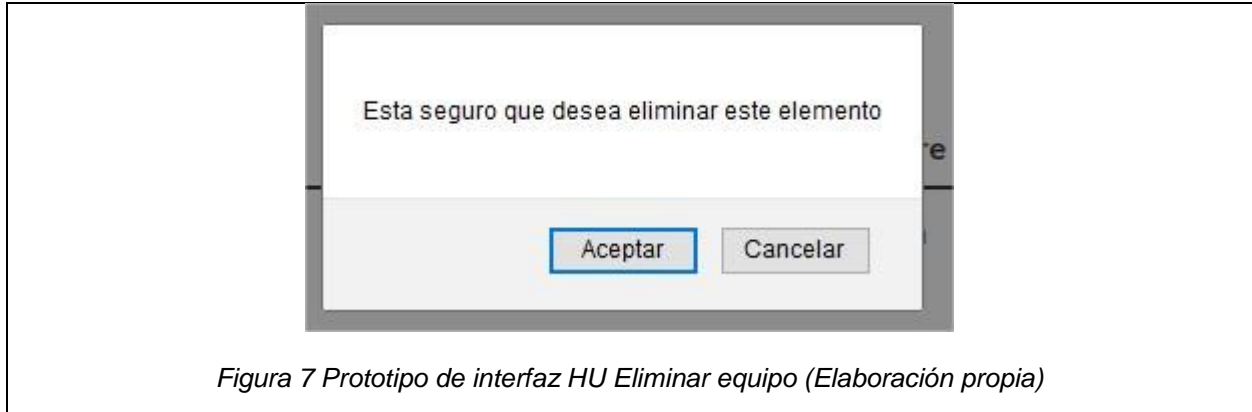
Tabla 18 HU Editar equipo (Elaboración propia)

<b>Historia de Usuario</b>	
<b>Número:</b> 3	<b>Usuario:</b> Info_Admin
<b>Nombre:</b> Editar equipo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>

<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón
<b>Descripción:</b> El administrador podrá modificar un equipo que se encuentre en el sistema.
<h2>Editar Equipo</h2>
Inventario <input type="text" value="108976"/>
Estado <input type="text" value="Bueno"/>
Cantidad <input type="text" value="1"/>
Local <input type="text" value="ONAT Provincial"/>
<input type="button" value="Cancelar"/> <input type="button" value="Aceptar"/>
<i>Figura 6 Prototipo de interfaz HU Editar equipo (Elaboración propia)</i>

Tabla 19 HU Eliminar equipo (Elaboración propia)

<b>Historia de Usuario</b>	
<b>Número:</b> 4	<b>Usuario:</b> Info_Admin
<b>Nombre:</b> Eliminar equipo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> El administrador podrá eliminar un equipo que se encuentre en el sistema.	



*Tabla 20 HU Listar equipo (Elaboración propia)*

<b>Historia de Usuario</b>	
<b>Número:</b> 5	<b>Usuario:</b> Info_Admin, Info_Munic, Economico
<b>Nombre:</b> listar equipo	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> El usuario podrá buscar un equipo con un número de inventario en específico.	

*Tabla 21 HU Listar historial (Elaboración propia)*

<b>Historia de Usuario</b>	
<b>Número:</b> 6	<b>Usuario:</b> Info_Admin, Info_Munic, Economico
<b>Nombre:</b> listar historial	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Muestra el historial de acciones realizado por los usuarios en el último trimestre	

Tabla 22 HU Listar inventario (Elaboración propia)

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> Info_Admin, Info_Munic, Economico
<b>Nombre:</b> listar inventario	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Muestra un listado de todos los equipos con su número de inventario, estado, cantidad, local.	

Tabla 23 HU Imprimir inventario (Elaboración propia)

Historia de Usuario	
<b>Número:</b> 8	<b>Usuario:</b> Info_Admin, Info_Munic, Economico
<b>Nombre:</b> Imprimir inventario	
<b>Prioridad en negocio:</b> Medio	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> Imprime un listado de todos los equipos con su número de inventario, estado, cantidad, local.	

Tabla 24 HU Añadir Usuario (Elaboración propia)

Historia de Usuario	
<b>Número:</b> 9	<b>Usuario:</b> Info_Admin
<b>Nombre:</b> Añadir Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> El administrador podrá adicionar un usuario con los campos usuario, nombre, rol y contraseña.	



The image shows a user registration form with the following fields and buttons:

- Username:** Tunas
- Roles:** ROLE\_INFO\_MUNICIPIO
- Password:** 1234
- Name:** Alberto
- Buttons:** Cancelar (red) and Guardar (dark blue)

*Figura 8 Prototipo de interfaz HU Añadir Usuario*

Tabla 25 HU Editar Usuario (Elaboración propia)

<b>Historia de Usuario</b>	
<b>Número:</b> 10	<b>Usuario:</b> Info_Admin
<b>Nombre:</b> Editar usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> El administrador podrá modificar un usuario que se encuentre en el sistema.	

Username  
 Municipio

Roles  
 ROLE\_INFO\_MUNICIPIO

Password  
 \$2y\$13\$1U20o3qxY05tpZvf6dQm6Op6TCuKsMXsPM/nevQQNU/FjJIGooNYa

Name  
 Otro prueba

Cancelar    Aceptar

*Figura 9 Prototipo de interfaz HU Editar Usuario (Elaboración propia)*

Tabla 26 HU Eliminar Usuario (Elaboración propia)

<b>Historia de Usuario</b>	
<b>Número:</b> 11	<b>Usuario:</b> Info_Admin
<b>Nombre:</b> Eliminar equipo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b> Nelson Alejandro Quiñones Remón	
<b>Descripción:</b> El administrador podrá eliminar un usuario que se encuentre en el sistema.	

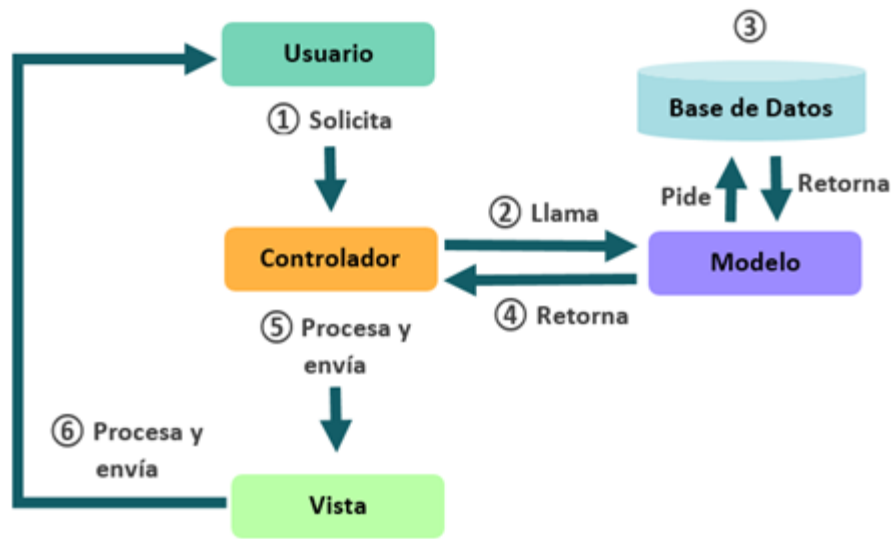
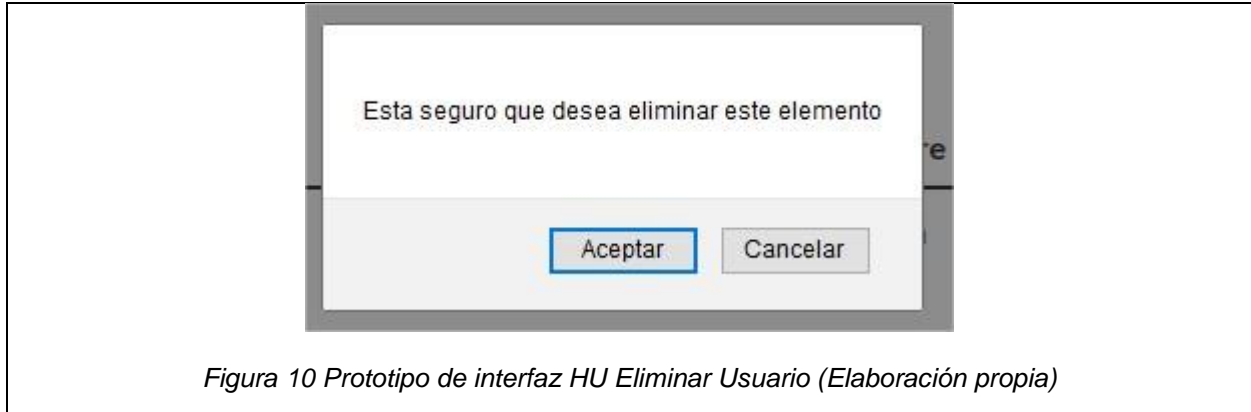


Figura 9 Modelo Vista Controlador [34]

Tabla 27 Prueba de aceptación #6 (Elaboración propia)

Caso de prueba de aceptación	
Código: HU9_P1	Historia de usuario: 9
Nombre: Añadir usuario	
Descripción: Prueba realizada a la funcionalidad: Añadir usuario.	
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema con el rol Info_Admin.	
Pasos de ejecución:	
6. El usuario debe haber iniciado la sesión con el rol correspondiente. 7. Acceder a la sección de gestión de usuarios.	



<p>8. Seleccionar la opción añadir usuario.</p> <p>9. Rellenar los campos con la información correspondiente.</p> <p>10. Seleccionar la opción Aceptar.</p>
<p><b>Resultados esperados:</b> Satisfactorio</p>

Tabla 28 Prueba de aceptación #7 (Elaboración propia)

Caso de prueba de aceptación	
<b>Código:</b> HU10_P1	<b>Historia de usuario:</b> 7
<b>Nombre:</b> Modificar usuario	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Modificar usuario.	
<b>Condiciones de ejecución:</b> El usuario tiene que estar autenticado en el sistema con el rol Info_Admin.	
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. El usuario debe haber iniciado la sesión con el rol correspondiente.</li> <li>2. Acceder a la sección de gestión de usuarios.</li> <li>3. Seleccionar un usuario y posteriormente seleccionar la opción Editar.</li> <li>4. Modificar los datos. Seleccionar la opción Aceptar.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	

Tabla 29 Prueba de aceptación #8 (Elaboración propia)

Caso de prueba de aceptación	
<b>Código:</b> HU11_P1	<b>Historia de usuario:</b> 11
<b>Nombre:</b> Eliminar trabajador	
<b>Descripción:</b> Prueba realizada a la funcionalidad: Eliminar usuario.	
<b>Condiciones de ejecución:</b> El usuario tiene que estar autenticado en el sistema con el rol Info_Admin.	
<p><b>Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>6. El usuario debe haber iniciado la sesión con el rol correspondiente.</li> <li>7. Acceder a la sección de gestión de usuarios.</li> <li>8. Seleccionar un usuario.</li> </ol>	

9. Seleccionar la opción Eliminar.

10. Confirmar la eliminación.

**Resultados esperados:** Satisfactorio

**Entrevista al cliente; Julio Yokiro Quiñones (Jefe del departamento de informática de la ONAT provincial en Las Tunas):**

- ¿Cuál es la principal necesidad informática que posee la entidad?

Actualmente la ONAT está pasando por un proceso de firma digital y es necesario informatizar la mayoría de los procesos que se archivan con el formato en papel, para evitar pérdidas de información y tiempo. Los de mayor prioridad son el portal web de la ONAT en la provincia, aunque existe el portal web de la ONAT para el país la provincia necesita de su propio portal para informar a la población y nuestros clientes sobre los cambios y eventos propios de la provincia. También es necesario informatizar el proceso de gestión del inventario.

- Hábleme más sobre la necesidad de informatizar la gestión del inventario.

Desde hace un año se está viendo la posibilidad de implementar un sistema para gestionar el inventario de equipos electrónicos en la provincia y la empresa de desarrollo de software DESOFT estaba dispuesta a implementar el sistema, pero actualmente la ONAT no posee los fondos para pagar tal petición. Actualmente el proceso se está realizando de forma manual y archivándose en formato duro (papel) y por cuestiones naturales al pasar el tiempo este formato se deteriora y en ocasiones resulta en pérdida de la información lo que genera un costo de tiempo que podría ser utilizado en algo de mayor productividad.

- ¿Cuáles son las principales funciones que debería poseer el sistema de gestión que necesita la ONAT?

Como es un sistema para gestionar información y priorizar la actualización de la misma es necesario poder añadir, editar y eliminar las características que poseen los equipos de la entidad en la provincia, de esos equipos es necesario saber su número de inventario, estado y localización, en algunas ocasiones suele suceder que estos equipos pueden ser almacenados en cantidades mayores a 1 por lo que esa podría ser una característica necesaria.

- ¿A parte de la gestión de esos equipos existe alguna funcionalidad que sea necesaria implementar, aunque no tenga tanta prioridad?

De ser posible también sería necesario gestionar los usuarios y darle roles acordes a su cargo en la entidad. Además de poder imprimir dicho inventario para facilitar el trabajo al departamento de economía que es el encargado de llevar un seguimiento del inventario general de la entidad.

Para mí como jefe del departamento sería necesario poder acceder al historial de acciones para monitorear el correcto cumplimiento de los roles y que no haya infracciones por parte de los usuarios.