



TRABAJO FIN DE GRADO
INGENIERÍA EN INFORMÁTICA

NO-INVENTORY

Sistema de gestión para inventarios

Autor

César Hugo Bárzano Cruz

Directores

Nombre Apellido1 Apellido2 (tutor1)

Nombre Apellido1 Apellido2 (tutor2)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, mes de 201



Título del proyecto

Subtítulo del proyecto.

Autor

Nombre Apellido1 Apellido2 (alumno)

Directores

Nombre Apellido1 Apellido2 (tutor1)

Nombre Apellido1 Apellido2 (tutor2)

NO-INVENTORY: Sistema de gestión para inventarios

César Hugo Bárzano Cruz

Gestión, Almacén , Activos, Eficiencia, Ahorro, Informes, **Palabras clave:** Gestión_clave1, Almacen_clave2, Activos_clave3, Eficiencia_clave4, Ahorro_clave5 Informes _clave6

Resumen

Este proyecto surge a raíz de un problema real presentado por la Oficina de Software Libre. Desde principios de 2012 y a partir de un acuerdo con la Unidad de Calidad, la oficina se encarga de recoger material informático procedente de los distintivos organismos de la Universidad de Granada. Dicho material ha alcanzado una gran cantidad, por lo que es necesario un sistema para catalogarlo y procesarlo.

Partiendo de este problema, existen necesidades similares en empresas, organizaciones, instituciones, comercios y todo tipo de negocio con un almacén de activos. En dichos almacenes, las tareas de gestión suelen ser ineficientes, complejas y costosas. El tiempo y dinero que conllevan estas tareas suele ser un factor a tener en cuenta, ya que una gran mayoría de entidades, con almacenes pequeños o medianos, no utilizan un sistema comercial debido a los costes que suponen su mantenimiento. Este problema, sigue presente en entidades con almacenes de gran tamaño, ya que el número de empleados y el tiempo necesario para las tareas básicas de catalogación y administración suponen un coste a tener en cuenta.

Otro problema con el que se encuentran los empleados de estas entidades es la generación de informes. Recopilar información de los activos del almacén con el objetivo de representarla de manera adecuada a las necesidades de cada negocio, puede llegar a ser una tarea complicada.

Como solución a estos problemas, surge el sistema de gestión NO-INVENTORY que pretende mejorar estas tareas, con el objetivo de ahorrar tiempo y dinero al cliente, automatizando y facilitando las tareas administrativas.

La piedra angular del sistema es una plataforma web alojada en la nube por lo que no es necesaria la instalación de ningún software adicional en las máquinas de la entidad que desee comenzar a utilizarlo, solo es necesario un navegador. En dicha plataforma, las tareas de administración se realizan de manera intuitiva permitiendo a los distintos empleados trabajar de forma cooperativa utilizando un sistema de tag's personalizables en función de las características con las que se quieran clasificar los elementos. Facilita la agrupación de elementos con o sin propiedades comunes en colecciones denominadas catálogos en función de las necesidades de cada cliente y uti-

liza dichas colecciones para la generación automática de informes y gráficos representativos del estado del almacén.

El sistema cuenta con una aplicación Android como extensión para realizar tareas de catalogación y clasificación dentro del propio almacén. La ventaja de esto reside en que hoy en día casi todo el mundo cuenta con un smartphone, dando la posibilidad a los empleados de llevar la gestión del almacén con sus propios dispositivos.

En función de los activos que formen el almacén, y del presupuesto que se quiera dedicar a etiquetar cada elemento, el cliente puede decidir que método utilizar, ya que la funcionalidad de la extensión Android es la de lector/escritor de los identificadores de cada elemento, con soporte para:

1. Códigos de Barras
2. Códigos QR
3. Etiquetas NFC

En resumen, el sistema pretende reducir tiempo y costes a las entidades que decidan implantarlo en su almacén con el objetivo de optimizar las tareas de los empleados consiguiendo así un mayor rendimiento.

Por último, resaltar que todo el código desarrollado para el sistema, está liberado en Github, para que pueda ser consultado o mejorado por la comunidad de software libre.

Project Title: Project Subtitle

First name, Family name (student)

Keywords: Keyword1, Keyword2, Keyword3,

Abstract

Write here the abstract in English.

Yo, **César Hugo Bárzano Cruz**, alumno de la titulación TITULACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77138361h, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Nombre Apellido1 Apellido2

Granada a X de mes de 201 .

D. **Nombre Apellido1 Apellido2 (tutor1)**, Profesor del Área de XXXX del Departamento YYYY de la Universidad de Granada.

D. **Nombre Apellido1 Apellido2 (tutor2)**, Profesor del Área de XXXX del Departamento YYYY de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Título del proyecto, Subtítulo del proyecto***, ha sido realizado bajo su supervisión por **Nombre Apellido1 Apellido2 (alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de mes de 201 .

Los directores:

Nombre Apellido1 Apellido2 (tutor1) **Nombre Apellido1 Apellido2 (tutor2)**

Agradecimientos

Poner aquí agradecimientos...

Índice general

1. Introducción	15
2. Análisis y Estado del Arte	17
2.1. Requisitos de Información Plataforma web	17
2.2. Requisitos Funcionales Plataforma web	18
2.3. Requisitos No Funcionales Plataforma web	19
2.4. Requisitos No Funcionales Plataforma web	20
3. Resolución del trabajo	23
3.1. Recursos	23
3.2. Especificación de requisitos	23
3.3. Planificación	24
3.4. Análisis funcional	24
3.5. Implementación y pruebas	25
4. Conclusiones y trabajo futuro	27
5. Bibliografía	29
6. Anexo	31

Capítulo 1

Introducción

Este proyecto tiene el objetivo de solucionar un problema real. Desde principios de 2012, la Oficina de Software Libre (OSL) tiene un acuerdo con Unidad de Calidad de la Universidad de Granada. Dicho acuerdo establece que la oficina es la encargada de recoger el material informático procedente de los distintivos organismos de la Universidad. Debido a la acumulación de material a lo largo de estos 5 años, la oficina de software libre necesita un sistema de gestión que le permita catalogar y clasificar este material.

El problema principal radica en el desconocimiento del estado en el que este material es recogido, es decir, la OSL no sabe si el material recibido funciona, no funciona, necesita ser actualizado o directamente es inservible y ha de ser reciclado. Hay que tener en cuenta el estado actual del almacén. Tras acumular material a lo largo de estos años, el almacén sufre una situación caótica debido al gran número de equipos sin catalogar ni reflejar el estado en el que se encuentran.

Por otra parte, el acuerdo mencionado anteriormente, obliga a la oficina a realizar informes cada cierto periodo de tiempo, reflejando el estado de las campañas de recogida, el estado del material recolectado, el peso...etc

El problema presentado no es algo aislado a la OSL, gran parte de las pequeñas y medianas empresas con almacenes que gestionar presentan problemas similares. Para las empresas, no usar plataformas de gestión para sus productos supone carencia de estrategias de crecimiento, una inadecuada utilización de las tecnologías y conocimientos, pérdidas de recursos, debilidad financiera y deficiencias en toda la organización.

En los últimos años, el análisis de la información asociada a los recursos o servicios de las empresas, ha tomado una gran importancia. El análisis de la información se ha convertido en un factor clave para la toma de decisiones en diversos sectores. El valor de la información ha alcanzado el nivel de otros recursos como por ejemplo los recursos financieros, materiales y humanos. El conocimiento del entorno y la información que podemos aprovechar de él son vitales para la toma de decisiones, ya sea fuera del entorno empresarial o

dentro de él. Decisiones como por ejemplo entrar en nuevos mercados, proteger la empresa de agentes externos, innovación, desarrollo, inversiones...etc. Por ello las Tecnologías de Información forman un factor decisivo para dar lugar al crecimiento tanto de las PyMEs como de cualquier empresa.

Lo principales motivos por los que estas empresas no utilizan sistemas de gestión para sus productos se deben a los costes elevados, a la carencia de recursos, la falta de acceso a la información, etc...

Las Tecnologías de la Información y Comunicaciones (TIC), se han convertido en la herramienta perfecta para llevar a cabo las tareas necesarias para aumentar la eficacia y eficiencia en el manejo de los recursos de la empresa.

Capítulo 2

Análisis y Estado del Arte

Para realizar un mejor Analisis del sistema, vamos a desglosarlo en dos partes. Primero analizaremos la plataforma web y despues analizaremos la aplicación Android

2.1. Requisitos de Información Plataforma web

Los requisitos de información se caracterizan por reunir la información relevante para el cliente que debe gestionar y almacenar el sistema software.

RI-1. Item: Representación de cada uno de los elementos en el sistema. Contenido: nombre, descripción, fecha de alta, ID, Localizador, organización de la que forma parte, el usuario que lo dio de alta, atributos. Los atributos son una tupla de 3 valores asignados en función de las preferencias con las que se quiera clasificar cada elemento. Por ejemplo lugar del que procede, tipo de elemento y estado del elemento.

RI-2. Catalogo: Representación de cada una de las colecciones de elementos del sistema. Contenido: nombre, descripción, fecha de alta, ID, mensaje de aviso, fecha de expiración, tipo de catalogo, organización de la que forma parte, el usuario que lo dio de alta, colección con las ID de los elementos que forman el catalogo.

RI-3. Gráfico: Información del estado del sistema en forma de grafico, en función de los catálogos creados por los distintos usuarios de una misma organización. Gráficos representativos basados en los atributos de los items que forman cada catalogo. Comparativa entre unidades y peso de esas unidades en el catalogo. Contenido: catalogo.

RI-4. Informe: Información del estado del sistema en forma de documento. Generado en función de los catálogos creados por los distintos usuarios de una misma organización y de los gráficos representativos. Contenido: nombre, fecha de creación, usuario, organización, datos del informe.

RI-5. Perfil de Usuario: Información sensible del usuario que va a trabajar con el sistema. Contenido: nombre de usuario, contraseña, correo

electrónico, organización.

RI-6. Preferencias: Información de los atributos que se desean utilizar para clasificar los elementos del sistema dentro del entorno colaborativo de la organización. Contenido: archivo csv de preferencias 1, archivo csv de preferencias 2, archivo csv de preferencias 3.

2.2. Requisitos Funcionales Plataforma web

Como se define en la ingeniería de requisitos, los requisitos funcionales establecen los comportamientos del sistema.

RF-1. Gestión de items: El sistema deberá gestionar el ciclo de vida de items o elementos existentes en el sistema.

RF-1.1. El sistema permitirá añadir items

RF-1.2. El sistema permitirá buscar items

RF-1.3. El sistema permitirá modificar items

RF-1.4. El sistema permitirá eliminar items

RF-1.5. El sistema permitirá agrupar items en colecciones

RF-2. Gestión de catálogos: El sistema deberá gestionar el ciclo de vida de conjuntos de items, estas colecciones reciben el nombre de catálogos o agrupaciones de items con atributos u objetivos similares.

RF-2.1. El sistema permitirá añadir catálogos

RF-2.2. El sistema permitirá buscar catálogos

RF-2.3. El sistema permitirá modificar catálogos

RF-2.4. El sistema permitirá eliminar catálogos

RF-2.5. El sistema permitirá limpiar catálogos

RF-2.6. El sistema permitirá eliminar un item del catalogo

RF-2.7. El sistema permitirá exportar los códigos qr de sus elementos

RF-2.8. El sistema permitirá exportar los códigos de barras de sus elementos

RF-2.9. El sistema permitirá controlar el peso de los elementos del catálogo

RF-2.10. El sistema permitirá controlar el número de unidades del catálogo.

RF-3. Gestión de Gráficos: El sistema deberá generar gráficos a raíz de los elementos que forman parte de un catálogo, mostrando una comparativa entre el numero de unidades y peso de dichas unidades. Los gráficos representativos se dividen en función de las preferencias del usuario.

RF-3.1. El sistema permitirá generar gráficos RF-3.1. El sistema permitirá exportar gráficos RF-3.1. El sistema permitirá seleccionar el tipo de gráfico

RF-4. Gestión de Informes: El sistema deberá generar informes significativos a raíz de los elementos que forman parte de un catálogo. El sistema proveerá al usuario de una plantilla editable para personalizar los aspectos e

RF-4.1. El sistema permitirá generar informes RF-4.2. El sistema permitirá almacenar informes RF-4.3. El sistema permitirá modificar informes RF-4.4. El sistema permitirá eliminar informes RF-4.5. El sistema permitirá añadir catálogos a los informes RF-4.6. El sistema permitirá añadir gráficos a los informes RF-4.7. El sistema permitirá exportar informes

RF-5.1. El sistema permitirá cargar las preferencias desde fichero CSV

RF-6.1. El sistema permitirá al administrador dar de alta a nuevos usuarios. RF-6.1. El sistema permitirá al administrador asignar perfiles de usuario. RF-6.2. El sistema permitirá iniciar sesión a los usuario. RF-6.2. El sistema permitirá cerrar sesión a los usuario. RF-6.2. El sistema registrará internamente las acciones realizadas por cada usuario, de cada organización.

Los requisitos no funcionales, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar por el sistema, sino características de funcionamiento.

RNF-2 Necesitaremos analizar la cantidad de elementos que una organización va a administrar. Si es considerada una cantidad importante, dicha organización recibirá servicio desde una imagen del sistema asilada (Docker). De esta forma conseguimos que el flujo de datos no afectes a la experiencia de los clientes.

DIAGRAMAAAAAAAAAAAAOooooooooooooooooooooooooooooSSS

2.4. Requisitos Funcionales Aplicación Android

Los requisitos funcionales de la aplicación Android establecen los comportamientos o funcionalidades de la aplicación.

RF-1. Administración de items La aplicación deberá administrar los items que formen parte del almacén.

RF-1.1. La aplicación permitirá detectar items mediante texto o localizador RF-1.2. La aplicación permitirá detectar items mediante código QR RF-1.3. La aplicación permitirá detectar items mediante código de barras RF-1.4. La aplicación permitirá detectar items mediante etiquetas NFC RF-1.5. La aplicación permitirá escribir el localizador de un item en una etiqueta NFC RF-1.6. La aplicación permitirá consultar un item en detalle RF-1.7. La aplicación permitirá añadir un item detectado a un catálogo determinado RF-1.8. La aplicación permitirá eliminar items

RF-2. Administración de Catálogos La aplicación deberá administrar los catálogos que se utilicen para clasificar los items del almacén.

RF-2.1. La aplicación permitirá detectar catálogos mediante texto o ID RF-2.2. La aplicación permitirá añadir items al catálogo mediante código QR RF-2.3. La aplicación permitirá añadir items al catálogo mediante código de barras RF-2.4. La aplicación permitirá añadir items al catálogo mediante etiquetas NFC RF-2.5. La aplicación permitirá eliminar items de un catálogo RF-2.6. La aplicación permitirá limpiar la lista de items de un catálogo

RF-3. Sesión de Usuario La aplicación deberá identificar a cada usuario de manera única para certificar las tareas de administración que se lleven a cabo desde la aplicación.

RF-3.1. La aplicación permitirá al usuario iniciar sesión. RF-3.2. La aplicación permitirá al usuario cerrar sesión.

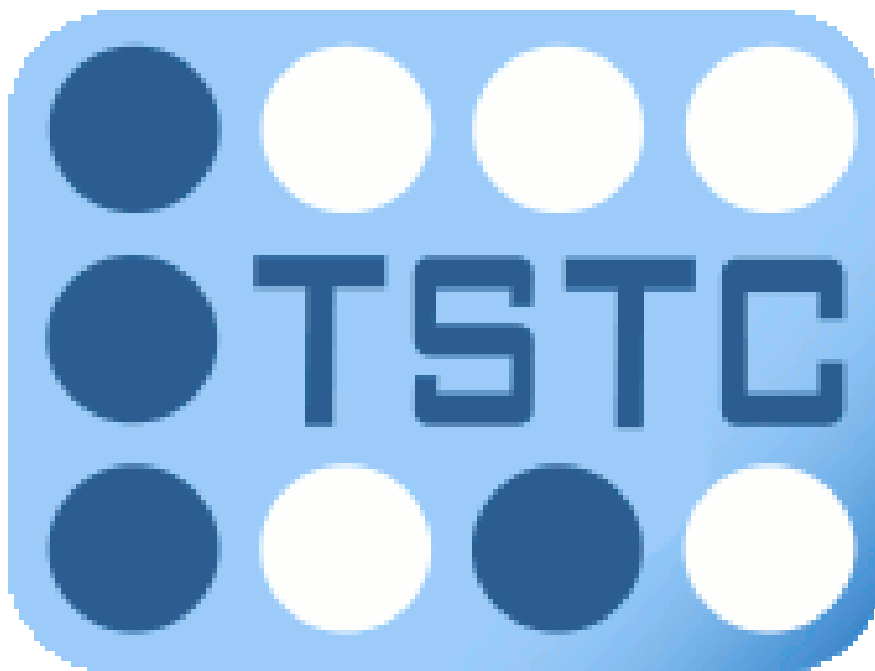


Figura 2.1: man apt-cache

Capítulo 3

Resolución del trabajo

Como método de ingeniería del software decir que vamos a seguir la técnicas de modelo de prototipos rápido o también llamado modelado de prototipado rápido.

Ver http://www.ecured.cu/index.php/Modelo_de_Prototipos

3.1. Recursos

Decir los recursos humanos (autor y directores), hardware y software que se van a utilizar.

3.2. Especificación de requisitos

Decir que se partió de una especificación inicial de requisitos que a medida que se fueron implementando los prototipos se fue refinando posteriormente. Se puede poner la inicial y la final o solo la final indicando que se están poniendo los requisitos que finalmente tiene que tener el sistema.

Los requisitos se pueden referir a las necesidades del usuario del sistema (requisitos del usuario), a lo que tiene que hacer la aplicación (requisito funcional) o a cómo tiene que hacerlo (requisito no funcional). Ejemplo:

En este sistema un robot tiene que coger con sus pinzas un envase de medicamento y llevárselo a una persona anciana que por sí misma no puede recordar su medicación.

Requisitos del usuario:

RU1. La persona puede moverse libremente por una habitación donde está el robot.

RU2. La persona es capaz de coger el medicamento cuando se lo ofrece el robot.

RU3. La persona es capaz de tomarse el medicamento por sí misma.

Requisitos funcionales.

RF1. El robot mediante la cámara kinect debe poder localizar a la persona.

RF2. El robot conoce la posición de la mesa pues tiene un mapa de la habitación.

RF3. El robot debe identificar el medicamento correcto según un plan de medicación previamente establecido.

RF4. El robot debe poder coger el medicamento con sus pinzas.
etc...

Requisitos no funcionales

RNF1. El robot no puede atropellar ni dañar a la persona en ningún momento.

RNF2. La aplicación debe ejecutarse en entornos linux

RNF3. La aplicación debe utilizar pocos recursos para reaccionar con rapidez.

algo de la interfaz, como tratar posibles fallos, etc.

3.3. Planificación

Poner una tabla de tiempos con las planificación del proyecto diciendo cuando se tiene previsto alcanzar cada subobjetivo planteado. Con su correspondiente división en fases y tareas, y la posterior comparación con los datos reales obtenidos tras realizar el proyecto. Entre las fases está la realización de los diferentes prototipos I, II y III por ejemplo.

Poner presupuesto según horas de trabajo estimadas.

3.4. Análisis funcional

A partir de aquí nos referimos solamente al prototipo final que da lugar a la aplicación final.

Hay que describir la funcionalidad que debe poseer el sistema para poder cumplir con los objetivos y requisitos que se han dicho previamente. La descripción de esta funcionalidad puede hacerse analizando las tareas (que aparecerán en la planificación) y estudiando la inter-relación entre ellas y sus conexiones.

Para la realización de este análisis se pueden utilizar Diagramas de Flujo para poder conocer generalmente un único punto de inicio y un único punto de término o en varios.

https://es.wikipedia.org/wiki/Diagrama_de_flujo

Se pueden plantear también casos de uso. Los diagramas de casos de uso sirven para describir la inter-relación entre el sistema y el usuario del mismo. Se pueden utilizar para plantear diferentes casos de interacción entre el robot y la persona y cómo tiene que reaccionar el sistema en cada caso.

https://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso

3.5. Implementación y pruebas

Decir qué lenguaje de programación se ha utilizado y las tecnologías implicadas aunque se hayan comentado en el apartado de recursos. Justificar su uso (rendimiento, disponibilidad, etc.). Si se ha usado open source decirlo y explicar las ventajas.

Las pruebas se realizan para comprobar la verificación y validación del producto software. La verificación consiste en comprobar que el producto realiza lo que está programado, es decir la programación no tiene errores y funciona en todos los casos cumpliendo los requisitos. La validación tiene que ver con que cumpla con lo que espera el usuario.

Verificación y Validación: Conjunto de procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes.

Verificación: ¿Estamos construyendo el producto correctamente? e su especificacin.

Validación: ¿Estamos construyendo el producto correcto? Comprueba que el software cumple las expectativas que el cliente espera Importante: Nunca se va a poder demostrar que el software está completamente libre de defectos

las pruebas que pueden utilizarse son muy diversas. Aconsejo centrarnos en pruebas de caja blanca y de caja negra.

Las pruebas de la caja blanca se centran en la estructura interna del programa para elegir los casos de prueba. El objetivo de estas pruebas consiste en probar todos los posibles casos de ejecución de la aplicación para comprobar que los datos se comportan de manera correcta internamente.

Decir que se han hecho las pruebas de caja blanca.

Las pruebas de caja negra son aquellas que se centran en las salidas y entradas de los módulos, sin atender a su comportamiento interno (comprobando mediante las pruebas de caja blanca). Las pruebas de caja negra garantizan la interconectividad entre los diferentes módulos de la aplicación, así como su correcto funcionamiento final.

Poner algunos casos de prueba de caja negra.

Capítulo 4

Conclusiones y trabajo futuro

Decir lo que se ha conseguido realizar comentando sus puntos fuertes y débiles. Decir si se han alcanzado los objetivos específicos y el general propuesto y en qué grado.

Indicar las asignaturas del grado más relacionadas con la ejecución del TFG y cómo el TFG ha ayudado a afianzar los conocimientos adquiridos en el Grado.

Valoración personal si se quiere.

Avanzar algunas líneas de trabajo futuro para solucionar las debilidades detectadas o para conseguir nuevas funcionalidades interesantes.

Capítulo 5

Bibliografía

Poner las citas bibliográficas, direcciones de internet, etc.

Capítulo 6

Anexo

Al final de la memoria hay que añadir un anexo de una página o dos, explicando como se usa el software a modo de manual de usuario. Es decir, como se llaman los comandos, qué parámetros hay que darle, como se llaman los ficheros de datos de entrada, etc.

