ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

EPA Explorer

Software de preparación, procesado y análisis de datos de la EPA  
(Encuesta de Población Activa)

Curso 2016-2017

José Saúco Delgado

Cádiz, 30 de mayo de 2017

ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

EPA Explorer

Software de preparación, procesado y análisis de datos de la EPA  
(Encuesta de Población Activa)

DEPARTAMENTO: Ingeniería Informática.

DIRECTORES DEL PROYECTO: Dña. Elisa Guerrero Vázquez y Don Andrés Yáñez Escolano.

AUTORA DEL PROYECTO: Don José Saúco Delgado

Cádiz, 30 de mayo de 2017

José Saúco Delgado

Este documento se halla bajo la licencia FDL[[1]](#footnote-1).

Según estipula la licencia, se muestra aquí el aviso de copyright. Se ha usado la versión inglesa de la licencia, al ser la única que se ha reconocido oficialmente por la FSF[[2]](#footnote-2).

Copyright© 2014 José Saúco Delgado

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the FSF; With no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled “GNU Free Documentation License”.

*A mi madre, al final lo hice.*

*"The programmers of tomorrow are the wizards of the future.  
You're going to look like you have magic powers compared to everybody else."*

**Gabe Newell**, fundador y presidente de Valve.

**Agradecimientos**

* A mi novia *Deborah* por estar ahí para mí en todo momento.
* A todos y cada uno de los miembros de mi familia, en especial a mi madre. No sé cuántas veces me habrá preguntado que “como voy con el proyecto”.
* A la sección de pescadería de Mercadona, por facilitar un encuentro fortuito con Andrés. Si no llega a ser por eso no estamos hoy aquí.
* A *Elisa Guerrero Vázquez* y *Andrés Yáñez Escolano,* por ayudarme y motivarme a terminar este proyecto.

Muchas gracias a todos.

ÍNDICE GENERAL

[Capítulo 1: Introducción 5](#_Toc483940047)

[1.1. Objetivos 7](#_Toc483940048)

[1.2. Tecnologías empleadas 8](#_Toc483940049)

[1.3. Definiciones, acrónimos y abreviaturas 9](#_Toc483940050)

[1.4. Contenido 11](#_Toc483940051)

[Capítulo 2: Desarrollo del calendario 13](#_Toc483940052)

[2.1. Iteraciones 13](#_Toc483940053)

[2.2. Diagrama de Gantt 15](#_Toc483940054)

[2.3. Esfuerzos 16](#_Toc483940055)

[2.4. Presupuesto 18](#_Toc483940056)

[Capítulo 3: Descripción general del proyecto 21](#_Toc483940057)

[3.1. Perspectiva del producto 21](#_Toc483940058)

[3.2. Características de los usuarios 21](#_Toc483940059)

[3.3. Funciones 22](#_Toc483940060)

[3.4. Restricciones generales 24](#_Toc483940061)

[3.5. Suposiciones y dependencias 24](#_Toc483940062)

[3.6. Requisitos para futuras versiones 26](#_Toc483940063)

[Capítulo 4 : Desarrollo del proyecto 27](#_Toc483940064)

[4.1. Metodología de desarrollo 27](#_Toc483940065)

[4.2. Especificación de requisitos del sistema 29](#_Toc483940066)

[4.2.1. Requisitos interfaces externas 29](#_Toc483940067)

[4.2.2. Disposiciones legales y estándares 32](#_Toc483940068)

[4.2.3. Requisitos funcionales 32](#_Toc483940069)

[4.2.4. Atributos 33](#_Toc483940070)

[4.3. Análisis del sistema 33](#_Toc483940071)

[4.3.1. Modelo de casos de uso 35](#_Toc483940072)

[4.3.2. Modelo conceptual de datos 50](#_Toc483940073)

[4.3.3. Modelo de comportamiento 53](#_Toc483940074)

[4.4. Diseño del sistema 68](#_Toc483940075)

[4.4.1. Arquitectura de sistema software 68](#_Toc483940076)

[4.4.2. Diseño de base de datos 70](#_Toc483940077)

[4.4.3. Diseño detallado del sistema 75](#_Toc483940078)

[4.4.4. Interfaz con el usuario 82](#_Toc483940079)

[4.5. Codificación 83](#_Toc483940080)

[4.5.1. Implementación 83](#_Toc483940081)

[4.5.2. Otros programas 86](#_Toc483940082)

[4.6. Pruebas y validación 86](#_Toc483940083)

[4.6.1. Entorno de pruebas 87](#_Toc483940084)

[Capítulo 5: Conclusiones 95](#_Toc483940085)

[5.1 Calendario 95](#_Toc483940086)

[5.2 Interfaz 96](#_Toc483940087)

[5.3 Futuras ampliaciones 96](#_Toc483940088)

[5.4 Valoraciones personales 96](#_Toc483940089)

[Bibliografía y Referencias electrónicas 99](#_Toc483940090)

[Referencias 99](#_Toc483940091)

[Otras Webs de consulta 100](#_Toc483940092)

[Bibliografía 101](#_Toc483940093)

[Anexo A: Manual de usuario 103](#_Toc483940094)

[Anexo B: Proceso licitatorio 125](#_Toc483940095)

[B.1 La Licitación Pública 125](#_Toc483940096)

[B.1.1 Oposición o libre concurrencia 126](#_Toc483940097)

[B.1.2 Publicidad 126](#_Toc483940098)

[B.1.3 Igualdad 126](#_Toc483940099)

[B.2 Etapas del procedimiento licitatorio 126](#_Toc483940100)

[B.2.1 Elaboración del pliego de bases y condiciones 126](#_Toc483940101)

[B.2.2 El llamado a licitación 127](#_Toc483940102)

[B.2.3 Oferta y presentación del pliego 127](#_Toc483940103)

[Apéndice A: GNU Free Documentation License v 1.3 129](#_Toc483940104)

[Apéndice B: Documentación Estándar de Licitación Pública 141](#_Toc483940105)

ÍNDICE DE FIGURAS

[Figura 1: Base de datos jerárquica 5](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939972)

[Figura 2: Menú principal Natural 6](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939973)

[Figura 3: Mantenimiento de prendas Natural 7](#_Toc483939974)

[Figura 4: Diagrama de modelo incremental 13](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939975)

[Figura 5: Tiempo estimado de las iteraciones 16](#_Toc483939976)

[Figura 6: Diagrama de Gantt 17](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939977)

[Figura 7: Tabla días estimados y reales 18](#_Toc483939978)

[Figura 8: Comparación de los costes del proyecto 19](#_Toc483939979)

[Figura 9: Logueo único 25](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939980)

[Figura 10: Mis aplicaciones 25](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939981)

[Figura 11: Conexión con la LDAP 26](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939982)

[Figura 12: Prácticas de la metodología XP 28](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939983)

[Figura 13: Pantalla principal de Vestium 30](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939984)

[Figura 14: Pantalla de lista de mantenimiento de centros Vestium 31](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939985)

[Figura 15: Ver / editar centro Vestium 31](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939986)

[Figura 16: Diagrama de casos de uso 36](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939987)

[Figura 17: CU de Copia de ejercicio 37](#_Toc483939988)

[Figura 18: CU de listar grupos 38](#_Toc483939989)

[Figura 19: CU de listar oficios 40](#_Toc483939990)

[Figura 20: CU de generar necesidades 43](#_Toc483939991)

[Figura 21: CU de consultar necesidades 44](#_Toc483939992)

[Figura 22: CU de alta de actuación 45](#_Toc483939993)

[Figura 23: CU de rehacer alta actuación 46](#_Toc483939994)

[Figura 24: CU de generar tallaje 47](#_Toc483939995)

[Figura 25: CU de consultar tallaje 48](#_Toc483939996)

[Figura 26: CU de generar precio de prendas 49](#_Toc483939997)

[Figura 27: Diagrama de clases conceptual 50](file:///D:\workarea\epa_explorer\mem\memoria.docx#_Toc483939998)

[Figura 28: Clase Munic y Centro 51](#_Toc483939999)

[Figura 29: Clase Oficio y Prenda 51](#_Toc483940000)

[Figura 30: Clase Grupo y OficiosCentro 52](#_Toc483940001)

[Figura 31: Clase PrendasOficio y Estados 52](#_Toc483940002)

[Figura 32: Clase Prov y País 53](#_Toc483940003)

[Figura 33: DSS del caso de uso de gestión de perfil 54](#_Toc483940004)

[Figura 34: DSS del caso de uso de copia de ejercicio 55](#_Toc483940005)

[Figura 35: DSS del caso de uso de listar grupos 56](#_Toc483940006)

[Figura 36: DSS del caso de uso de listar oficios 59](#_Toc483940007)

[Figura 37: DSS del caso de uso de generar necesidades 61](#_Toc483940008)

[Figura 38: DSS del caso de uso de consultar necesidades 62](#_Toc483940009)

[Figura 39: DSS del caso de uso de alta de actuación 63](#_Toc483940010)

[Figura 40: DSS del caso de uso de rehacer alta actuación 64](#_Toc483940011)

[Figura 41: DSS del caso de uso de generar tallaje 65](#_Toc483940012)

[Figura 42: DSS del caso de uso de generar precios prendas 67](#_Toc483940013)

[Figura 43: Arquitectura por capas 68](#_Toc483940014)

[Figura 44: Modelo vista controlador 70](#_Toc483940015)

[Figura 45: Definición lógica de la tabla de municipios 71](#_Toc483940016)

[Figura 46: Definición lógica de la tabla Centro 71](#_Toc483940017)

[Figura 47: Definición lógica de la tabla Oficio 72](#_Toc483940018)

[Figura 48: Definición lógica de la tabla Prenda 72](#_Toc483940019)

[Figura 49: Definición lógica de la tabla Grupo 73](#_Toc483940020)

[Figura 50: Definición lógica de la tabla OficiosCentro 73](#_Toc483940021)

[Figura 51: Definición lógica de la tabla PrendasOficio 74](#_Toc483940022)

[Figura 52: Definición lógica de la tabla Estados 74](#_Toc483940023)

[Figura 53: Operaciones de la clase TPais 75](#_Toc483940024)

[Figura 54: Operaciones de la clase TProv 75](#_Toc483940025)

[Figura 55: Operaciones de la clase TMunic 76](#_Toc483940026)

[Figura 56: Operaciones de la clase TPrenda 76](#_Toc483940027)

[Figura 57: Operaciones de la clase TCentro 77](#_Toc483940028)

[Figura 58: Operaciones de la clase TOficio 78](#_Toc483940029)

[Figura 59: Operaciones de la clase TOficioCentro 80](#_Toc483940030)

[Figura 60: Operaciones de la clase TPrendaOficio 81](#_Toc483940031)

[Figura 61: Operaciones de la clase TGrupo 81](#_Toc483940032)

[Figura 62: Operaciones de la clase TEstados 82](#_Toc483940033)

[Figura 63: Pruebas caso de uso listar grupo 88](#_Toc483940034)

[Figura 64: Pruebas caso de uso listar prenda 89](#_Toc483940035)

[Figura 65: Pruebas caso de uso listar oficio 90](#_Toc483940036)

[Figura 66: Pruebas de caso de uso listar centros 91](#_Toc483940037)

[Figura 67: Pruebas de caso de uso generar necesidades 91](#_Toc483940038)

[Figura 68: Pruebas del caso de uso consulta de necesidades 92](#_Toc483940039)

[Figura 69: Pruebas caso de uso alta de actuación 92](#_Toc483940040)

[Figura 70: Pruebas de caso de uso deshacer alta actuación 92](#_Toc483940041)

[Figura 71: Pruebas de caso de uso generar tallaje 93](#_Toc483940042)

[Figura 72: Pruebas de caso de uso consulta tallaje 93](#_Toc483940043)

[Figura 73: Pruebas de caso de uso generar precio prendas 93](#_Toc483940044)

[Figura 74: Otras pruebas realizadas 94](#_Toc483940045)

[Figura 75: Esquema proceso licitación pública 125](#_Toc483940046)

Capítulo 1: Introducción

Vestium, gestión de vestuario de la Diputación de Cádiz, es un sistema de gestión donde se va a realizar el control del conjunto de prendas, complementos y accesorios para los diferentes oficios desempeñados en los centros de la Diputación.

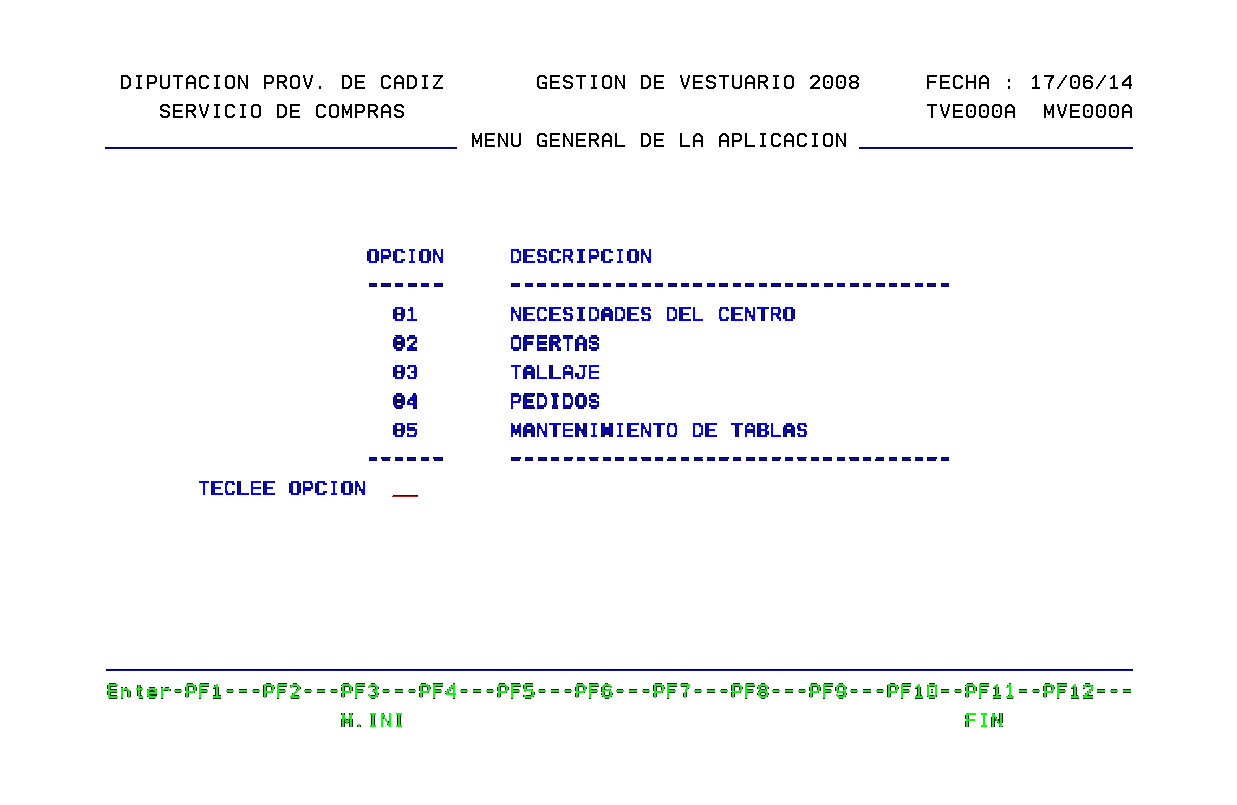
La gestión del vestuario consiste principalmente en, una vez realizadas las negociaciones entre los Sindicatos y la Diputación y definido el catálogo de prendas para el año en cuestión, se lleva a cabo un mantenimiento de los diferentes objetos que componen dicho catálogo. Tras la realización de este mantenimiento se generan las necesidades para los diferentes oficios pertenecientes a los centros que forman parte de la Diputación. Las necesidades se ponen en conocimiento de la Central de Contratación y comienza el proceso licitatorio. Una vez gestionado el contrato, la empresa adjudicataria realiza el tallaje y se formaliza el contrato y el pedido. La empresa informa a la Central de Contratación sobre el resultado del tallaje y el usuario realiza el mantenimiento de este en la gestión de vestuario.

Actualmente, en la Central de Contratación, el vestuario se gestiona mediante una aplicación desarrollada en lenguaje Natural y base de datos ADABAS. La base de datos ADABAS es de tipo jerárquico (figura 1).

Figura 1: Base de datos jerárquica

Este tipo de modelo se caracteriza por la redundancia de datos que se almacenan, por lo que no optimiza el espacio. Las operaciones de inserción, extracción y eliminación son complejas por la forma de árbol que toma el modelo, además, existe la necesidad de tener conocimiento de las distintas unidades y relaciones que tienen entre sí.

En la figura 2 se muestra el menú principal de la aplicación en Natural donde se puede observar que es poco intuitiva y dinámica, además de ser de difícil manejo ya que se debe tener conocimiento de los elementos con los que se quiere trabajar.

Por ejemplo, para hacer una modificación de una prenda, se debe conocer el código de la misma previamente para poder realizar los cambios, como se puede apreciar en la figura 3.

En los últimos años en la Central de Contratación han aparecido nuevas necesidades que, por las características de la aplicación, desarrollada en Natural, no se han podido abastecer. La motivación principal de este proyecto es aportar, a este departamento de la Diputación, una herramienta útil, simple y sencilla para gestionar el vestuario, que cubra las limitaciones de la aplicación actual.

Figura 2: Menú principal Natural

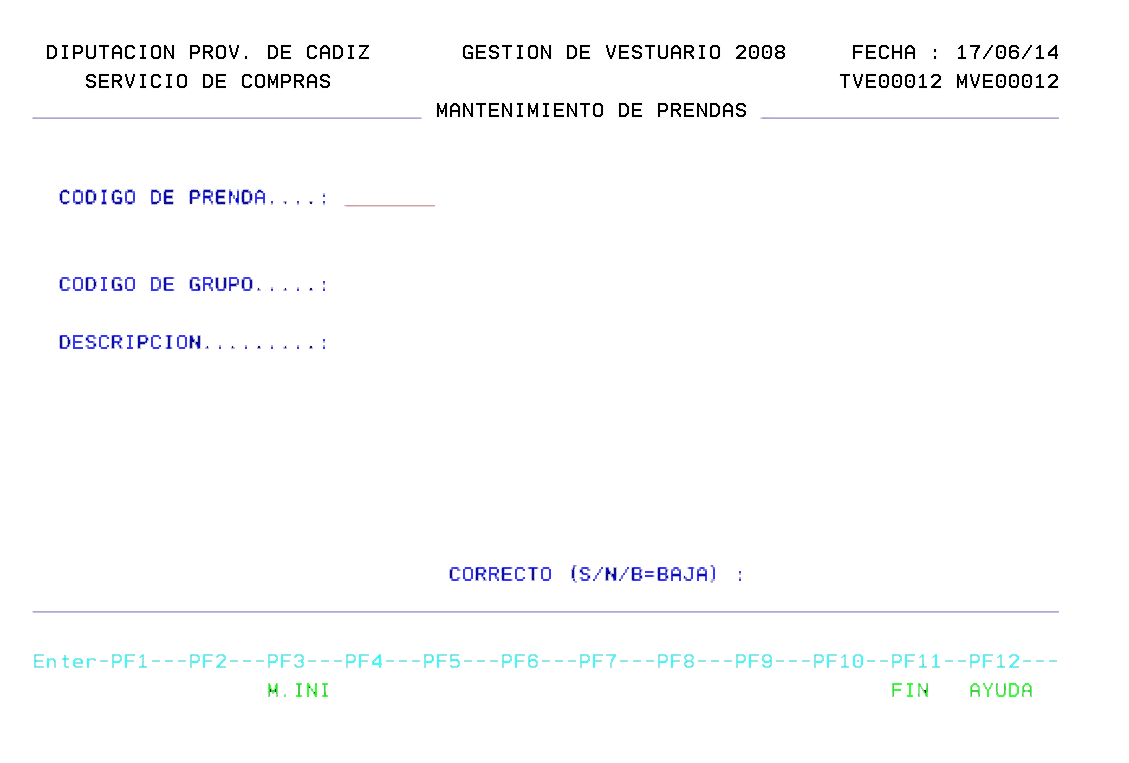


Figura 3: Mantenimiento de prendas Natural

Objetivos

* Desarrollar una nueva aplicación web con una interfaz más dinámica e interactiva, fácil e intuitiva .
* Utilizar una base de datos de tipo relacional ya que este tipo de modelo permite establecer conexiones entre datos almacenados en tablas y mantener la integridad referencial de forma que al eliminar un registro se eliminarán todos los registros dependientes que estén relacionados.
* Incluir, la gestión del vestuario, en el ámbito de la administración electrónica para la agilización de todas las partes del proceso que lo forman.
* Integrar, la nueva aplicación, mediante servicios webs y otros medios, con los aplicativos que se indican a continuación:
* **Gestión de la configuración**, capa de abstracción que conforma los menús, gestiona los perfiles y permisos de los diferentes usuarios a través de servicios web para la aplicación.
* **Josso** [2], software libre que valida con la **LDAP** [3] (Lightweight Directory Access Protocol, protocolo utilizado para acceder a información almacenada en un directorio de información) los usuarios, comprobando que formen parte de la organización y realiza un login único para todas las aplicaciones.
* **Contrat@**, que contempla las contrataciones por procedimientos abiertos, negociados y contratos menores. Desde Vestium se realizarán inserciones en la base de datos de Contrat@ creándose así una actuación en el aplicativo para las adjudicaciones de los contratos y desarrollo de los mismos.

Tecnologías empleadas

Este proyecto se ha desarrollado en **Java** ya que este tipo de lenguaje permite desarrollar aplicaciones de alto rendimiento para un amplio rango de plataformas informáticas [1]. El software que se realiza para la Diputación es libre y de código abierto. La versión utilizada es Java Standard Edition 6, Java SE 6Delopment Kit 1.6.0, JDK 1.6.0.

Como frameworks se han aplicado **Struts** que implementa el patrón modelo vista controlador y es utilizado muy comúnmente en aplicaciones tipo web. **Spring**, la inyección de dependencias string consigue controlar la creación de los objetos.

**Eclipse IDE for Java EE Developers** versión Juno 2013, es la plataforma elegida para desarrollar el entorno.

Para base de datos, se ha utilizado **Oracle Database**, sistema de gestión de bases de datos de tipo objeto-relacional. Para mapear los atributos y el modelo de objetos de la aplicación se utiliza la herramienta para la plataforma Java **Hibernate**. **Oracle SQL Developer** versión 3.2.20.09 es la herramienta gráfica seleccionada para la editar, y realizar registros en la base de datos Oracle.

**Axis2** necesario para que la aplicación de Gestión de la Configuración mediante, un logueo único para entrar en todas las aplicaciones, cargue el fichero **XML** con el menú de forma automática.

Con respecto a la interfaz del usuario, se ha desarrollado mediante **CSS**. Como editor de imágenes se ha utilizado **Gimp 2.8** de GNU y licencia GPL versión 2.8.10.

**DIA** versión 0.97.2, aplicación gráfica con licencia GPL para la creación de diagramas.

Para la estimación de tiempo y esfuerzos además de la realización del diagrama se ha utilizado **Gantt Project**, aplicación de código abierto, versión 2.0.10.

La visión de documentos PDF se ha realizado mediante **Adobe Reader X** versión 10.1.10.

Definiciones, acrónimos y abreviaturas

A continuación se describen un conjunto de definiciones, acrónimos y abreviaturas que se utilizarán a lo largo de este proyecto.

* **Prenda**. Cada una de las partes de una vestimenta que son necesarias para formar un uniforme. Por ejemplo, camisa blanca, pantalón negro, calcetines grises y zapatos ejecutivo. Cada uno de ellos es una prenda.
* **Grupo**. Una serie de prendas que reúnen unas características comunes es lo que se llamará a partir de ahora un grupo. Por ejemplo, el grupo calzado lo forman las prendas zapato ejecutivo, zapato seguridad, zanco blanco, etc.
* **Oficio**. Un oficio es una ocupación a la que se le va a suministrar un vestuario específico. Por ejemplo, entre los oficios se incluyen: auxiliar de clínica, conductor/a, jardinero/a, cocinero/a, profesor/a, etc.
* **Centro**. Un centro es un lugar físico perteneciente a la Diputación de Cádiz donde se van a desarrollar una serie de oficios. Independientemente del centro donde se desempeñe un oficio, este va a tener el mismo conjunto de prendas.
* **Catálogo**. El catálogo lo forma el conjunto de todas las prendas que se establecen en la negociación previa que se realiza entre la Diputación y los Sindicatos. En este catálogo se especifica la cantidad y periodicidad con la que se suministran las prendas acordadas.
* **Necesidades**. Todos los centros no van a tener los mismos oficios, ni tampoco todos los oficios utilizan las misma prendas, por lo que se debe conocer e indicar para cada oficio qué prendas son las que se van a utilizar, teniendo en cuenta para ello la cantidad de personas que desempeñan el oficio, en cada centro, y la periodicidad y cantidad con la que se suministra cada prenda anualmente. Por ejemplo, el centro “Parque Móvil” sólo tiene el oficio de conductor, desempeñado por 3 trabajadores. Dicho oficio tiene como prendas estipuladas camisa blanca (2 unidades anuales), pantalón negro (2 unidades anuales), calcetín gris (2 unidades trimestrales) y zapato de ejecutivo (1 unidad anual). Las necesidades de ese centro consistirán en:
* 3 trabajadores X 2 unidades de camisa blancas al año = 6 camisas blancas.
* Del mismo modo serán 6 pantalones negros.
* Para los calcetines se tienen 3 trabajadores X 2 unidades de calcetines X 4 trimestres que tiene el año = 24 Calcetines grises.
* Consistiría en 3 zapatos de ejecutivo.

La necesidades de ese centro por lo tanto consistirá en, 6 camisas blancas, 6 pantalones negros, 24 calcetines grises y 3 zapatos ejecutivos anuales.

* **Tallaje**. El tallaje consiste en comprobar las necesidades reales in situ, verificando el número de prendas y empleados.
* **Licitadores**. Un licitador es una persona o una entidad que ofrece un precio en una subasta o en un concurso.
* **XP**. eXtreme Programming
* **PFC**. Proyecto Fin de Carrera.
* **BD**. Base de datos

Contenido

Este documento está dividido en diferentes capítulos donde se han ido describiendo con detalle las distintas partes de su desarrollo.

En el capítulo 2 se explican las iteraciones en las que se ha dividido el PFC, así como el diagrama temporal de las fases que han ido desarrollando para la realización del mismo. En esta misma sección se podrán comparar dos presupuestos, correspondientes a dos circunstancias diferentes.

En el siguiente capítulo, el capítulo 3, se realiza una descripción general del proyecto donde se pueden encontrar las funcionalidades que lo caracteriza, los usuarios a los que está destinado y las funciones que están asignadas a cada uno de ellos.

El capítulo 4 se divide en seis grandes bloques donde se explican las partes que forman el desarrollo del proyecto. Esto bloques son:

* Metodología de desarrollo.
* Especificación de requisitos.
* Análisis del sistema.
* Diseño del sistema.
* Codificación.
* Pruebas y validación.

En el último capítulo, el de las conclusiones, se hará un balance de lo que ha supuesto el trabajo y de las decisiones que se han ido tomando durante la realización del mismo, así como una valoración personal.

Tras el capítulo de las conclusiones se indican las referencias electrónicas, la bibliografía que se ha consultado y dos anexos. El anexo A corresponde al manual del usuario y en el anexo B se explica con todo detalle en qué consiste un proceso licitatorio.

Al final del presente documento y para concluir con el mismo se incluyen las documentaciones sobre la licencia FLD y la legislación vigente sobre la licitación pública en España.

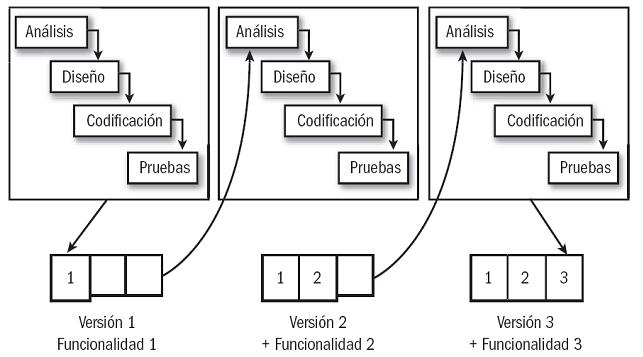
Capítulo 2: Desarrollo del calendario



Iteraciones

Para el desarrollo de este proyecto se ha utilizado el modelo de ciclo de vida incremental, en la figura 4 se puede ver un esquema sobre este tipo de modelo [4]. Este modelo se desarrolla para satisfacer unos requisitos previamente especificados. En las posteriores versiones que se vayan realizando, se incrementa el sistema, añadiendo nuevas funcionalidades con la finalidad de satisfacer más requisitos cada vez [5].

Figura 4: Diagrama de modelo incremental

Gracias a este sistema se aumentan gradualmente las necesidades del software ya que los ciclos se van repitiendo hasta la obtención de un producto completo.

A continuación, se indican cuáles han sido las iteraciones necesarias para el desarrollo de este proyecto.

**Primera iteración: Especificación de requisitos**

La primera toma de contacto se lleva a cabo cuando se efectúa la primera entrevista con el jefe de proyecto, Joaquín Ruiz. Tras esta entrevista se realiza el análisis de requisitos para saber cuáles son las necesidades y se desarrolla el estudio para concretar las diferentes funcionalidades. Se bajaran varias alternativas para seleccionar las tecnologías necesarias para el desarrollo del sistema. Tras ello se realiza un esquema básico, un prototipo que ayuda a tener una primera toma de contacto con el proyecto y que permite establecer cuáles son las pantallas principales de las que el usuario ha de disponer.

**Segunda Iteración: Diseño de la base de datos**

Tas concretar las funcionalidades se diseñan las tablas provisionales de la base de datos ya que, a medida que avanza el proyecto, irán sufriendo pequeños cambios que se adapten a las nuevas necesidades.

**Tercera Iteración: Menú y autenticación de usuarios**

Una vez diseñada la base de datos, se realiza la configuración de Vestium en la aplicación Gestión de la Configuración para la formación del menú automático y creación de los perfiles de los usuarios para ocultar los puntos de menú correspondientes. Se integra la herramienta Josso para la validación de los usuarios con la LDAP, necesario para realizar el login único para todas las aplicaciones de las que disponga el usuario. En esa iteración se realizan pruebas de caja negra para comprobar la visibilidad del menú según si el usuario es administrador o técnico.

**Cuarta Iteración: Gestión para el administrador**

Se realiza el mantenimiento de todas las tablas necesarias en el punto de menú de Mantenimiento. En esta iteración se realizan de pruebas de caja blanca para cada consulta, inserción y eliminación lógica de cada tabla y comprobar cómo afecta a la eliminación lógica de un registro de una tabla que es clave foránea de otra.

**Quinta Iteración: Copia de ejercicio y gestión de necesidades**

Una vez terminado el mantenimiento de las prendas se incluye el punto de menú de copia de ejercicio, requisito añadido por el usuario, donde se realiza la copia de todas las tablas de mantenimiento para el año que se indique. Tras ello se realiza el mantenimiento de las necesidades, punto necesario para realizar el alta de la actuación de la siguiente iteración. Se realizan pruebas de caja negra para la comprobación del correcto funcionamiento.

**Sexta Iteración: Alta actuación en Contrat@**

Tras realizar el mantenimiento de las necesidades se desarrolla el módulo en el que se elabora el alta de una nueva actuación en el aplicativo de contratación, Contrat@. Para la comprobación de esta iteración las pruebas seleccionadas son de caja negra, realizando altas de actuaciones.

**Séptima Iteración: Gestión de tallaje**

Una vez realizada el alta de la actuación, y completado el proceso licitatorio, se lleva a cabo el tallaje. El módulo de tallaje consiste en realizar un mantenimiento de las necesidades, que pasan a llamarse tallaje. Esta información actualizada es facilitada por el licitador y consiste en realizar una comprobación in situ de cuáles son las necesidades reales de cada centro. Se realizan las pruebas de caja negra de edición de un tallaje creado y la creación de un nuevo tallaje.

Diagrama de Gantt

Para la realización de este diagrama de Gantt se ha empleado la herramienta Gantt Project. Dicha aplicación es de código abierto y está desarrollada en Java y disponible en la dirección de enlace [6].

La finalidad de este tipo de herramientas es la utilización de gráficos para la representación de diferentes fases, actividades y tareas que se programan como parte de un proyecto y mostrarlas en una línea temporal.

Una vez introducido generalmente el diagrama de Gantt en la figura 5 se puede comprobar las iteraciones y su estimación en el tiempo en la figura 6.

Figura 5: Tiempo estimado de las iteraciones

Esfuerzos

En la figura 7 se representa los días invertidos estimados y los reales por iteración. En primer lugar el proyecto se desarrollaría en totalidad en horario laboral, pero para poder cumplir con los plazos se decidió añadir a los días de trabajo los no laborables por lo que la fecha del fin del mismo se respeta pero el tiempo invertido aumenta.

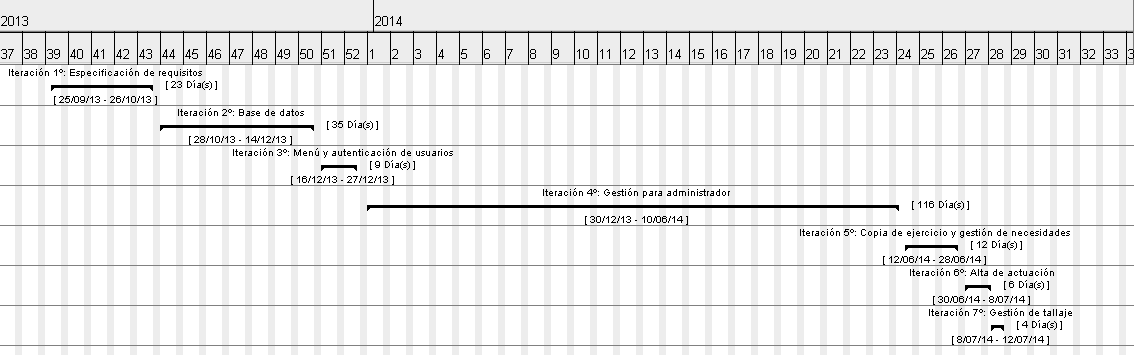


Figura 6: Diagrama de Gantt

|  |  |  |
| --- | --- | --- |
| Tareas Realizadas | Estimados | Reales |
|  |  |  |
| Iteración 1º: Especificación de requisitos | 23 | 27 (+4) |
| Iteración 2º: Base de datos | 35 | 37 (+2) |
| Iteración 3º: Menú y autenticación de usuarios | 9 | 11 (+2) |
| Iteración 4º: Gestión para administrador | 116 | 140 (+24) |
| Iteración 5º: Copia de ejercicio y gestión de necesidades | 12 | 14 (+2) |
| Iteración 6º: Alta de actuación | 6 | 8 (+2) |
| Iteración 7º: Gestión de tallaje | 4 | 4 |
|  |  |  |
| Totales | 205 días | 241 días |

Figura 7: Tabla días estimados y reales

Presupuesto

Para la realización del presupuesto se ha consultado la herramienta Plan de carrera de infojobs [7].

La herramienta hace distinción entre hombres y mujeres y provincia por lo que se ha hecho la estimación utilizando para ello el equipo que ha trabajado en el proyecto. Una analista programadora y una programadora junior en la provincia de Cádiz. Se ha estipulado que cada mes tiene 30 días y se estima que los gastos indirectos son un 10% de los gastos totales del proyecto.

Para la analista programadora el salario medio es de 23.400€ brutos anuales lo que hace un total de 1.950€ brutos mensuales y para la programadora junior el salario medio es de 15.600€ brutos anuales, lo que hace un total de 1.300€ brutos mensuales, ambos en una jornada laboral de 8 horas.

La estimación se realizó en 205 días, de los cuales 27 días se realizaron tareas de análisis y los restantes 178 días de diseño, codificación y pruebas.

El presupuesto total del proyecto sería: destinado a tareas de **análisis** 1.820€, destinado a tareas de **diseño, codificación y pruebas** 7.722€ y destinado a **gastos indirectos** 954€. Total del presupuesto estimado medio 10.496€.

El proyecto se ha realizado en EPICSA, SA. Empresa Provincial de Información de Cádiz, durante el periodo de prácticas por lo que los salarios varían, y por lo tanto el coste real es el que se desglosa a continuación. Para una mejor comparativa se puede observar la figura 8.

Tanto las tareas de análisis como de diseño, codificación y pruebas han sido realizadas por una programadora en prácticas que cobra 440€ brutos mensuales en una jornada de 5 horas. Lo que haría un total de 704€ brutos mensuales si la jornada fuera de 8 horas. Por lo tanto, destinado a tareas de **análisis** 634€, destinadas a tareas de **diseño, codificación y pruebas** 4.182€ y destinado a **gastos indirectos** 481€. Total del presupuesto estimado medio 5.298€ .

|  |  |  |
| --- | --- | --- |
| Tareas | Circunstancias normales | Circunstancias reales |
|  |  |  |
| Análisis | 1.820 € | 634 € |
| Codificación, diseño y pruebas | 7.722 € | 4.182 € |
| Gastos indirectos (10% total) | 954 € | 481 € |
| Total | 10.496 € | 5.298 € |

Figura 8: Comparación de los costes del proyecto

Capítulo 3: Descripción general del proyecto

Perspectiva del producto

Vestium, gestión de vestuario de la Diputación de Cádiz, es un producto que se encuentra integrado en el espacio web de la Diputación de Cádiz, concretamente en el recurso desarrollado que se conoce como Intranet. La Intranet es el sitio web [8] donde se encuentran los accesos corporativos y las aplicaciones asignadas al usuario logueado. Por este motivo se ha debido respetar la identidad visual evitando quitarle uniformidad al producto.

Se ha utilizado el diseño definido por EPICSA y ha sido adaptado para su correcta visualización en diferentes navegadores ya que en la Diputación de Cádiz es libre el uso de cualquier tipo de navegador.

La interfaz debe ser intuitiva, fácil de asimilar y aprender. Sus accesos deben ser rápidos debido a que uno de los objetivos de este proyecto es disminuir el esfuerzo y el tiempo que se dedicaba a la gestión del vestuario. Además, debe contar con una serie de botones descriptivos que identifiquen claramente las actividades que llevan a cabo cada uno. Debe contar también con un menú donde se contemplan todas las funcionalidades del sistema consiguiendo, de esta forma, un acceso directo sin necesidad de realizar búsquedas por el sistema.

Características de los usuarios

Esta aplicación está destinada a usuarios pertenecientes al departamento de la Central de Contratación de la Diputación de Cádiz. El proyecto debe ser fácil e intuitivo en el manejo porque no requiere un nivel de conocimientos altos de la aplicación.

La determinación del tipo de usuario que accede a la aplicación se define en la Gestión de la Configuración. Estos pueden ser administradores o técnicos.

El administrador es el usuario que va a tener acceso a todas las funciones de las que dispone la aplicación. El técnico, es un usuario más básico y solo podrá realizar tareas de gestión de necesidades, gestión de tallaje y consulta en el catálogo de vestuario.

Por otra parte, el mantenimiento informático de la aplicación debe de correr a cargo de una persona con nociones con el manejo de las bases de datos relacionales, además de conocer Java y Css por lo general.

Funciones

A continuación se indican las funcionalidades clasificadas principalmente según el usuario y punto de menú.

Todas las funciones se realizarán sobre el ejercicio cargado, pudiendo cambiar este en todo momento.

Funcionalidades de un **técnico**:

* **Mantenimiento**
* Catálogo de Vestuario:
* El usuario podrá realizar una consulta del catálogo de vestuario.
* **Gestión de necesidades**
* Consulta de necesidades:
* Podrá realizar consultas de las necesidades.
* Generar necesidades:
* El usuario podrá realizar la generación de las necesidades a partir de la fecha del sistema.
* Realizar el alta de la actuación en Contrat@ para comenzar con el proceso licitatorio.
* **Gestión de tallaje:**
* Consulta de tallaje:
* Realizar la consulta del tallaje de cualquier año registrado.
* Generar tallaje:
* Realizar el tallaje, indicando para ello las cantidades reales van a ser necesarias finalmente.
* Generar precios prendas:
* Introducir para cada prenda el precio establecido por la empresa adjudicataria.

Además de las funciones de un técnico el **administrador** podrá:

* **Mantenimiento**
* Rehacer alta de actuación:
* El usuario podrá volver a activar la posibilidad de realizar el alta de la actuación en caso de equivocación por parte del técnico.
* Copia de ejercicio:
* El usuario podrá realizar una copia de un ejercicio anterior para la preparación de el nuevo ejercicio.
* Centros:
* Consultar el listado de los centros.
* El usuario podrá realizar el alta de un nuevo centro.
* El usuario podrá realizar modificaciones sobre un centro ya creado.
* El usuario podrá asociar al centro los diferentes oficios que se desempeñan en él, asignando para ello la cantidad de personas que lo desarrollan.
* Realizar modificaciones sobre las asociaciones
* Oficios:
* Consultar el listado de los oficios existentes.
* Podrá realizar el alta de un nuevo oficio.
* El usuario podrá realizar modificaciones sobre un oficio ya creado.
* El usuario realizará la asociación de las prendas que son necesarias para formar el uniforme de este, indicando el periodo y la cantidad por trabajador que se estipule para su suministro.
* Prendas:
* El usuario podrá consultar la lista que incluye todas las prendas del año.
* Realizar el alta de una prenda.
* Asociar a la prenda el grupo al que pertenece.
* Modificar una prenda existente.
* Grupos:
* Consultar el listado de todos los grupos existentes.
* Realizar el alta de un nuevo grupo.
* Realizar modificaciones en un grupo ya definido.

En la gestión de cada una de las tareas se incluyen los siguientes subapartados:

* Poder realizar una exportación a Excel del listado de cada uno de los elementos anteriores.
* Dar de baja elementos y asociaciones a partir de una fecha dada o con la fecha actual del sistema.

Restricciones generales

Debe ser una aplicación escalable para que, al añadir nuevas funcionalidades, esto no impida volver a realizar la acción en el futuro. El proyecto debe ser abierto a cualquier tipo de modificación.

Suposiciones y dependencias

El usuario debe estar dado de alta previamente en LDAP, base de datos donde se encuentran registrados todos los trabajadores de la Diputación de Cádiz.

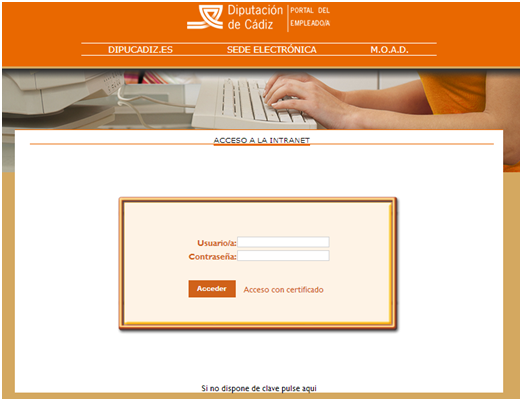
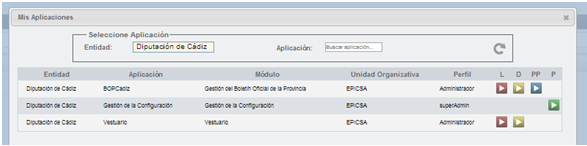
La razón de esto es que uno de los requisitos funcionales de este proyecto es precisamente la integración con Josso, herramienta que permite al usuario realizar un login único y tener acceso a todas sus aplicaciones con el permiso que se le ha asignado para cada una de ellas. En las figuras 9 y 10 se pueden ver las pantallas de login y mis aplicaciones de las que dispone el usuario.

Figura 9: Logueo único

Figura 10: Mis aplicaciones

El usuario accede desde un equipo a la Intranet y se loguea, mediante Josso se comprueba que el usuario se encuentre en la LDAP y tras ello la Gestión de la Configuración gestiona los perfiles y los permisos de las diferentes aplicaciones de manera que al acceder a la aplicación el menú se ajusta a los permisos establecidos. En la figura 11 se puede ver un esquema del proceso.

Requisitos para futuras versiones

En la siguiente versión se realizará, mediante servicios web, el envío de informes a Port@firmas. Para ello es requisito indispensable que los informes sean de formato PDF.

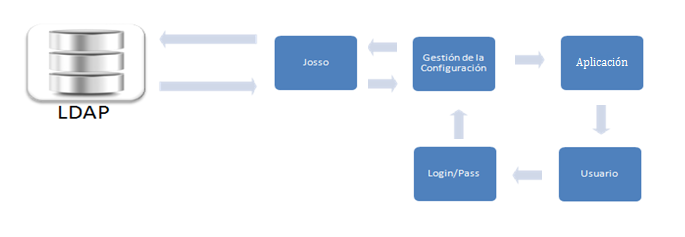
Port@firmas es la herramienta destinada a facilitar a los órganos y unidades administrativas el uso de la firma electrónica basada en certificado reconocido de documentos procedentes de diferentes sistemas de información, con la consiguiente agilización de la actividad administrativa [9].

Figura 11: Conexión con la LDAP

Capítulo 4 : Desarrollo del proyecto

Metodología de desarrollo

La metodología de desarrollo es una estructura cuya finalidad es planear y controlar el desarrollo de un sistema. De entre todos los métodos, se han evaluado las diferentes alternativas disponibles y finalmente se ha optado por utilizar metodología XP, eXtreme Programming [9.1].

Las razones que han impulsado a la selección de esta metodología son:

* A diferencia de otras, donde el cliente es una entidad externa al proyecto con el que se sólo se comunica con el analista, en XP es una parte integral del equipo. Se ocupa de formular los requisitos a nivel conceptual, darles su prioridad, y resolver dudas que pueda tener el resto del equipo. Así, las decisiones de funcionalidad son del cliente, y las decisiones técnicas y el calendario lo establecen los desarrolladores.
* Es una metodología ágil de desarrollo software, aplicable por lo general a proyectos de pequeño y mediano tamaño como es la gestión de vestuario.
* En XP, los detalles se comunican en conversaciones directas entre las partes implicadas, incluyendo al cliente, aclarando confusiones y dudas en el momento. En cada iteración, se entrega una versión funcional del software que el cliente puede probar para ampliar la información disponible en la siguiente iteración.

En la figura 12 se pueden ver las prácticas que se utilizan en la metodología XP.

Estas prácticas se dividen en reglas que un proyecto debe cumplir y en prácticas recomendadas. Éstas prácticas son :

Figura 12: Prácticas de la metodología XP

* **Historias**: No se refiere a casos de uso, si no que describen funcionalidades o propiedades deseadas del sistema explicadas por el usuario y no por el que desarrolla. De esta forma se pueden captar más tipos de requisitos y no solo los funcionales.
* **Integración continua**: Ir probando e integrando continuamente los cambios para corregir fallos introducidos. Si no se realiza así será más difícil cuanto más amplios sean los cambios.
* **Ciclos cortos**: La aplicación se ha realizado en pequeños ciclos manteniendo en cada uno de ellos comunicación con el cliente.
* **Programación basada en pruebas**: Primero se define la prueba que demuestre la funcionalidad. Una vez descrita se comprueba si falla. Se comprueba que la prueba se supere y también supere las anteriores. Esto garantiza la corrección
* **Diseño incremental**: Las metodologías tradicionales siguen un modelo de vida en cascada realizando análisis y diseño en primera instancia y siguiendo un modelo de ingeniería tradicional. Utilizando la analogía con la construcción de una casa, una vez implantados los cimientos, el cambio de la estructura serán órdenes de magnitud mayor. En cambio, con el diseño incremental, el diseño del proyecto no se hace solo una vez si no contiguamente sufre cambios y nuevas necesidades.

Se ha utilizado DIA [10] como herramienta de modelado para la creación de diagramas. Es considerado un sustituto de la aplicación Visio de Microsoft aunque este es un programa multiplataforma bajo licencia GNU . Esta herramienta incluye diagramas de entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, etc.

Especificación de requisitos del sistema

1. Requisitos interfaces externas

Se describen a continuación los requisitos que deben cumplir las interfaces con respecto al software y al usuario.

El gestor de vestuario ofrece a los usuarios la misma interfaz principal. Esta interfaz está compuesta por las estructuras básicas como son la cabecera, menú lateral y pie de página. En la figura 13 se pueden observar cada una de estas partes que se describen a continuación.

En la cabecera aparecen el logotipo de la Diputación de Cádiz, así como el de la aplicación Vestium.

El pié de página está compuesto por la versión de la aplicación, así como un enlace directo donde el usuario podrá visitar el sitio web de la Diputación de Cádiz.

El menú permitirá al usuario navegar por la aplicación pudiendo elegir cualquier funcionalidad disponible para su perfil. Este está dividido en los grupos funcionales:

* Gestión de Necesidades
* Consulta de necesidades
* Generar necesidades
* Rehacer alta actuación
* Gestión de tallaje
* Consulta de tallaje
* Generar tallaje
* Generar precios prendas
* Mantenimiento
* Catálogo de vestuario
* Centros
* Oficios
* Prendas
* Grupos
* Copia ejercicio

La interfaz mostrada al usuario debe ser en cada momento coherente con el estado de la aplicación.

En las figuras 14 y 15 se pueden ver ejemplos de capturas del escritorio de la aplicación. En la primera de ellas se muestra la pantalla de listado de centros que aparece cuando se selecciona del menú principal el mantenimiento de centros. En la segunda se aprecia la pantalla disponible cuando del listado anterior se selecciona uno de los centros con el lápiz para su edición.

Figura 13: Pantalla principal de Vestium

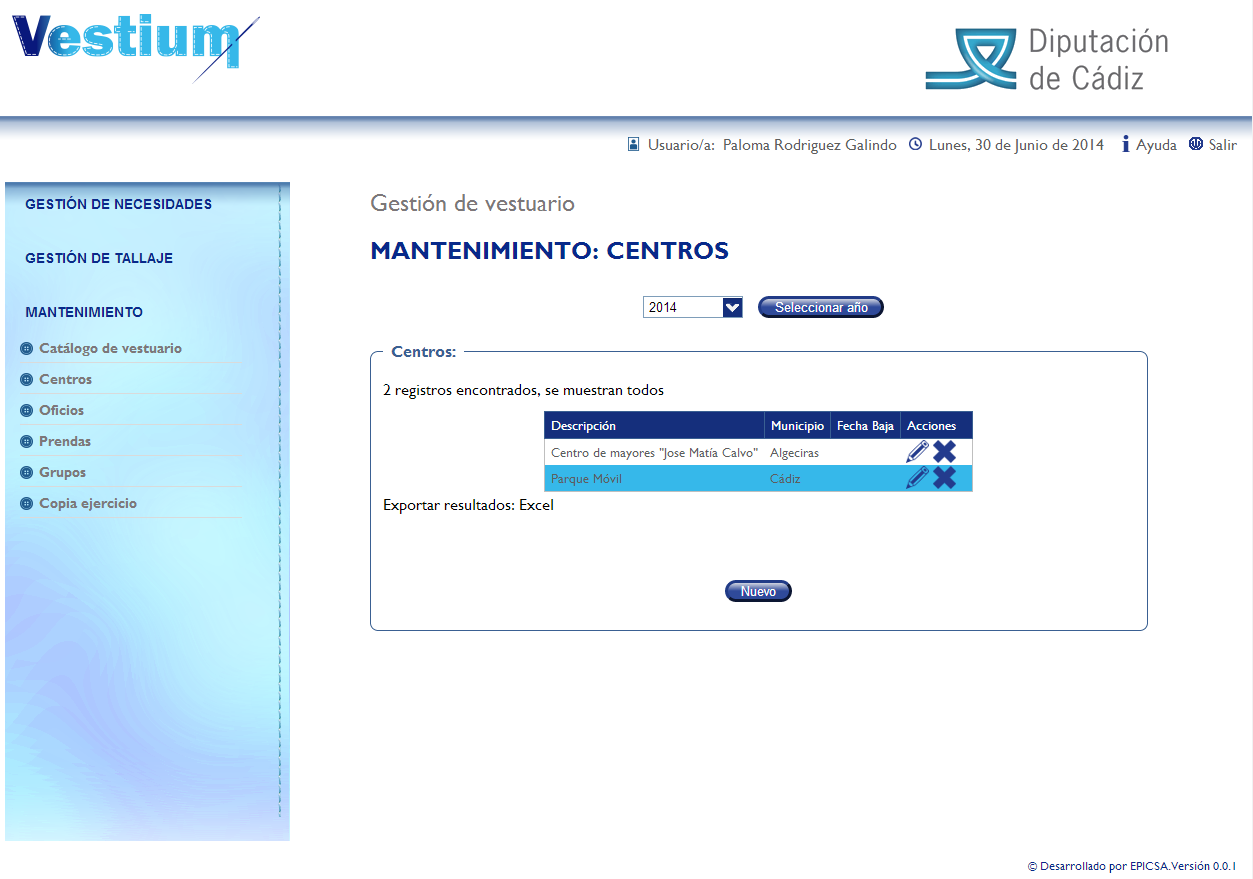
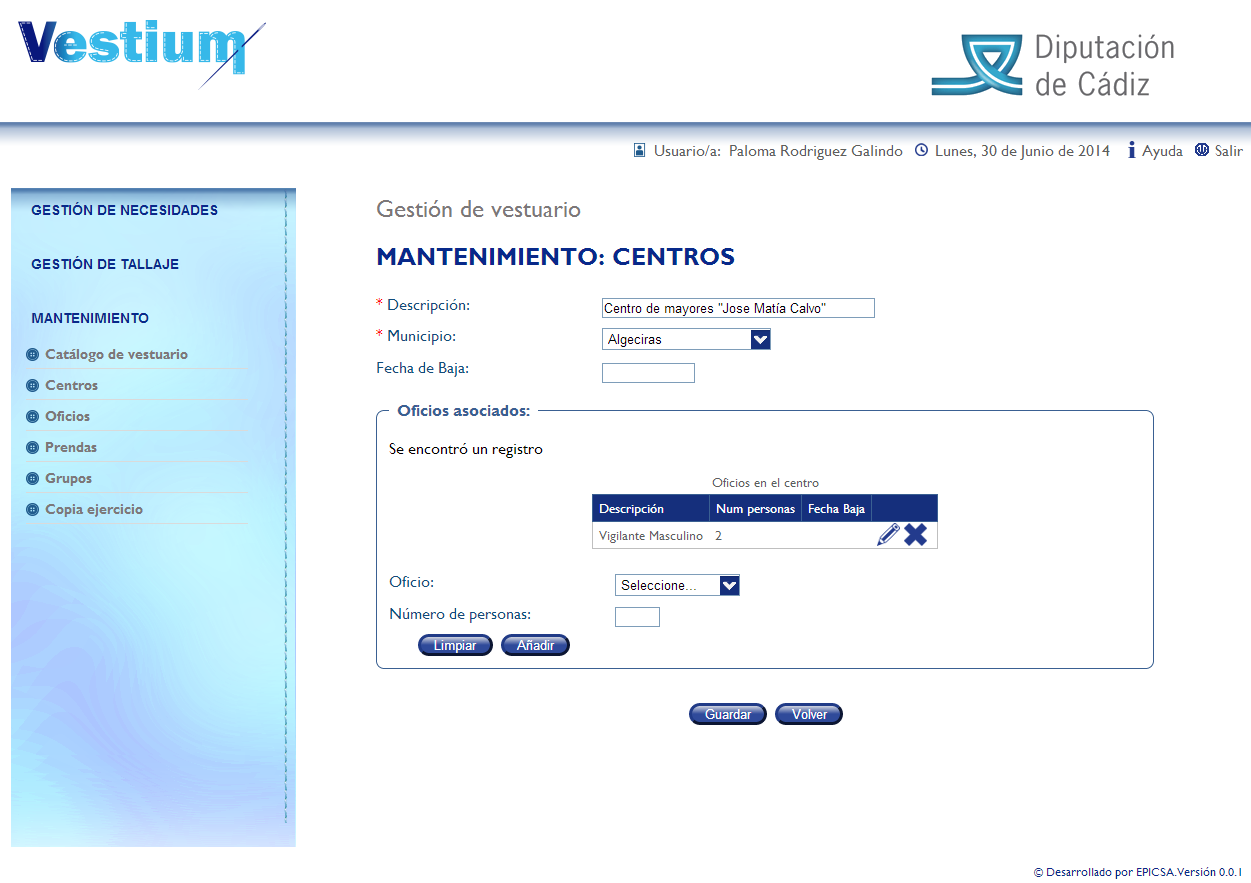


Figura 14: Pantalla de lista de mantenimiento de centros Vestium

Figura 15: Ver / editar centro Vestium

1. Disposiciones legales y estándares

* IEEE 380 Especificación de requisitos según el estándar IEEE Std. 830 - 1998 : Estándar donde se describe el proceso de creación de la especificación de requisitos software
* UNE 157001-2002: Criterios generales para la elaboración de proyectos - Febrero 2002

1. Requisitos funcionales

A continuación se indican los requisitos funcionales que han sido especificados por el cliente:

* El menú debe cargarse en base a la configuración realizada en Gestión de la Configuración.
* Debe ser posible cambiar en cualquier momento el ejercicio con el que se esté trabajando.
* Se pueda realizar una copia de las tablas de los diferentes objetos grupos, prendas, oficios y centros para comenzar a preparar el nuevo ejercicio. Cuando se realice dicha copia se debe conservar las asociaciones entre los elementos oficios pertenecientes a un centro y prendas pertenecientes a un oficio.
* El usuario debe poder visualizar, añadir y modificar información de la base de datos de los grupos, prendas, oficios, y centros, así como poder realizar modificaciones en los elementos que estén asociados.
* El borrado de los elementos en la base de datos debe ser lógico y no físico ya que, introduciendo fecha de baja de los diferentes elementos, además de controlarse los que están disponibles o no, se controla desde que momento dejan de estarlo.
* Que exista la posibilidad de consultar el catálogo de vestuario definido por los sindicatos y Diputación.
* Se generen las necesidades tantas veces se quiera y se puedan consultar las mismas. Una vez generadas se pueda realizar el alta una actuación con las necesidades en el aplicativo Contrat@. En el caso que el alta ya esté realizada debe aparecer un mensaje de aviso al usuario impidiendo la acción. Para volver a realizar el alta de la actuación debe haber un punto de menú para su reactivación.
* Una vez completado el proceso licitatorio se deberá generar el tallaje con el número de prendas reales que han sido necesarias para cubrir las necesidades de cada centro. También debe existir la posibilidad de consultar el mismo.
* Debe existir un punto de menú donde se indique el precio que se ha establecido por prenda en la licitación ya que, en el alta de la actuación del próximo ejercicio, el presupuesto aproximado vendrá calculado en base a estos precios.
* Que exista la posibilidad de realizar extracciones de todas las tablas en documentos Excel.
* Cada elemento que se inserte o modifique en la base de datos debe incluir la auditoría. Esta función va a consistir en guardar el DNI de la persona que ha realizado el alta y/o modificación, así como la fecha y la hora en la que se ha realizado la misma.

1. Atributos

Dada la utilidad de la aplicación es importante la seguridad de los datos, por ello, solo deben acceder a ella usuarios que hayan realizado la autenticación correctamente. Además para las acciones de auditoría es necesaria esta autenticación.

Toda la aplicación, incluyendo su base de datos, podrá ser trasladada en cualquier momento a otro servidor.

Análisis del sistema

En apartados anteriores se indicó que se seguía la metodología XP por lo que el proyecto se estructura en torno a las historias de los usuarios, que describen el comportamiento deseado del sistema en sus propias palabras.

Mientras se redactan las necesidades, los desarrolladores realizan el análisis del trabajo, se le asigna a cada historia un tiempo. El cliente, en función de esos factores y de sus intereses, marcan la prioridad de una historia. En la siguiente lista se puede ver junto cada historia la abreviatura que ayudará a entender mejor el diagrama de los casos de uso general. Para comenzar a listar las historias se dividirán en :

* Modulo de login
* Gestión de perfil (GP)
* Modulo de mantenimiento, las historias de este grupo son:
* Consulta de catálogo (CC)
* Listar centros (MC)
* Extracción a Excel
* Añadir / editar / eliminar centros
* Añadir / editar / eliminar oficios pertenecientes a ese centro
* Listar oficios (MO)
* Extracción a Excel
* Añadir / editar / eliminar oficios
* Añadir / editar / eliminar prendas pertenecientes a ese oficio
* Listar prendas (MP)
* Extracción a Excel
* Añadir / editar / eliminar prendas
* Editar grupo al que pertenece la prenda
* Listar grupos (MG)
* Extracción a Excel
* Añadir/editar/eliminar grupos
* Copia de ejercicio (CE)
* Módulo de gestión, las historias correspondientes a este grupo son:
* Consulta necesidad (CN)
* Generar necesidad (GN)
* Rehacer el alta de una actuación para un ejercicio(RA)
* Consulta de tallaje (CT)
* Generar de tallaje (GT)
* Generar precios prenda (GP)

1. Modelo de casos de uso

Los casos de usos ayudan a conocer a los distintos actores que utilizan la aplicación y las funciones que tienen disponibles dentro de esta además de las relaciones que aparecen entre los diferentes casos de uso.

Para cada caso de uso, se especifica el comportamiento que tiene, indicando la secuencia de las acciones y los roles de los usuarios. En la figura 16 se puede ver el diagrama general de casos de uso que tiene el sistema que permite facilitar la visión que luego que verá con más nivel de detalle para cada caso de uso en particular. El mantenimiento de grupo queda representado en la figura 18, como el mantenimiento de prendas es igual no se ha representado el caso de uso. La figura 19 representa el caso de uso mantenimiento de oficio, como en el caso anterior, es análogo al caso de uso de centros por lo que solo se ha hecho una representación. La figura 18 y 19 guardan similitud en el gráfico pero el caso de uso se comporta de manera ligeramente diferente. El caso de uso de consulta de catálogo es similar a todos los casos de consulta.

Para todos los casos de uso, si se produce un error al sistema realizar la acción, esta se aborta y el sistema muestra un mensaje indicando el motivo del error y quedando el caso de uso sin efecto.

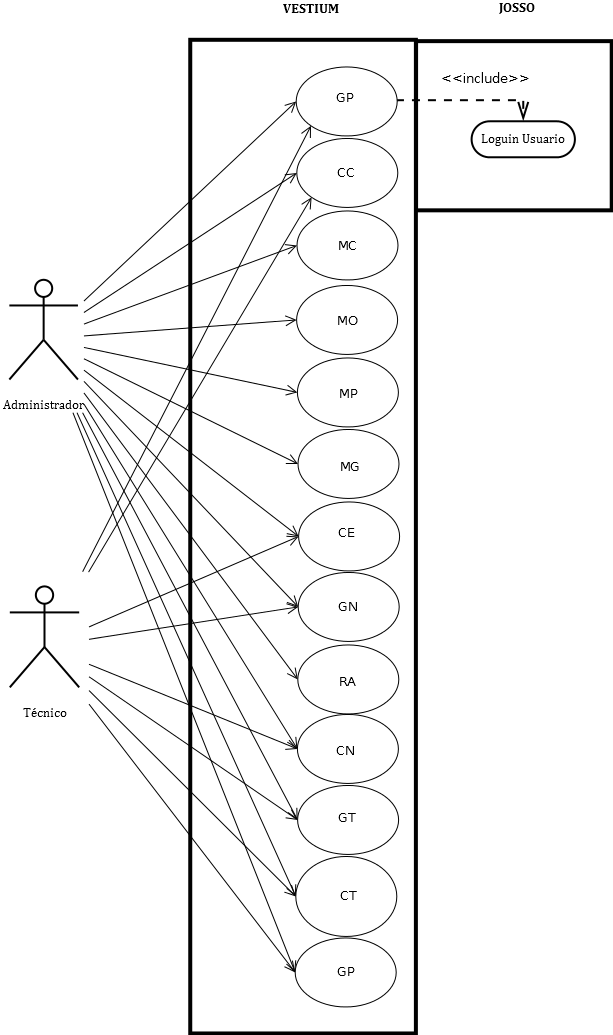


Figura 16: Diagrama de casos de uso

**Descripción del caso de uso "gestión de perfil"**

Actor principal: Administrador o técnico.

Precondiciones: El usuario debe estar dado de alta en la LDAP.

Postcondiciones: Se inicia sesión en Vestium.

Escenario principal o flujo básico: El usuario entra en la aplicación.

Escenario alternativo 1: El usuario no ha iniciado sesión en Josso.

Escenario principal o flujo básico:

1. El usuario desea entrar en la aplicación Vestium.
2. El sistema obtiene el identificador de la sesión de Josso.
3. El sistema carga el usuario en sesión.
4. El sistema carga el año del sistema en sesión.
5. El sistema carga el menú XML correspondiente al perfil del usuario.

Escenario alternativo 1:

1. El sistema detecta que el usuario no ha iniciado sesión.
2. El sistema muestra el mensaje de error y pide que el usuario introduzca su usuario y contraseña.

**Caso de uso "copia ejercicio"**

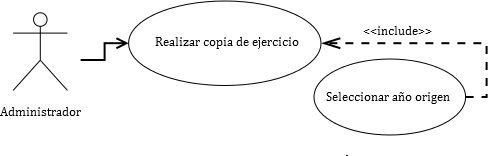


Figura 17: CU de Copia de ejercicio

Actor principal: Administrador.

Precondiciones: Hay al menos un ejercicio en la base de datos.

Postcondiciones: Se realiza una copia de un ejercicio a otro.

Escenario principal o flujo básico: El usuario realiza la copia del ejercicio.

Escenario alternativo 1: El ejercicio destino no es correcto.

Escenario Principal o flujo básico:

1. El usuario selecciona el año de origen y año destino.
2. El sistema comprueba que el año destino es correcto.
3. El sistema realiza la copia del ejercicio y muestra un mensaje de éxito.

Escenario alternativo 1:

1. El año de destino no es correcto.
2. El sistema muestra el error y solicita nuevo ejercicio destino.

**Caso de uso "Listar grupos"**

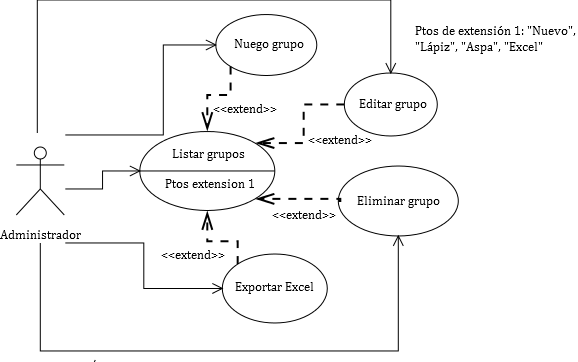


Figura 18: CU de listar grupos

Actor principal: Administrador.

Precondiciones: Ninguna.

Postcondiciones: Aparece el listado de los grupos.

Escenario principal o flujo básico: Se muestra el listado de los grupos.

Escenario alternativo 1: El usuario no selecciona ejercicio.

Escenario alternativo 2: El usuario crea un nuevo grupo.

Escenario alternativo 3: El usuario edita un grupo ya creado.

Escenario alternativo 4: El usuario elimina un grupo.

Escenario alternativo 5: El usuario exporta el listado de grupos a Excel.

Escenario principal o flujo básico: Listar grupos.

1. El usuario selecciona el ejercicio del que desea sacar el listado.
2. El sistema carga la lista con todos los grupos de ese año.
3. El usuario sale del menú.

Escenario alternativo 1:

1. El usuario no selecciona ejercicio.
2. El sistema toma por defecto el año del sistema.

Escenario alternativo 2:

1. El usuario pulsa "nuevo"
2. El sistema carga el formulario vacío.
3. El usuario introduce los datos y pulsa guardar.
4. El sistema comprueba que los datos son correctos.
5. Los datos no son correctos.
6. El sistema muestra el error y vuelve a pedir los datos al usuario.
7. El sistema guarda los datos, muestra el listado de los grupos y un mensaje de éxito.

Escenario alternativo 3:

1. El usuario pulsa el "lápiz".
2. El sistema carga el formulario con los datos del grupo.
3. El usuario realiza las modificaciones oportunas.
4. El sistema compruebas que los datos son correctos.
5. Los datos no son correctos.
6. El sistema muestra el error y vuelve a pedir los datos al usuario.
7. El sistema guarda los datos, muestra el listado y un mensaje de éxito.

Escenario alternativo 4:

1. El usuario pulsa el "aspa".
2. El sistema introduce al elemento como fecha de baja la fecha actual del sistema.
3. El sistema comprueba todas las prendas perteneciente a ese grupo y le introduce la fecha de baja si no tiene o si esta es posterior a la fecha del sistema.
4. El sistema muestra el listado e los grupos y un mensaje de éxito.

Escenario alternativo 5:

1. El usuario pulsa "Excel".
2. El sistema realiza la exportación a Excel del documento.

**Descripción del caso de uso "listar oficios"**

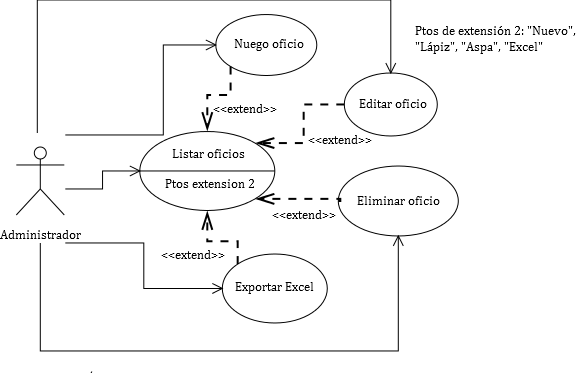


Figura 19: CU de listar oficios

Actor principal: Administrador.

Precondiciones: Ninguna.

Postcondiciones: Listar oficios.

Escenario principal o flujo básico: listar oficios.

Escenario alternativo 1: El usuario no introduce el ejercicio.

Escenario alternativo 2: El usuario crea un nuevo oficio.

Escenario alternativo 3: El usuario edita un oficio previamente creado.

Escenario alternativo 4: El usuario elimina un oficio.

Escenario alternativo 5: El usuario exporta el listado de oficios a Excel.

Escenario principal o flujo básico: Listar Oficios.

1. El usuario selecciona el ejercicio.
2. El sistema carga el listado con todos los oficios de este.
3. El usuario sale del menú.

Escenario alternativo 1:

1. El usuario no selecciona ejercicio.
2. El sistema toma por defecto el año del sistema.

Escenario alternativo 2:

1. El usuario pulsa "nuevo".
2. El sistema carga el formulario vacío.
3. El usuario introduce los datos del oficio.
4. El usuario añade asociación de prendas con el oficio y pulsa añadir.
5. El sistema comprueba que la asociación es correcta.
6. La asociación no es correcta.
7. El sistema indica el error y pide que se introduzcan los datos correctamente.
8. El usuario pulsa guardar.
9. El sistema comprueba que los datos son válidos.
10. Los datos no son válidos.
11. El sistema muestra el error y pide que el usuario vuelva a introducir los datos.
12. El sistema guarda los datos, muestra el listado de oficios y un mensaje de éxito.

Escenario alternativo 3:

1. El usuario pulsa el "lápiz".
2. El sistema carga el formulario con los datos del grupo.
3. El usuario realiza las modificaciones al oficio.
4. El usuario edita asociación de prendas con el oficio.
5. El sistema comprueba que la asociación es correcta.
6. La asociación no es correcta.
7. El sistema indica el error y pide que se introduzcan los datos correctamente.
8. El usuario pulsa guardar.
9. El sistema comprueba que los datos son válidos.
10. Los datos no son válidos.
11. El sistema muestra el error y pide que el usuario vuelva a introducir los datos.
12. El sistema guarda los datos, muestra el listado de oficios y un mensaje de éxito.

Escenario alternativo 4:

1. El usuario pulsa el "aspa".
2. El sistema introduce al elemento, como fecha de baja, la fecha actual del sistema.
3. El sistema introduce fecha de baja a todas las asociaciones de prendas con ese oficio en el caso que no tuviera o fuera posterior a la fecha del sistema.
4. El sistema comprueba todos los centros que tiene asociado ese oficio y le introduce la fecha de baja del sistema siempre que no tuviera o esta fuera posterior.
5. El sistema muestra el listado de los grupos y un mensaje de éxito.

Escenario alternativo 5:

1. El usuario pulsa "Excel".
2. El sistema realiza la exportación a Excel del documento.

**Caso de uso "generar necesidades"**

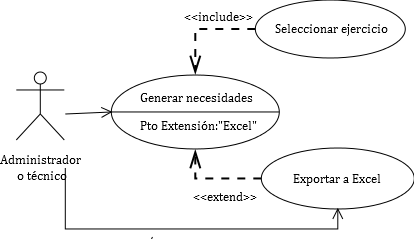


Figura 20: CU de generar necesidades

Actor principal: Administrador o técnico.

Precondiciones: Ninguna.

Postcondiciones: Generar las necesidades para el año en cuestión.

Escenario principal o flujo básico: Se generan las necesidades con éxito.

Escenario alternativo 1: Ya se han generado las necesidades con anterioridad.

Escenario alternativo 2: No existen datos para ese ejercicio.

Escenario alternativo 3: Exportar el listado de necesidades a Excel.

Escenario principal o flujo básico:

1. El usuario selecciona el ejercicio y pulsa generar necesidades.
2. El sistema comprueba que no se han generado necesidades para ese ejercicio.
3. El sistema comprueba que existen datos para ese ejercicio y genera las necesidades.
4. El sistema muestra el listado de las necesidades y un mensaje de éxito.

Escenario alternativo 1:

1. Existen necesidades para el ejercicio.
2. El sistema borra las necesidades anteriores de la base de datos.
3. El sistema genera las necesidades.

Escenario alternativo 2:

1. No existen centros para crear necesidades.
2. El sistema muestra el mensaje de error.
3. El sistema pide que se generen las necesidades .

Escenario alternativo 3:

1. El usuario pulsa "Excel".
2. El sistema realiza la exportación a Excel del documento.

**Caso de uso "consultar necesidades"**

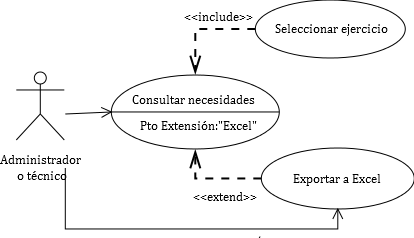


Figura 21: CU de consultar necesidades

Actor principal: Administrador o técnico.

Precondiciones: Ninguna.

Postcondiciones: Consultar las necesidades del año en cuestión.

Escenario principal o flujo básico: Se consultan las necesidades con éxito.

Escenario alternativo 1: No existen necesidades.

Escenario alternativo 2: El usuario exporta a Excel.

Escenario principal o flujo básico:

1. El usuario selecciona el filtro de consulta y pulsa seleccionar.
2. El sistema comprueba que existen necesidades para el ejercicio indicado.
3. El sistema filtra por centro y grupo indicado por el usuario.
4. El sistema muestra el listado con las necesidades.

Escenario alternativo 1:

1. No existen necesidades.
2. El sistema muestra la tabla vacía de necesidades.

Escenario alternativo 2:

1. El usuario pulsa Excel.
2. El sistema realiza la exportación a Excel.

**Caso de uso "alta de actuación"**

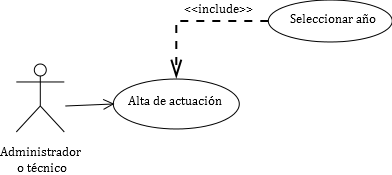


Figura 22: CU de alta de actuación

Actor principal: Administrador o técnico.

Precondiciones: Ninguna.

Postcondiciones: Se generan las necesidades para el año en cuestión.

Escenario principal o flujo básico: Se realiza el alta de la actuación en Contrat@ con éxito.

Escenario alternativo 1: No existen necesidades generadas para el ejercicio.

Escenario principal o flujo básico:

1. El usuario indica el año de la actuación y pulsa alta actuación.
2. El sistema comprueba que existan necesidades generadas.
3. El sistema realiza inserciones en la base de datos de Contrat@.
4. El sistema bloquea el alta de actuación para ese ejercicio y muestra un mensaje de error.

Escenario alternativo 1:

1. No existen necesidades para ese ejercicio.
2. El sistema muestra el error y pide que se generen las necesidades para el ejercicio.

**Caso de uso "rehacer alta"**



Figura 23: CU de rehacer alta actuación

Actor principal: Administrador .

Precondiciones: Se haya realizado una alta de actuación previamente.

Postcondiciones: El sistema vuelve a permitir realizar un alta nueva.

Escenario principal o flujo básico: Se desbloquea el caso de uso de realizar alta de actuación.

Escenario alternativo 1: No existen necesidades para el ejercicio.

Escenario principal o flujo básico:

1. El usuario selecciona el ejercicio y pulsa activar alta.
2. El sistema comprueba que existan necesidades para ese ejercicio.
3. El sistema desbloquea el alta de actuación y muestra mensaje de éxito.

Escenario alternativo 1:

1. No existen necesidades para el ejercicio en cuestión.
2. El sistema muestra el mensaje de error y sale del menú.

**Caso de uso "generar tallaje"**

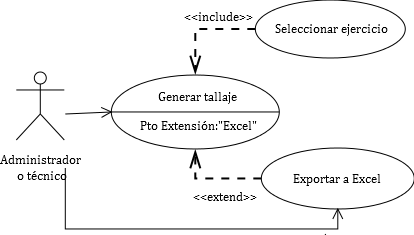


Figura 24: CU de generar tallaje

Actor principal: Administrador o técnico.

Precondiciones: Ninguna.

Postcondiciones: Generación de tallaje.

Escenario principal o flujo básico: Se genera el tallaje con éxito.

Escenario alternativo 1: El tallaje ha sido generado con anterioridad.

Escenario alternativo 2: El usuario realiza exportación a Excel.

Escenario principal o flujo básico:

1. El usuario selecciona el ejercicio.
2. El sistema comprueba que no exista tallaje generado para ese ejercicio.
3. El sistema muestra el listado con las necesidades generadas.
4. El usuario realiza las modificaciones y pulsa guardar.
5. El sistema guarda el tallaje y muestra el listado con el tallaje nuevo y un mensaje de éxito.

Escenario alternativo 1:

1. Existe tallaje generado para el ejercicio.
2. El sistema muestra el listado con el tallaje para que sea modificado.

Escenario alternativo 2:

1. El usuario pulsa Excel
2. El sistema realiza la exportación a Excel del listado.

**Caso de uso "consulta tallaje"**

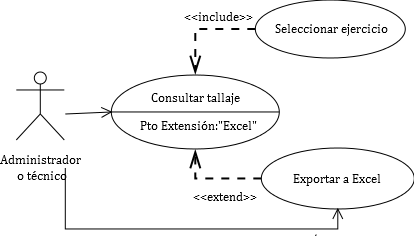


Figura 25: CU de consultar tallaje

Actor principal: Administrador o técnico.

Precondiciones: Ninguna.

Postcondiciones: Se consulta el tallaje del año en cuestión.

Escenario principal o flujo básico: Se realiza la consulta de tallaje con éxito.

Escenario alternativo 1: No exista el tallaje para el ejercicio.

Escenario alternativo 2: El usuario realiza exportación a Excel.

Escenario principal o flujo básico:

1. El usuario selecciona el año y pulsa consultar
2. El sistema comprueba que exista el tallaje para ese año
3. El sistema muestra el listado del tallaje para ese ejercicio

Escenario alternativo 1:

1. No existe tallaje para ese año.
2. El sistema muestra un mensaje de error y pide que se introduzca un nuevo ejercicio.

Escenario alternativo 2:

1. El usuario pulsa Excel.
2. El sistema realiza la exportación a Excel del listado.

**Caso de uso "generar precio prendas"**

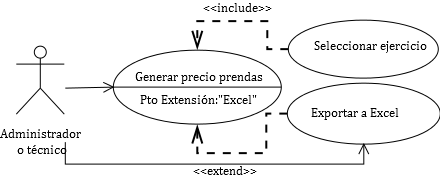


Figura 26: CU de generar precio de prendas

Actor principal: Administrador o técnico.

Precondiciones: Haya prendas dada de alta.

Postcondiciones: Se guarda el valor de la prenda.

Escenario principal o flujo básico: Se genera el precio de las prendas con éxito.

Escenario alternativo 1: No existan prendas.

Escenario alternativo 2: El usuario realiza exportación a Excel.

Escenario principal o flujo básico:

1. El usuario selecciona el año y pulsa aceptar.
2. El sistema comprueba que existan prendas para ese ejercicio.
3. El sistema muestra el listado con las prendas.
4. El usuario realiza las modificaciones oportunas en el precio y pulsa guardar.
5. E sistema guarda los nuevos valores y muestra el listado con los nuevos valores.

Escenario alternativo 1:

1. No existen prendas para ese ejercicio
2. El sistema muestra el mensaje de error y vuelve a pedir que se introduzca el año.

Escenario alternativo 2:

1. El usuario pulsa Excel.
2. El sistema realiza la exportación a Excel del listado.
3. Modelo conceptual de datos

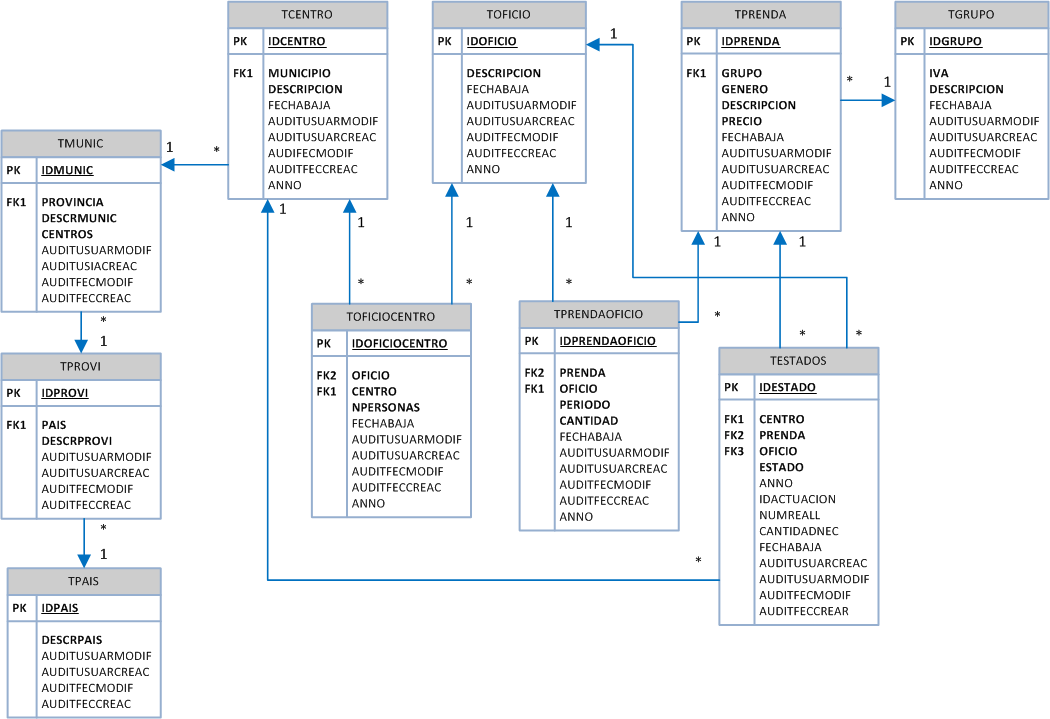
 En este apartado se describen las estructuras de datos del sistema y las relaciones que existen. Para su modelado se ha utilizado el diagrama de clases conceptual, en la figura 27 se puede ver una representación de este diagrama. A continuación se muestran las tablas donde se han descrito los atributos de las clases.

Figura 27: Diagrama de clases conceptual

|  |  |  |
| --- | --- | --- |
| TMUNIC: Tabla Municipios | | |
| Atributo | **Tipo** | **Nulidad** |
| idMunic | entero | No nulo |
| descrMunic | cadena | No nulo |
| provincia | provincia | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |

|  |  |  |
| --- | --- | --- |
| TCENTRO: Tabla Centros | | |
| Atributo | **Tipo** | **Nulidad** |
| idCentro | entero | No nulo |
| descripcion | cadena | No nulo |
| municipio | municipio | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

Figura 28: Clase Munic y Centro

|  |  |  |
| --- | --- | --- |
| TPRENDAS: Tabla Prendas | | |
| Atributo | **Tipo** | **Nulidad** |
| idPrenda | entero | No nulo |
| descripcion | cadena | No nulo |
| grupo | Grupo | No nulo |
| género | Cadena | No nulo |
| precio | entero | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

|  |  |  |
| --- | --- | --- |
| TOFICIO: Tabla Oficios | | |
| Atributo | **Tipo** | **Nulidad** |
| idOficio | entero | No nulo |
| descripcion | cadena | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

Figura 29: Clase Oficio y Prenda

|  |  |  |
| --- | --- | --- |
| TGRUPO: Tabla Grupos | | |
| Atributo | **Tipo** | **Nulidad** |
| idGrupo | entero | No nulo |
| descripcion | cadena | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

|  |  |  |
| --- | --- | --- |
| TOFICIOCENTRO: Tabla Oficios-Centro | | |
| Atributo | **Tipo** | **Nulidad** |
| idOficioCentro | entero | No nulo |
| oficio | Oficio | No nulo |
| centro | Centro | No nulo |
| nPersonas | entero | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

|  |  |  |
| --- | --- | --- |
| TPRENDAOFICIO: Tabla Prendas-Oficio | | |
| Atributo | **Tipo** | **Nulidad** |
| idPrendaOficios | entero | No nulo |
| prenda | Prenda | No nulo |
| oficio | Oficio | No nulo |
| cantidad | entero | No nulo |
| periodicidad | cadena | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

|  |  |  |
| --- | --- | --- |
| TESTADOS: Tabla Estados | | |
| Atributo | **Tipo** | **Nulidad** |
| idEstado | entero | No nulo |
| tipoEstado | entero | No nulo |
| centro | Centro | No nulo |
| prenda | Prenda | No nulo |
| oficio | Oficio | No nulo |
| cantidadNec | entero | Nulo |
| actuacion | booleano | nulo |
| numReal | entero | nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |
| fechaBaja | fecha | Nulo |
| anno | fecha | No nulo |

Figura 30: Clase Grupo y OficiosCentro

Figura 31: Clase PrendasOficio y Estados

|  |  |  |
| --- | --- | --- |
| TPROV: Tabla Provincia | | |
| Atributo | **Tipo** | **Nulidad** |
| idProv | entero | No nulo |
| descrProv | cadena | No nulo |
| pais | pais | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |

|  |  |  |
| --- | --- | --- |
| TPAIS: Tabla País | | |
| Atributo | **Tipo** | **Nulidad** |
| idPais | entero | No nulo |
| descrPais | cadena | No nulo |
| auditUruarCreac | cadena | No nulo |
| auditUruarModif | cadena | No nulo |
| auditFechaCreac | fecha | No nulo |
| auditFechaModif | fecha | No nulo |

Figura 32: Clase Prov y País

1. Modelo de comportamiento

En este apartado se describe el modelo de comportamiento que tiene el sistema, donde se representa la iteración de los usuarios y el sistema.

Los diagramas de secuencia del sistema (DSS) están en notación UML. En ellos se pueden ver las iteraciones de los atores y las operaciones iniciadas por los mismos. Un diagrama de secuencia del sistema es un gráfico que muestra para un escenario de un caso de uso los eventos que generan los actores externos y su orden. Los contratos de las operaciones describen el efecto de las operaciones del sistema.

Dado que los DSS de muchos casos de uso guardan gran similitud, sólo se muestran los más relevantes. Del mismo modo los contratos de los eventos se describen una sola vez, en la primera ocasión que aparezcan.

**DSS del caso de uso "gestión de perfil"**

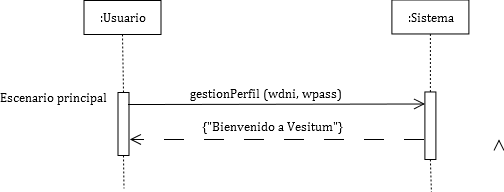


Figura 33: DSS del caso de uso de gestión de perfil

**CONTRATOS**

**Operación: gestiónPerfil (wdni,wpass).**

**Responsabilidades:** Iniciar sesión en Vestium.

**Referencias cruzadas:** Caso de Uso Gestión perfil.

**Precondiciones:** Existe un objeto usuario en la LDAP con dni = wdni.

**Postcondiciones:**

* Se crea el XML con el menú correspondiente al perfil del usuario.
* Se inicia sesión del usuario.

**DSS del caso de uso "copia de ejercicio"**

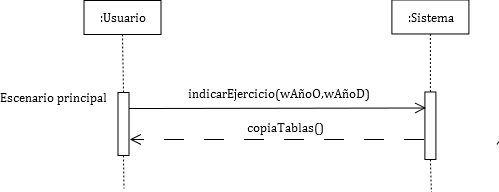


Figura 34: DSS del caso de uso de copia de ejercicio

**CONTRATOS**

**Operación: indicarEjercicio(wAñoO,wAñoD).**

**Responsabilidades:** Indicar origen y destino de la copia.

**Referencias cruzadas:** Caso de Uso copia de ejercicio.

**Precondiciones:** Ninguna.

**Postcondiciones:**

* El sistema busca todas las instancias de todas las tablas con anno = wAñoO.

**Operación: copiaTablas()**

**Responsabilidades:** Realizar la copia de de las tablas del ejercicio wAñoO .

**Referencias cruzadas:** Caso de Uso copia de ejercicio.

**Precondiciones:**

* Existe un objeto centro con centro.anno = wAñoO.
* No exista un objeto centro con centro.anno = wAñoD.

**Postcondiciones:**

* El sistema crea una instancia por cada objeto O de la tabla T del año = wAñoO en la tabla T con O.anno = wAñoD.

**DSS del caso de uso "listar grupos"**

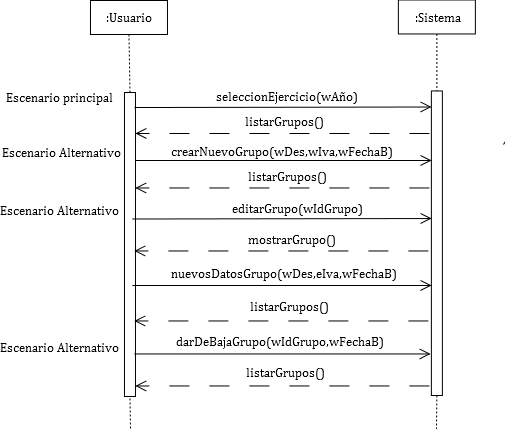


Figura 35: DSS del caso de uso de listar grupos

**CONTRATOS**

**Operación: seleccionEjercicio(wAño).**

**Responsabilidades:** Seleccionar el año del que queremos ver los grupos.

**Referencias cruzadas:** Caso de Uso listar grupo.

**Precondiciones:**

* Exista al menos una instancia grupo (g) con g.anno = wAño.

**Postcondiciones:**

* El sistema realiza una búsqueda todas las instancias de grupo con g.anno = wAño.

**Operación: listarGrupos()**

**Responsabilidades:** Mostrar el listado con todas las instancias de grupos (g) con g.anno = wAño.

**Referencias cruzadas:** Caso de Uso listar grupos.

**Precondiciones:**

* Existe un objeto grupo con grupo.anno = wAño.

**Postcondiciones:**

* El sistema muestra un listado con todas las instancias de grupos con g.anno = wAño.

**Operación: crearNuevoGrupo(Wdes,wIva,wFechaB).**

**Responsabilidades:** Crear una nueva instancia de grupo G.

**Referencias cruzadas:** Caso de Uso listar grupo.

**Precondiciones:**

* No exista ninguna instancia grupo con grupo.descripcion = wDes.

**Postcondiciones:**

* El sistema crea una instancia de grupo (g) con g.descripcion = wDes, g.iva = wIva, g.fechaBaja = wFechaB y g.anno = wAño..

**Operación: editarGrupo(wIdGrupo).**

**Responsabilidades:** Indicar al sistema el grupo que se quiere editar.

**Referencias cruzadas:** Caso de Uso listar grupos.

**Precondiciones:**

* Existe un objeto grupo con grupo.idGrupo = wIdGrupo.

**Postcondiciones:**

* El sistema busca los datos de la instancia grupo (g) con g.idGrupo = wIdGrupo.

**Operación: mostrarGrupo().**

**Responsabilidades:** Mostrar al usuario los datos del grupo.

**Referencias cruzadas:** Caso de Uso listar grupos.

**Precondiciones:** Ninguna.

**Postcondiciones:**

* El sistema muestra los datos de la instancia grupo (g) con g.idGrupo = wIdGrupo.

**Operación: nuevosDatos(wDes,wIva,wFechaB).**

**Responsabilidades:** Indicar al sistema los nuevos datos del grupo.

**Referencias cruzadas:** Caso de Uso listar grupos.

**Precondiciones:** Ninguna.

**Postcondiciones:**

* El sistema asigna a la instancia grupo g con g.idGrupo = widGrupo la descripción g.descripcion = wDes, el iva g.iva = wIva y la fecha de baja g.fechaBaja = wFechaB.

**Operación: darDeBaja(widGrupo,wFechaB)**

**Responsabilidades:** Dar de baja al grupo con grupo.idGrupo = widGrupo.

**Referencias cruzadas:** Caso de Uso listar grupos.

**Precondiciones:**

* Exista una instancia de grupo con grupo.idGrupo = widGrupo.

**Postcondiciones:**

* El sistema busca el grupo g con g.idGrupo = widGrupo.
* El sistema introduce la fecha de baja wFechaB en la instancia g.fechaBaja.
* El sistema busca todas las instancias de prendas asociadas a ese grupo e introduce para cada prenda p la fecha de baja p.fechaBaja = wFechaB siempre y cuando no tuviera o si tuviera fuera posterior a la fecha de baja wFechaB.
* El sistema busca todas las instancias de prendaOficio (preOfi) de cada una de las instancias de prendas e introduce la fecha de baja preOfi.fechaBaja = wFechaB siempre y cuando no tuviera o si la tuviera fuera posterior a la fecha de baja wFechaB.

**DSS del caso de uso "listar oficios"**

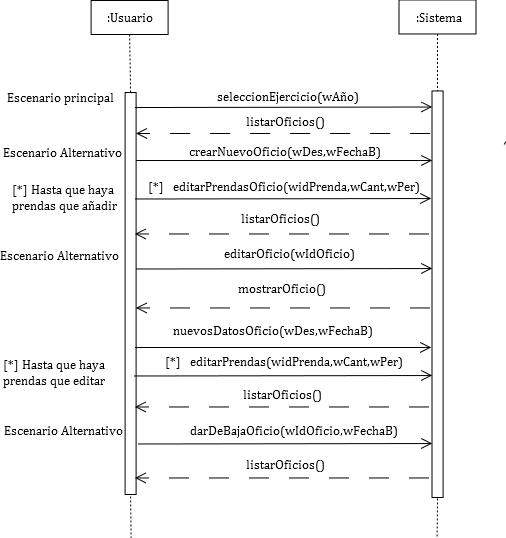


Figura 36: DSS del caso de uso de listar oficios

**CONTRATOS**

**Operación: añadirPrendasOficio(widPrenda,wCant,wPer).**

**Responsabilidades:** Crear asociaciones de prendas con el oficio.

**Referencias cruzadas:** Caso de Uso listar oficios.

**Precondiciones:** Exista una instancia de prenda con prenda.widPrenda = idPrenda.

**Postcondiciones:**

* El sistema crea una nueva instancia de prendaOficio (prendaOfi) con prendaOfi.idPrenda = widPrenda, prendaOfi.idOficio = widOficio, prendaOfi.cantidad = wCant y prendaOfi.periodicidad = wPer.

**Operación: editarPrendas(widPrenda,wCant,wPer).**

**Responsabilidades:** Editar asociaciones de prendas con el oficio.

**Referencias cruzadas:** Caso de Uso listar oficios.

**Precondiciones:** Exista una instancia de prendaOficio con prendaOficio.idPrenda = widPrenda.

**Postcondiciones:**

* El sistema edita la instancia de prendaOficio (prendaOfi) con prendaOfi.idPrenda = widPrenda, prendaOfi.idOficio = wIdOficio, prendaOfi.cantidad = wCant y prendaOfi.periodicidad = wPer.

**Operación: darDeBaja(widPrenda,wFechaB)**

**Responsabilidades:** Dar de baja el oficio con oficio.idOficio = widOficio.

**Referencias cruzadas:** Caso de Uso listar oficios.

**Precondiciones:**

* Exista una instancia de oficios con oficio.idOficio = widOficio.

**Postcondiciones:**

* El sistema busca el oficio o con o.idOficio= widOficio.
* El sistema introduce la fecha de baja wFechaB en la instancia o.fechaBaja.
* El sistema busca todas las instancias de prendasOficios a ese oficio e introduce para cada prendaOficio (po) la fecha de baja po.fechaBaja = wFechaB siempre y cuando no tuviera o si tuviera fuera posterior a la fecha de baja wFechaB.
* El sistema busca todas las instancias de oficioCentro (oc) de cada una de las instancias de oficios e introduce la fecha de baja oc.fechaBaja = wFechaB siempre y cuando no tuviera o si la tuviera fuera posterior a la fecha de baja wFechaB.

**DSS del caso de uso "generar necesidades"**

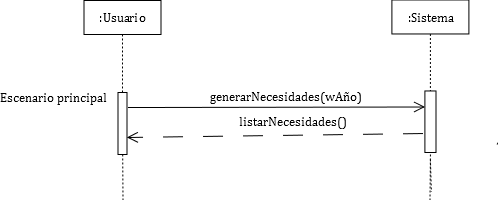


Figura 37: DSS del caso de uso de generar necesidades

**CONTRATOS**

**Operación: generarNecesidades(wAño).**

**Responsabilidades:** Se generan las necesidades para el año en cuestión.

**Referencias cruzadas:** Caso de Uso generar necesidades.

**Precondiciones:** Exista al menos un centro con centro.anno = wAño.

**Postcondiciones:**

* El sistema busca todas las instancias de centro (c) con c.anno = wAño.
* Por cada instancia de centro el sistema busca todas las intancias de centroOficio con (oc) con oc.idCentro = c.idCentro.
* Por cada instancia oficioCentro el sistema busca todas las instancias de prendaOficio (po) con po.idOficio = oc.idOficio.
* El sistema crea una instancia de Estados e con e.tipoEstado = 1 (se trata de una necesidad), e.anno = wAño y por cada Centro c, Oficio o y Prenda p calcula las necesidades conforme:

oc.nPersonas \* po.cantidad \* po.periodicidad = e.CantidadNec

**DSS del caso de uso "consultar necesidades"**

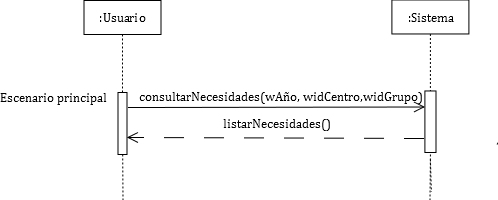


Figura 38: DSS del caso de uso de consultar necesidades

**CONTRATOS**

**Operación: consultarNecesidades(wAño,widCentro,widGrupo).**

**Responsabilidades:** Se consultan las necesidades con el filtro

**Referencias cruzadas:** Caso de Uso consultar necesidades.

**Precondiciones:** Exista al menos una instancia de Estado con estado.tipoEstado = 1 y estado.anno = wAño.

**Postcondiciones:**

* El sistema busca todas las intancias de estados con estado.tipoEstado = 1, estado.idCentro = widCentro y estado.widGrupo = widGrupo.

**DSS del caso de uso "alta de actuación"**

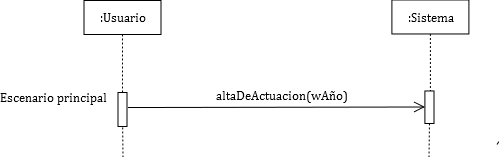


Figura 39: DSS del caso de uso de alta de actuación

**CONTRATOS**

**Operación: altaDeActuacion(wAño).**

**Responsabilidades:** Se realiza el alta de actuación en el aplicativo de Contrat@

**Referencias cruzadas:** Caso de Uso alta de actuación

**Precondiciones:** Exista al menos una instancia de Estado con estado.tipoEstado = 1 y estado.anno = wAño.

**Postcondiciones:**

* El sistema busca todas las intancias de estado(s) con e.tipoEstado = 1 y e.anno = wAño.
* El sistema calcula el importe aproximado de la actuación en base al precio de las prendas.
* El sistema crea una instancia en la tabla cont\_tactua (c) del aplicativo Contrat@ con c.cod\_entidad = 4 (Diputación), c.tipo = 3 (suministros), c.situa\_actua = 0, c.descr\_actua = Necesidades para el año de vestuario, c.exento\_iva\_sn = no, c.servi\_petic = Área de Hacienda, Recaudación y Contratación, c.impor\_actua = importe actuación.
* El sistema crea una instancia en la tabla cont\_tod\_sum\_serv (t) del aplicativo Contrat@ con t.actuacion = c.id, t.area\_peticionaria = Área de Hacienda, Recaudación y Contratación, t.centro\_destino = Área de Hacienda, Recaudación y Contratación, t.centro\_pericionario = Área de Hacienda, Recaudación y Contratación, t.domicilio\_destino = Plaza de España s/n, t.lugar\_solicitud = Cádiz, t.numero\_prorrogas = 0, t.servicio\_peticionario = Contratación , t.fecha\_solicitud = fecha del sistema.
* El sistema crea una instancia en la tabla cont\_tartic\_sum (s) por cada instancia necesidad con s.actuacion = c.actuacion, s.canti = e.CantidadNec, s.descr\_art = e.idPrenda.descripcion.
* El sistema introduce por cada instancia estado e e.actuacion = true.

**DSS del caso de uso "rehacer alta de actuación"**

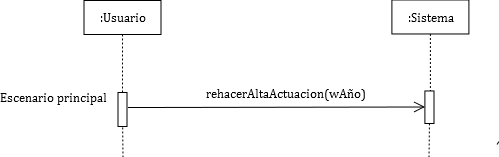


Figura 40: DSS del caso de uso de rehacer alta actuación

**CONTRATOS**

**Operación: rehacerAltaActuacion(wAño).**

**Responsabilidades:** Se vuelve a activar la posibilidad de realizar alta de Contrat@ para el año wAño

**Referencias cruzadas:** Caso de Uso rehacer alta de actuación

**Precondiciones:**

* Exista al menos una instancia de estados con estado.tipoEstado = 1
* Se haya dado con anterioridad el alta de actuación en Contrat@

**Postcondiciones:**

* El sistema busca todas las instancias de estado (e) con e.tipoEstado = 1 y e.anno = wAño.
* El sistema inserta por cada instancia e.actuacion = false.

**DSS del caso de uso "generar tallaje"**

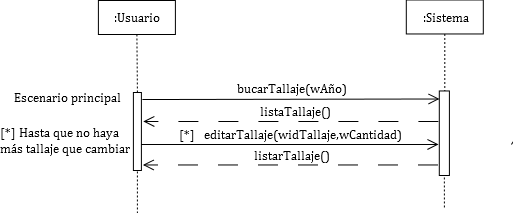


Figura 41: DSS del caso de uso de generar tallaje

**CONTRATOS**

**Operación: bucarTallaje(wAño).**

**Responsabilidades:** Se indica al sistema el tallaje que se desea realizar

**Referencias cruzadas:** Caso de Uso generar tallaje

**Precondiciones:**

* Exista al menos una instancia de estados con estado.tipoEstado = 1 con estado.anno = wAño.

**Postcondiciones:**

* El sistema busca todas las instancias de estado (e) con e.tipoEstado = 1 y e.anno = wAño.

**Operación: editarTallaje(widTallaje,wCantidad).**

**Responsabilidades:** Generar el tallaje para el ejercicio

**Referencias cruzadas:** Caso de Uso generar tallaje

**Precondiciones:**

* Exista al menos una instancia de estados con estado.tipoEstado = 1 con estado.anno = wAño.

**Postcondiciones:**

* El sistema busca todas las instancias de estado (e) con e.tipoEstado = 1 y e.anno = wAño.
* El sistema crea por cada instancia e encontrada una instancia de estado s con s.tipoEstado = 2 (tallaje) con s.idCentro = e.idCentro, s.idOficio = e.idOficio, s.idPrenda = e.idPrenda y e.numReal = wCantidad y e.anno = wAño.

**DSS del caso de uso "generar precios prendas"**



Figura 42: DSS del caso de uso de generar precios prendas

**CONTRATOS**

**Operación: editarPrendas(widPrenda,wPrecio).**

**Responsabilidades:** Se guarda el precio por prenda que ha ofertado el licitador

**Referencias cruzadas:** Caso de Uso generar precio prenda

**Precondiciones:**

* Exista al menos una instancia de prenda p con p.anno = wAño

**Postcondiciones:**

* El sistema guarda para la instancia de prenda p con p.idPrenda = widPrenda con p.precio = wPrecio.

Diseño del sistema

1. Arquitectura de sistema software

Este sistema requiere su descomposición en grupos y cada uno de ellos con un nivel de abstracción determinado. Cada uno de los componentes se deben poder reutilizar o reemplazar por otras implementaciones por lo que se debe intentar en la medida de lo posible favorecer la cohesión y disminuir el acoplamiento. Para solucionar esto se ha recurrido a la arquitectura en capas. En la figura 43 se puede observar una representación de esta arquitectura.

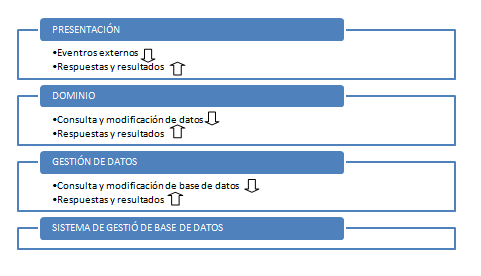


Figura 43: Arquitectura por capas

Capa de **presentación**: Esta capa es la responsable de las iteraciones con el usuario. Ella es la que ordena la ejecución de las acciones y comunica los resultado de las mismas a los usuarios, además, se ocupa del mantenimiento de ventanas.

Capa de **dominio**: Es la responsable de la implementación de las funcionalidades del sistema. Recibe, controla y ejecuta las acciones y envía las respuestas a la capa de presentación.

Capa de **gestión de datos**: Es la responsable de gestionar la base de datos. Permite que el dominio ignore donde se almacenan los datos.

**El modelo vista controlador (MVC)**

Para la realización de este proyecto se ha optado por usar el modelo vista controlador [11] ya que tiene como principal ventaja permitir separar los componentes del programa, precisamente, lo que se ha pretendido con el modelo en tres capas, en la figura 44 [12] podemos ver una representación de esta estructura. Este modelo se divide en :

* **Modelo**: Encargado de administrar la lógica de la aplicación. Su finalidad es servir de abstracción a algún proceso. El modelo desconoce la existencia de las vista y del controlador. En la práctica debe existir interfaces que permitan a los módulos comunicarse entre sí por lo que el modelo sugiere que esté dividido en dos partes.
* **Modelo de dominio** y **modelo de la aplicación**. El primero es el conjunto de clases que un ingeniero de software modela al analizar el problema que desea resolver. El segundo es un conjunto de clases que se relacionan con el modelo de dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar sobre los cambios que se dieran en el modelo de dominio.
* **Vista**: Representación visual. Una vista obtiene solamente la información que necesita para desplegar y se analiza cada vez que el modelo de dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.
* **Controlador**: Se encarga de captar los eventos que genera el usuario. Permite al usuario interactuar.

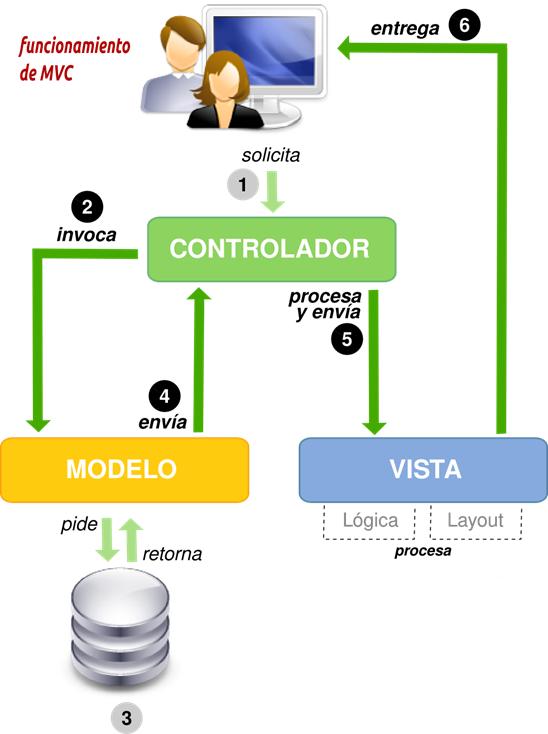


Figura 44: Modelo vista controlador

1. Diseño de base de datos

En la gestión de la base de datos se ha utilizado Oracle. Se ha implementado una base de datos relacional ya que este tipo de modelo permite establecer conexiones entre datos almacenados en tablas, permite mantener la integridad referencial.

**Diseño lógico de base de datos**

* TMUNIC = {idMunic, descrMunic,provincia, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif }
* TCENTRO = { idCentro, descripcion, municipio, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno}
* TOFICIO = { idOficio, descripcion, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno }
* TOFICIOCENTRO = {idOfiCen, idOficio, idCentro, nPersonas, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno }
* TPRENDA = { idPrenda, descripcion, grupo, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno }
* TPRENDAOFICIO = {idPreOfi, idPrenda, idOficio, cantidad, periodicidad, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno }
* TGRUPO = { idGrupo, descripcion, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno }
* TESTADOS = { idEstado, tipoEstado, idCentro, idPrenda, idOficio, cantidadNec, numReal, actuacion, fechaBaja, auditUruarCreac, auditUruarModif, auditFechaCreac, auditFechaModif , anno }

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TMUNIC: Tabla Municipios | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idMunic | entero | No nulo | Si | No |
| descrMunic | cadena | No nulo | No | No |
| provincia | provincia | No nulo | No | Sí |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |

Figura 45: Definición lógica de la tabla de municipios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TCENTRO: Tabla Centros | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idCentro | entero | No nulo | Si | No |
| descripcion | cadena | No nulo | No | No |
| municipio | municipio | No nulo | No | Sí(TMUNIC.idMunic) |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 46: Definición lógica de la tabla Centro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TOFICIO: Tabla Oficios | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idOficio | entero | No nulo | Si | No |
| descripcion | cadena | No nulo | No | No |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 47: Definición lógica de la tabla Oficio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TPRENDAS: Tabla Prendas | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idPrenda | entero | No nulo | Si | No |
| descripcion | cadena | No nulo | No | No |
| grupo | Grupo | No nulo | No | Sí(TGRUPO.idGrupo) |
| precio | entero | No nulo | No | No |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 48: Definición lógica de la tabla Prenda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TGRUPO: Tabla Grupos | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idGrupo | entero | No nulo | Si | No |
| descripcion | cadena | No nulo | No | No |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 49: Definición lógica de la tabla Grupo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TOFICIOCENTRO: Tabla Oficios-Centro | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idOficioCentro | entero | No nulo | Si | No |
| oficio | Oficio | No nulo | No | Si(TOFICIO.idOficio) |
| centro | Centro | No nulo | No | Si(TCENTRO.idCentro) |
| nPersonas | entero | No nulo | No | No |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 50: Definición lógica de la tabla OficiosCentro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TPRENDAOFICIO: Tabla Prendas-Oficio | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idPrendaOficios | entero | No nulo | Si | No |
| prenda | Prenda | No nulo | No | Si(TPRENDA.idPrenda) |
| oficio | Oficio | No nulo | No | Si(TOFICIO.idOficio) |
| cantidad | entero | No nulo | No | No |
| periodicidad | cadena | No nulo | No | No |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 51: Definición lógica de la tabla PrendasOficio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TESTADOS: Tabla Estados | | | | |
| Atributo | **Tipo** | **Nulidad** | **PK** | **FK** |
| idEstado | entero | No nulo | Si | No |
| tipoEstado | entero | No nulo | No | No |
| centro | Centro | No nulo | No | Si(TCENTRO.idCentro) |
| prenda | Prenda | No nulo | No | Si(TPRENDA.idPrenda) |
| oficio | Oficio | No nulo | No | Si(TOFICIO.idOficio) |
| cantidadNec | entero | Nulo | No | No |
| actuacion | booleano | nulo | No | No |
| numReal | entero | nulo | No | No |
| auditUruarCreac | cadena | No nulo | No | No |
| auditUruarModif | cadena | No nulo | No | No |
| auditFechaCreac | fecha | No nulo | No | No |
| auditFechaModif | fecha | No nulo | No | No |
| fechaBaja | fecha | Nulo | No | No |
| anno | fecha | No nulo | No | No |

Figura 52: Definición lógica de la tabla Estados

1. Diseño detallado del sistema

Se describen a continuación las clases y las operaciones que se pueden encontrar en cada una de las clases.

|  |  |
| --- | --- |
| TPais | |
| Operación | **Descripción** |
| getPais(Integer id) | Devuelve el objeto país asociado con idPais "id" |
| guardarPais(Pais pais) | Guarda el objeto pais "pais" en la BD |
| findPaises() | Devuelve un listado de objetos pais de la BD |
| findPaises(String textoBusqueda) | Devuelve un listado de objetos pais con descrPais "textoBusqueda" |
| findPaises(List<Integer> ids) | Devuelve un listado de objetos pais asociados a la lista de ids. |

Figura 53: Operaciones de la clase TPais

|  |  |
| --- | --- |
| TProv | |
| Operación | **Descripción** |
| getProvicia(Integer id) | Devuelve el objeto provincia asociado con idProv "id" |
| guardarProvincia(provincia) | Guarda el objeto provincia "provincia" en la BD |
| findProvincias() | Devuelve un listado de objetos provincia de la BD |
| findProvincias(String textoBusqueda) | Devuelve un listado de objetos provincia con descrProv "textoBusqueda" |
| findProvincias(ids) | Devuelve un listado de objetos provincia asociados a la lista de ids. |
| findProvinciasPorPais(pais) | Devuelve un listado de objetos provincia asociado al objeto pais "pais" |

Figura 54: Operaciones de la clase TProv

|  |  |
| --- | --- |
| TMunic | |
| Operación | **Descripción** |
| getMunicipio(Integer id) | Devuelve el objeto municipio con idMunic "id" |
| guardarMunicipio(municipio) | Guarda el objeto municipio "municipio" en la BD |
| findMunicipios() | Devuelve un listado de objetos municipio de la BD |
| findMunicipios(String textoBusqueda) | Devuelve un listado de objetos municipio con descrMunic "textoBusqueda" |
| findMunicipios(ids) | Devuelve un listado de objetos municipio asociados a la lista de ids. |
| findMunicipiosPorProvi(prov) | Devuelve un listado de objetos municipio asociado al objeto provincia "prov" |

Figura 55: Operaciones de la clase TMunic

|  |  |
| --- | --- |
| TPrenda | |
| Operación | **Descripción** |
| getPrenda(id) | Devuelve el objeto prenda asociado a la idPrenda "id" |
| guardarPrenda(prenda) | Guarda el objeto prenda "prenda" en la BD |
| eliminarPrenda(id) | Eliminación lógica del objeto prenda asociado a la idPrenda "id" introduciendo como fecha de baja la fecha actual del sistema |
| findPrenda(anno) | Devuelve un listado de objetos prenda asociadas al año "anno" que estén dadas de alta en el momento de la consulta |
| findAllPrendas(anno) | Devuelve un listado de todos los objetos prenda asociadas al año "anno" |
| findPrendasPorGrupo(anno,grupo) | Devuelve un listado de objetos prenda asociadas al año "anno" y al objeto grupo "grupo" dados de alta el día que se realiza la consulta |
| findPrendaDescripcion(anno,textoBusqueda) | Devuelve un objeto prenda asociado al año "anno" y la descripción "textoBusqueda" que esté dada de alta en el momento de realizar la consulta |
| findPrendasPorIdGrupo(anno,idGrupo) | Devuelve un listado de objetos prenda asociadas al año "anno" y al objeto grupo con idGrupo "idGrupo" dadas de alta el día que se realiza la consulta |
| findAllPrendasPorIdGrupo(anno,idGrupo) | Devuelve un listado de objetos prenda asociadas al año "anno" y al objeto grupo con idGrupo "idGrupo" |

Figura 56: Operaciones de la clase TPrenda

|  |  |
| --- | --- |
| TCentro | |
| Operación | **Descripción** |
| getCentro(id) | Devuelve el objeto centro con idCentro "id" |
| guardarCentro(centro) | Guarda el objeto centro "centro" en la BD |
| guardarNuevoCentro(centro, oficios) | Guarda el objeto centro "centro" en la BD y sus oficios asociados de la clase oficioscentro "oficios" |
| eliminarCentro(Integer id) | Eliminación lógica del objeto centro con idCentro "id" introduciendo como fecha de baja la fecha actual del sistema |
| findCentros(Integer anno) | Devuelve un listado de objetos centro asociado al año "anno" que estén dados de alta en el momento de la consulta |
| findAllCentros(Integer anno) | Devuelve un listado de todos los objetos centro asociado al año "anno" |
| findCentros(anno,ids) | Devuelve un listado de objetos centro asociado al año "anno" e idCentro "ids" que estén dados de alta el día que se realiza la consulta |
| findAllAnnos() | Devuelve el listado de todos los años que se encuentran en los objetos centros y atributo anno |
| findCentros(anno,o) | Devuelve un listado de objetos centro asociado al año "anno" y al objeto oficio "o" que estén dados de alta el momento que se realiza la consulta |
| findCentrosPorIdOficio(anno,idOficio) | Devuelve un listado de objetos centros asociados al año "anno" y al objeto oficio con idOficio "idOficio" que estén dados de alta en el momento de realizar la consulta |
| findCentroDescripcion(anno,textoBusqueda) | Devuelve un objeto centro asociado al año "anno" y la descripción "textoBusqueda" que estén dados de alta en el momento de realizar la consulta |
| findAllCentros(Integer anno, String textoBusqueda) | Devuelve una lista de objetos centro que estén asociados con el año "anno" y la descripción "textoBusqueda" |

Figura 57: Operaciones de la clase TCentro

|  |  |
| --- | --- |
| TOficio | |
| Operación | **Descripción** |
| getOficio(id) | Devuelve el objeto oficio asociado con idOficio "id" |
| guardarOficio(oficio) | Guarda el objeto oficio "oficio" en la BD |
| guardarNuevoOficio(oficio, prendas) | Guarda el objeto oficio en la BD y sus prendas asociadas de la clase prendaoficio "prendas" |
| eliminarOficio (id) | Eliminación lógica del objeto oficio asociado con idOficio "id" introduciendo como fecha de baja la fecha actual del sistema |
| findOficios(anno) | Devuelve un listado de objetos oficio asociados al año "anno" que estén dados de alta en el momento de la consulta |
| findAllOficios(anno) | Devuelve un listado de todos los objetos oficio asociados al año "anno" |
| findOficio(anno,ids) | Devuelve un listado de objetos oficio asociados al año "anno" e idOficio "ids" que estén dados de alta el día que se realiza la consulta |
| findOficios(anno,c) | Devuelve un listado de objetos oficio asociados al año "anno" y al objeto centro "c" que estén dados de alta el momento que se realiza la consulta |
| findOficiosPorIdCentro(anno,idCentro) | Devuelve un listado de objetos oficios asociados al año "anno" y al objeto centro con idCentro "idCentro" dados de alta el momento que se realiza la consulta |
| findOficioDescripcion(anno,textoBusqueda) | Devuelve un objeto oficio asociado al año "anno" y la descripción "textoBusqueda" que esté dado de alta en el momento de realizar la consulta |
| findOficiosPorCentro(anno,c); | Devuelve una lista de objetos oficio que estén asociados con el año "anno" y el objeto centro "c" dados de alta el día que se realiza la consulta |
| findAllOficiosPorIdCentro(anno,idCentro) | Devuelve un listado de objetos oficios asociados al año "anno" y al objeto centro con idCentro "idCentro" |

Figura 58: Operaciones de la clase TOficio

|  |  |
| --- | --- |
| TOficioCentro | |
| Operación | **Descripción** |
| getCentros(oficio) | Devuelve una lista de objetos oficiocentro asociados al objeto oficio "oficio" que estén dados de alta en el momento de realizar la consulta |
| getOficios(centro) | Devuelve una lista de objetos oficiocentro asociados al objeto centro "centro" que estén dados de alta en el momento de realizar la consulta |
| getObjetoOficiosPorcentroId(idCentro,idOficio) | Devuelve el objeto oficiocentro asociado al objeto oficio con idOficio "idOficio" y al objeto centro con idCentro "idCentro" dado de alta en el momento de realizar la consulta |
| getObjetoOficiosPorCentro(centro,oficio) | Devuelve el objeto oficiocentro asociado al objeto oficio "oficio" y al objeto centro "centro" dado de alta en el momento de realizar la consulta |
| guardarObjetoOficiosPorCentro(ofiCentro) | Guarda el objeto oficiocentro "ofiCentro" en la BD |
| eliminarOficiosPorCentro(ofiCentro) | Eliminación lógica del objeto oficiocentro "ofiCentro" introduciendo como fecha de baja la fecha actual del sistema |
| eliminarObjetoOficiosPorCentro(centro,oficio) | Eliminación lógica del objeto oficiocentro asociado al objeto oficio "oficio" y al objeto centro "centro" introduciendo como fecha de baja la fecha actual del sistema |
| eliminarObjetoOficioCentroPK(oficioCentroPK) | Eliminación lógica del objeto oficiocentro con oficioCentroPK "oficioCentroPK" introduciendo como fecha de baja la fecha actual del sistema |
| findOficiosPorCentro(anno) | Devuelve un listado de objetos oficiocentro asociados al año "anno" dados de alta en el momento de realizar la consulta |
| findAllOficiosPorCentro(anno) | Devuelve un listado de objetos oficioscentro asociados al año "anno" |
| findOficiosPorCentros(anno,textoBusqueda) | Devuelve una lista de objetos oficiocentro que estén asociados con el año "anno" o de descripción "textoBusqueda" |
| findOficiosPorCentros(anno,idCentro,idOficio) | Devuelve una lista de objetos oficioscentro que estén asociados con el año "anno" o el objeto centro con idCentro "idCentro" o asociados al objeto oficio con idOficio "idOficio" dados de alta en el momento de realizar la consulta |
| findListOficiosPorCentro(anno,idCentro) | Devuelve un listado de objetos oficio asociados al año "anno" y al objeto centro con idCentro "idCentro" dados de alta en el momento de realizar la consulta |
| findListCentrosPorOficio(anno,idOficio) | Devuelve un listado de objetos centro asociados al año "anno" y al objeto oficio con idOficio "idOficio" dados de alta en el momento de realizar la consulta |

Figura 59: Operaciones de la clase TOficioCentro

|  |  |
| --- | --- |
| TPrendaOficio | |
| Operación | **Descripción** |
| getPrendas(oficio) | Devuelve una lista de objetos prendaoficio asociadas al objeto oficio "oficio" que estén dadas de alta en el momento de realizar la consulta |
| getOficios(prenda) | Devuelve una lista de objetos prendaoficio asociadas al objeto prenda "prenda" que estén dadas de alta en el momento de realizar la consulta |
| getObjetoPrendasPorOficioId(idOficio,idPrenda) | Devuelve el objeto prendaoficio asociada al objeto oficio con idOficio "idOficio" y al objeto prenda con idPrenda "idPrenda" dada de alta en el momento de realizar la consulta |
| getObjetoPrendasPorOficio(oficio,prenda) | Devuelve el objeto prendaoficio asociada al objeto oficio "oficio" y al objeto prenda "prenda" dada de alta en el momento de realizar la consulta |
| guardarObjetoPrendasPorOficio(prendaofi) | Guarda el objeto predaoficio "prendaofi" en la BD |
| eliminarOficiosPorPrenda(prendaofi) | Eliminación lógica del objeto prendaoficio "prendaofi" introduciendo como fecha de baja la fecha actual del sistema |
| eliminarObjetoOficiosPorPrenda(oficio,prenda) | Eliminación lógica del objeto prendaoficio asociado al objeto oficio "oficio" y al objeto prenda "prenda" introduciendo como fecha de baja la fecha actual del sistema |
| eliminarObjetoOficioPrendaPK(oficioPrendaPK) | Eliminación lógica del objeto prendaoficio con prendaOficioPK "oficioPrendaPK" introduciendo como fecha de baja la fecha actual del sistema |
| findPrendasPorOficio(anno) | Devuelve un listado de objetos prendaoficio asociadas al año "anno" dadas de alta en el momento de realizar la consulta |
| findAllPrendasPorOficio(anno) | Devuelve un listado de objetos prendaoficio asociadas al año "anno" |
| findPrendasPorOficio(anno,textoBusqueda,idPrenda,idOficio); | Devuelve una lista de objetos prendaoficio que esté asociada con el año "anno" o de descripción "textoBusqueda" o asociada a la prenda con idPrenda "idPrenda" o asociada al oficio con idOficio "idOficio" que estén dadas de alta en el momento de realizar la consulta |
| findAllPrendasPorOficio(anno,textoBusqueda,idPrenda,idOficio); | Devuelve una lista de objetos prendaoficio que estén asociadas con el año "anno" o de descripción "textoBusqueda" o asociadas a la prenda con idPrenda "idPrenda" o asociadas al oficio con idOficio "idOficio" |
| findListPredasPorOficio(anno,idOficio) | Devuelve un listado de objetos prenda asociadass al año "anno" y al objeto oficio con idOficio "idOficio" dadas de alta en el momento de realizar la consulta |
| findListOficiosPorPrenda(anno,idPrenda) | Devuelve un listado de objetos oficio asociados al año "anno" y al objeto prenda con idPrenda "idPrenda" dados de alta en el momento de realizar la consulta |

Figura 60: Operaciones de la clase TPrendaOficio

|  |  |
| --- | --- |
| TGrupo | |
| Operación | **Descripción** |
| getGrupo(id) | Devuelve el objeto grupo asociado con idGrupo "id" |
| guardarGrupo(grupo) | Guarda el objeto grupo "grupo" en la BD |
| eliminarGrupo(id) | Eliminación lógica del objeto grupo asociado a la idGrupo "id" introduciendo como fecha de baja la fecha actual del sistema |
| findGrupos(anno) | Devuelve un listado de objetos grupo asociados al año "anno" que estén dados de alta en el momento de la consulta |
| findGrupos(anno,ids) | Devuelve un listado de todos los objetos grupo asociados al año "anno" y con idGrupo asociados al listado de "ids" |
| findGrupoDescripcion(anno,textoBusqueda) | Devuelve un objeto grupo asociado al año "anno" y la descripción "textoBusqueda" que esté dado de alta en el momento de realizar la consulta |

Figura 61: Operaciones de la clase TGrupo

|  |  |
| --- | --- |
| TEstados | |
| Operación | **Descripción** |
| getEstado(idEstado) | Devuelve el objeto estados asociado con idEstado "idEstado" |
| guardarEstado(estado) | Guarda el objeto estados "estado" en la BD |
| eliminarEstado(idEstado) | Eliminación física del objeto estados asociado a la idEstado "idEstado" |
| findEstado(anno,ids) | Devuelve un listado de objeto estados asociados al año "anno" que estén dados de alta en el momento de la consulta |
| findEstado(anno,tipoEstado,idCentro,idPrenda,idOficio) | Devuelve un listado de todos los objetos grupo asociados al año "anno" y tipoEstado "tipoEstado" o al objeto centro con idCentro "idCentro" o al objeto oficio con idOficio "idOficio" o al objeto prenda con idPrenda "idPrenda" que estén dados de alta en el momento de realizar la consulta |

Figura 62: Operaciones de la clase TEstados

1. Interfaz con el usuario

**Requisitos de diseño**

La interfaz ha sido diseñada siguiendo los estilos de la entidad corporativa. Su diseño ha seguido las pautas para ser una interfaz dinámica e interactiva, fácil e intuitiva. Se ha utilizado una serie de iconos que hacen más fácil el manejo de la aplicación.

**Resolución de la pantalla**

Para la realización de este proyecto se ha utilizado una resolución de 1200x800. Aunque como anteriormente se comentó en el apartado 3.1 Perspectiva del producto, su funcionamiento no se ve afectado si no se utiliza este tipo de resolución.

**Iconografía**

Como anteriormente se ha indicado, existen una serie de iconos descriptivos que hacen el manejo de esta aplicación más fácil e intuitivo. A continuación se pueden ver las diferentes iconografías con las que nos podemos encontrar:

perfil.png Usuario fecha.png Fecha calendario.png Calendarioeditar.png Editar

eliminar.png Eliminarinfo.pngManual de usuario (PDF) salir.png Salir de la aplicación

Codificación

1. Implementación

**Java**

Java es un lenguaje orientado a objetos y de plataforma neutra por lo que es posible ejecutarlo en diferentes ámbitos y sistemas operativos siempre que tengan un intérprete Java. Este lenguaje tuvo como objetivo ser más sencillo, así como no incluir ni punteros, ni permitir la herencia múltiple y ante todo que fuera capaz de ocuparse automáticamente de la asignación y des asignación de memoria.

**Struts2**

Struts2 [13] es un framework modelo vista controlador. Principalmente se caracteriza por separar las diferentes partes del programa y definir una navegación clara y unificada. Principales ventajas que han favorecido el desarrollo de este proyecto:

* Integración sencilla para Spring
* Añadido de plugins de forma fácil.
* Reporte de errores preciso
* Muy cómodo y sencillo la forma de testear los actions.
* Controladores fáciles de configurar.
* Acciones POJO.

**Spring**

Spring posibilita la generación de aplicaciones usando Plain Old Java Objets, POJO, objetos Java sin restricciones. POJO se dedica solo a la lógica mientras que el framework Spring provee al bean de lo necesario para permitir su funcionamiento y lograr que la integración del aplicativo sea efectiva. A continuación se indican algunos beneficios de Spring [14]:

* Spring está diseñado para que los objetos negocio no tengan dependencias con la API. Por lo tanto si se desea eliminar la gestión que realiza Sprins sobre un objeto se puede realizar sin problema.
* Facilita tener buenas prácticas de programación. Además as aplicaciones son fáciles de testear.
* Organiza objetos de negocio.

**Hibernate**

Hibernate es un mapeador de objeto relacional cuyo objetivo principal es mapear los modelos relacionales con modelos orientados a objetos. Ventajas de usar Hibernate [15]:

* Persistencia de Datos: Hibernate permite definir de forma sencilla las clases persistentes que representan a Entidades, y permite una gestión flexible de las Entidades asociadas.
* Independencia BBDD: Puesto que no se tiene por que usar SQL en el código, siendo este sustituido por la propia API de Hibernate, se permite el no tener que modificar nada en el código por un cambio de tecnología de base de datos.
* Coste bajo de desarrollo: Tras la curva de aprendizaje inicial de Hibernate, es mayor el beneficio obtenido por la rapidez de desarrollo (construcción de consultas, modificación de datos) que el coste de aprendizaje.
* Reducción de errores de programación: Haciendo uso de las API de Hibernate se reducen las probabilidades de errores a la hora de realizar consultas.

**AXIS2**

Para la realización de este proyecto se ha utilizado para lograr que la aplicación de Gestión de la Configuración cargue el XML con el menú según el perfil del usuario de forma automática.

Sobre Axis se puede decir que es un proyecto basado en Java donde se implementa el cliente y un servidor de servicios web que permite mejorar la portabilidad. [16]

**JDBC**

Para el este proyecto se ha utilizado como base de datos Oracle.

Los programas en Java se comunican con las bases de datos y manipulan los datos mediante la API JDBC.

La Java Database Connectivity es independiente del sistema operativo sobre el que se ejecute el programa o la base de datos que se esté utilizando. Una clase llamada driver actúa como puente con la BD, y cada BD tiene su propio driver.

**Oracle**

Sistema de gestión de base de datos. Se ha utilizado este tipo de base de datos ya que ofrece combinación de tecnologías de alto nivel y soluciones de negocio integradas. El rendimiento garantiza la integridad de los datos. Cumple con las cuatro propiedades que deben tener las base de datos: atomicidad, coherencia, aislamiento y durabilidad.

**Eclipse**

Entorno de desarrollo multiplataforma y de sofware libre. Se ha utilizado esta plataforma ya que cuenta que un navegador de clases y un depurador que ha sido muy útil a la hora de desarrollar la aplicación.

**Subversion**

Es una herramienta de control de versiones open source basada en un repositorio [17]. Es software bajo licencia de tipo Apache/BSD

1. Otros programas

**Redmine**

Gestor y planificador de proyecto con interfaz web, orientado a la coordinación de tareas y comunicación de participantes. [18]

**DIA**

Herramienta para la creación de diagramas. Disponible bajo licencia GPL. Utilizada para los diagramas de caso de uso y diagramas de secuencia de sistema de este mismo capítulo.

**Gantt Project**

Herramienta gratuita para la gestión de proyecto necesaria para la realización del diagrama de Gantt del capítulo 2.

Pruebas y validación

Para el éxito de este proyecto la realización continua de pruebas ha sido fundamental. Adicionalmente se incluyó un periodo de pruebas al final del mismo. Así, cuando finaliza cada iteración se produce una fase de prueba y error donde se prueba el prototipo final del PFC de Gestión de Vestuario. Se comprueba que se cumplen los requisitos establecidos correctamente y se realizan las subsanaciones .

Tras la realización de prueba y error y haber subsanado todos aquellos errores que aparecen se revisa todo para la mejora de ciertos aspectos del desarrollo ayudando a que el sistema sea más rápido y reduciendo tiempos de consultas y esperas.

Durante la fase de prueba se han utilizado las técnicas de **caja** **negra** y **caja** **blanca**.

* Las pruebas de caja blanca se centran en la implementación del programa para selección de casos de prueba. Es lo que llamamos enfoque estructural.
* Las pruebas de caja negra se centran en lo que se espera de cada parte. Se suministran datos de entrada y se estudia la salida sin preocupar que hace por dentro el módulo. Este tipo de pruebas se apoya en la especificación de requisitos. A diferencia de la caja blanca, este se conoce como enfoque funcional.

En la realización de este proyecto se han utilizado los dos tipos de pruebas aunque en mayor medida han sido de tipo caja negra. Para la eliminación lógica de elementos se han realizado pruebas de caja blanca para ver el comportamiento de baja de las asociaciones.

1. Entorno de pruebas

A continuación se muestran una serie de tablas donde se puede observar para los diferentes escenarios del sistemas los resultados que se esperan.

|  |  |
| --- | --- |
| Caso de uso listar grupo (listar/editar/nuevo/eliminar) | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se cambia el año seleccionado | Mensaje: "Se ha cargado el año seleccionado correctamente"  Aparece la lista con los objetos de ese año |
| Se recupera el listado correctamente | Se muestra el listado completo de grupos |
| Se selecciona un grupo del listado para editar | Se muestra el formulario con los datos cargados del grupo |
| Se selecciona "nuevo" | Se muestra el formulario vacío |
| Se elimina un elemento de la lista | Se muestra la lista con la nueva fecha de baja. Se añade la fecha de baja a las prendas asociadas al grupo. Se añade fecha de baja a las asociaciones prendaoficio de cada una de las prendas asociadas a ese grupo. En ambos casos siempre que no tuviera fecha de baja y si la tuviera fuera posterior. La fecha de baja es la fecha del sistema |
| No se introduce la descripción | Error: "El campo descripción es un campo obligatorio" |
| Se introduce descripción superior a 255 | Error: "El campo descripción no puede tener más de 255 caracteres" |
| Existe un objeto en la base de datos con la misma descripción | Error:"Existe un centro con esa descripción" |
| No se introduce el IVA | Error: "El campo IVA es obligatorio" |
| Se introduce fecha de baja en el campo para ese fin al objeto que se edita. | Se añade la fecha de baja a las prendas asociadas al grupo. Se añade fecha de baja a las asociaciones prendaoficio de cada una de las prendas asociadas a ese grupo. En ambos casos siempre que no tuviera fecha de baja y si la tuviera fuera posterior. |
| Se produce un error guardando el grupo | Error: "El grupo no se ha podido guardar" |
| Se guarda el grupo en la base de datos | Mensaje: "El grupo se ha guardado correctamente" |
| Se guarda el grupo correctamente | Se muestra el listado con todos los grupos |

Figura 63: Pruebas caso de uso listar grupo

|  |  |
| --- | --- |
| Caso de uso listar prenda (listar/editar/nuevo/eliminar) | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se cambia el año seleccionado | Mensaje: "Se ha cargado el año seleccionado correctamente"  Aparece la lista con los objetos de ese año |
| Se recupera el listado correctamente | Se muestra el listado completo de prendas |
| Se selecciona una prenda del listado para editar | Se muestra el formulario con los datos cargados de la prenda |
| Se selecciona "nuevo" | Se muestra el formulario vacío |
| Se elimina un elemento de la lista | Se muestra la lista con la nueva fecha de baja. Se comprueba si el grupo tiene fecha de baja anterior, si la tiene conserva la del grupo. Se dan de baja todas las asociaciones de prendaoficio siempre que no tuviera fecha de baja y si la tuviera fuera posterior. La fecha de baja es la fecha del sistema |
| No se introduce la descripción | Error: "El campo descripción es un campo obligatorio" |
| Se introduce descripción superior a 255 | Error: "El campo descripción no puede tener más de 255 caracteres" |
| Existe un objeto en la base de datos con la misma descripción | Error: "Existe una prenda con esa descripción" |
| No se introduce el grupo | Error: "El campo grupo es obligatorio" |
| Se introduce fecha de baja en el campo para ese fin al objeto que se edita. | Se comprueba si el grupo tiene fecha de baja anterior, si la tiene conserva la del grupo. Se dan de baja todas las asociaciones de prendaoficio siempre que no tuviera fecha de baja y si la tuviera fuera posterior. |
| Se produce un error guardando la prenda | Error: "La prenda no se ha podido guardar" |
| Se guarda la prenda en la base de datos | Mensaje: "La prenda se ha guardado correctamente" |
| Se guarda la prenda correctamente | Se muestra el listado con todas las prendas |

Figura 64: Pruebas caso de uso listar prenda

|  |  |
| --- | --- |
| Caso de uso listar oficio (listar/editar/nuevo/eliminar) | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se cambia el año seleccionado | Mensaje: "Se ha cargado el año seleccionado correctamente"  Aparece la lista con los objetos de ese año |
| Se recupera el listado correctamente | Se muestra el listado completo de oficios |
| Se selecciona un oficio del listado para editar | Se muestra el formulario con los datos cargados del oficio. Se muestra la tabla con las prendas asociadas a ese oficio. |
| Se selecciona "nuevo" | Se muestra el formulario vacío |
| Se elimina un elemento de la lista | Se muestra la lista con la nueva fecha de baja. Se da de baja todas prendaoficio asociadas a ese oficio siempre y cuando la prenda asociada a esa prendaoficio no tuviera fecha de baja y si la tuviera fuera posterior. Se da de baja todas las asociaciones oficiocentro de ese oficio siempre y cuando el centro asociado a ese oficio centro no tuviera fecha de baja y si lo tuviera fuera posterior. La fecha de baja es la fecha del sistema |
| No se introduce la descripción | Error: "El campo descripción es un campo obligatorio" |
| Se introduce descripción superior a 255 | Error: "El campo descripción no puede tener más de 255 caracteres" |
| Existe un objeto en la base de datos con la misma descripción | Error: "Existe una prenda con esa descripción" |
| No se introducen asociaciones con prendas | Aviso: "Va a guardar un oficio sin prendas asociadas ¿desea continuar?" |
| Limpiar formulario de prendaoficio | El formulario queda en blanco |
| Editar asociación prendaoficio | El formulario se carga con los datos de prendaoficio |
| Se añade asociación prendaoficio sin periodicidad, prenda o cantidad | Error: "Es necesario indicar la prenda, cantidad y periodo si desea añadir" |
| Se añade asociación ya existente | Error: "Ya existe la asociación de la prenda con el oficio" |
| Se borra asociación prendaoficio | Mensaje: "La prenda se ha dado de baja correctamente en el oficio"  Se comprueba que la fecha de baja del oficio no es anterior ni la fecha de baja de la prenda. En ese caso conserva la más reciente. |
| Se introduce fecha de baja en el campo para ese fin al objeto que se edita. | Se da de baja todas prendaoficio asociadas a ese oficio siempre y cuando la prenda asociada a esa prendaoficio no tuviera fecha de baja y si la tuviera fuera posterior. Se da de baja todas las asociaciones oficiocentro de ese oficio siempre y cuando el centro asociado a ese oficio centro no tuviera fecha de baja y si lo tuviera fuera posterior. |
| Se produce un error guardando el oficio | Error: "El oficio no se ha podido guardar" |
| Se guarda el oficio en la base de datos | Mensaje: "El oficio se ha guardado correctamente" |
| Se guarda el oficio correctamente | Se muestra el listado con todos los oficios |

Figura 65: Pruebas caso de uso listar oficio

|  |  |
| --- | --- |
| Caso de uso listar centro (listar/editar/nuevo/eliminar) | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se cambia el año seleccionado | Mensaje: "Se ha cargado el año seleccionado correctamente"  Aparece la lista con los objetos de ese año |
| Se recupera el listado correctamente | Se muestra el listado completo de cetros |
| Se selecciona un centro del listado para editar | Se muestra el formulario con los datos cargados del centro. Se muestra la tabla con los oficios asociados a ese centro |
| Se selecciona "nuevo" | Se muestra el formulario vacío |
| Se elimina un elemento de la lista | Se muestra la lista con la nueva fecha de baja. Se da de baja todos oficiocentro asociados a ese oficio siempre y cuando no tuviera y si lo tuviera fuera posterior y la fecha del oficio asociado a ese oficio centro no tuviera fecha de baja o si la tuviera fuera posterior. La fecha de baja es la fecha del sistema |
| No se introduce la descripción | Error: "El campo descripción es un campo obligatorio" |
| Se introduce descripción superior a 255 | Error: "El campo descripción no puede tener más de 255 caracteres" |
| Existe un objeto en la base de datos con la misma descripción | Error: "Existe una prenda con esa descripción" |
| No se introducen asociaciones con oficios | Aviso: "Va a guardar un centro sin prendas asociadas, ¿desea continuar?" |
| Limpiar formulario de oficio centro | El formulario queda en blanco |
| Editar asociación oficiocentro | El formulario se carga con los datos de oficiocentro |
| Se añade asociación oficio centro sin número de personas ni oficio | Error: "Es necesario indicar el oficio y el número de personas que desea añadir |
| Se añade asociación ya existente | Error: "Ya existe la asociación del oficio con el centro" |
| Se borra asociación oficiocentro | Mensaje: "El oficio se ha dado de baja correctamente en el centro"  Se comprueba que la fecha de baja del oficio no es anterior ni la fecha de baja del centro. En ese caso conserva la más reciente. |
| Se introduce fecha de baja en el campo para ese fin al objeto que se edita. | Se da de baja todos oficiocentro asociados a ese oficio siempre y cuando no tuviera y si lo tuviera fuera posterior y la fecha del oficio asociado a ese oficio centro no tuviera fecha de baja o si la tuviera fuera posterior. |
| Se produce un error guardando el centro | Error: "El centro no se ha podido guardar" |
| Se guarda el centro en la base de datos | Mensaje: "El centro se ha guardado correctamente" |
| Se guarda el centro correctamente | Se muestra el listado con todos los centros |

Figura 66: Pruebas de caso de uso listar centros

|  |  |
| --- | --- |
| Caso de uso generar necesidades | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se selecciona el año y no existen centros para ese año | Error: "No hay centros para generar necesidades"  Aparece la pantalla para volver a seleccionar el año |
| Se generan necesidades | Se realiza el cálculo correctamente |
| Se generan las necesidades | Mensaje: "Necesidades generadas correctamente"  Aparece el listado con las necesidades generadas |
| Existen necesidades generadas para ese año | Se eliminan las necesidades anteriores y se sobrescriben con las nuevas |
| Se produce un error cuando se generan las necesidades | Error: "No se han generado las necesidades correctamente" |

Figura 67: Pruebas de caso de uso generar necesidades

|  |  |
| --- | --- |
| Caso de uso consulta necesidades | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se selecciona el año y no existen necesidades | Error: "No hay necesidades generadas para ese ejercicio"  Aparece la pantalla para volver a seleccionar el año |
| Se selecciona un centro para ver sus necesidades | Aparece el listado para ese centro |
| Se selecciona un grupo para ver sus necesidades | Aparece el listado para ese grupo |
| Se selecciona un centro y un grupo para ver sus necesidades | Aparece el listado para ese centro y grupo |
| No se selecciona ni centro ni grupo | Aparece el listado de todas las necesidades |
| Existen necesidades generadas para ese año | Se eliminan las necesidades anteriores y se sobrescriben con las nuevas |

Figura 68: Pruebas del caso de uso consulta de necesidades

|  |  |
| --- | --- |
| Caso de uso alta actuación | |
| Descripción | **Salida** |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| La actuación ya fue dada de alta con anterioridad | Error: "La actuación ya fue dada de alta. Contacte con contratación"  Aparece la pantalla para volver a seleccionar el año |
| La actuación no fue dado de alta y no existen necesidades | Error: No se han generado necesidades para ese ejercicio |
| La actuación no fue dada de alta y existen necesidades | Se realizan las inserciones en la base de datos de Contrat@  Se bloquea el alta de actuaciones para ese ejercicio  Mensaje: "Se ha procesado el alta de actuación correctamente" |
| Se produce un error al procesar el alta | Error: "No se ha procesado el alta de actuación" |

Figura 69: Pruebas caso de uso alta de actuación

|  |  |
| --- | --- |
| Caso de uso deshacer alta de actuación | |
| Descripción | Salida |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| No existen necesidades para ese ejercicio | Error: "No se han dado de alta necesidades para ese ejercicio"  Vuelve a la pantalla de generar necesidades |
| Se ha procesado la activación del alta de actuación correctamente | Aviso: "Puede volver a realizar el alta de actuación. Comuníqueselo a Contratación" |
| Se produce un error al activar el alta | Error: "No se ha podido realizar la acción" |

Figura 70: Pruebas de caso de uso deshacer alta actuación

|  |  |
| --- | --- |
| Caso de uso generar tallaje | |
| Descripción | **Salida** |
| Se pincha en el punto de menú generar tallaje | Se guarda en sesión consultaTallaje a false |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| No hay tallaje generados para el año seleccionado y tampoco necesidades | Error: "No se han generado necesidades para ese ejercicio" |
| No hay tallaje pero si necesidades generadas para ese ejercicio | Se muestra el listado con las necesidades y en cantidad real aparece la cantidad necesaria |
| Se han generado tallaje con anterioridad | Se muestra el listado con el tallaje y el formulario relleno con los campos antiguos |
| Se guarda el tallaje | Mensaje: "Se ha generado el tallaje correctamente"  Se muestra la consulta del tallaje |
| Se produce un error al guardar el tallaje | Error: "No se ha podido generar el tallaje" |

Figura 71: Pruebas de caso de uso generar tallaje

|  |  |
| --- | --- |
| Caso de uso consulta tallaje | |
| Descripción | **Salida** |
| Se pincha en el punto de menú consultar tallaje | Se guarda en sesión consultaTallaje a true |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| Se selecciona el año y no existe tallaje | Error: "No hay tallaje generado para ese ejercicio"  Aparece la pantalla para volver a seleccionar el año |

Figura 72: Pruebas de caso de uso consulta tallaje

|  |  |
| --- | --- |
| Caso de uso generar precio prendas | |
| Descripción | Salida |
| No se ha cargado el año del sistema correctamente | Error: "Error estableciendo año" |
| No existen prendas para el ejercicio | Error: "No existen prendas para el ejercicio" |
| Existen prendas asociadas al año | Se muestra el listado con las prendas y el formulario relleno con los campos precio de cada prenda |
| Se guarda la prenda | Mensaje: "Se han guardado las prendas correctamente"  Se muestra el listado con la consulta del precio de prendas. |
| Se produce un error al guardar la prenda | Error: "No se ha podido guardar el precio de las prendas" |

Figura 73: Pruebas de caso de uso generar precio prendas

|  |  |
| --- | --- |
| Otras pruebas realizadas | |
| Descripción | **Salida** |
| Se crea un nuevo objeto | Se crea auditoria de creación con DNI de usuario y fecha del sistema  Se crea la auditoria de modificación con DNI del usuario y fecha del sistema |
| Se modifica un objeto en cualquier momento | Se edita la auditoria de modificación con DNI del usuario y fecha del sistema |
| Se pulsa volver | No se guardan en sesión datos innecesarios |
| Se crea o modifica un objeto | Se crea o modifica en la base de datos |
| Se elimina un objeto | Se elimina de la base de datos |
| Se da de baja un objeto | Se dan de baja los objetos asociados siempre y cuando no tuvieran fecha de baja o fuera posterior. |

Figura 74: Otras pruebas realizadas

Capítulo 5: Conclusiones

En este proyecto se ha desarrollado una nueva aplicación para la gestión del vestuario en la Diputación de Cádiz que cubre las necesidades que existían hasta ahora en la gestión de vestuario. Se caracteriza por su interfaz dinámica e interactiva que se adapta a esas necesidades volviéndola más útil, y que, además, se integra con la plataforma corporativa.

Con este proyecto se ha incluido la gestión del vestuario en el ámbito de la administración electrónica para la agilización de todas las partes del proceso que lo forman. Para ello el proyecto está integrado con diferentes aplicativos desarrollados por EPICSA.

1. Calendario

El proyecto se ha realizado en un periodo de 9 meses, durante las prácticas realizadas en EPICSA. Principalmente el proyecto se iba a desarrollar en el transcurso de la misma pero debido a la necesidad de trabajar en otros proyectos parte de la aplicación se ha tenido que trabajar fuera de ese horario.

En la estimación de días ideales y reales se puede observar un retraso en cada una de ellas. Esto es debido a que a lo largo de la realización del proyecto se han tenido que realizar múltiples cambios que han aumentado el tiempo del mismo. Considerando la fase de mayor esfuerzo, correspondiente a la iteración 4º la gestión del administrador.

Al optar por un modelo de ciclo de vida incremental se ha podido dividir en las iteraciones adecuadas para una correcta planificación y respetar los plazos.

La metodología XP ha resultado ser muy adecuada para el desarrollo del proyecto ya que como se ha realizado a lo largo de unas prácticas en empresa, ha sido posible la resolución de dudas in situ.

En el modelado de base de datos de tipo relacional, aun cuando constituye la mejor opción, se ha invertido un tiempo adicional, debido al reajuste del diseño provocado por contratiempos que ha surgido a lo largo del desarrollo. Este tiempo adicional ha permitido desarrollar una base de datos consistente que se ajusta a las necesidades del producto.

1. Interfaz

En cuanto a la interfaz, por una parte se ha mejorado sustancialmente el efecto visual de la aplicación, se ha optado por una gama de colores que está en sintonía con los colores corporativos de la empresa para la que va dirigido este proyecto. Por otra parte se ha dado uniformidad al conjunto, gracias a ello y a una serie de botones descriptivos se ha conseguido facilitar la navegación por la aplicación a cualquier tipo de usuario, desde el más neófito hasta el más experimentado en la materia

En cuanto a la interfaz de usuario, se ha construido una interfaz fácil de asimilar y entender, con botones descriptivos que identifican las acciones que ejecutan. Además cuenta con un menú donde se recogen todas las posibles funcionalidades del sistema de forma que el acceso a cualquiera de ellas es rápido.

1. Futuras ampliaciones

Está previsto ampliar el proyecto añadiéndole un nuevo requisito funcional. Este va a consistir en poder realizar envío de informes a Port@firmas para su firma electrónica. Un requisito indispensable para ello es que el documento sea formato PDF.

1. Valoraciones personales

Una vez hecho balance de lo que ha supuesto el trabajo del proyecto de Gestión de Vestuario la sensación es muy positiva.

Aplicando los conocimientos que he podido adquirir a lo largo de mi carrera, este proyecto me ha ayudado a afianzar, aplicar y ampliar esas herramientas.

La elaboración desde cero del aplicativo me ha ayudado a conocer más de cerca su proceso de diseño y desarrollo. Parte o totalidad de las horas empleadas me han ayudado a mejorar las técnicas de programación conocidas y a estudiar con profundidad una de las tecnologías actualmente más usada, Java.

Me siento muy orgullosa del trabajo y esfuerzo que he realizado ya que pienso que mejorará de forma considerable la gestión del vestuario.

Bibliografía y Referencias electrónicas

Referencias

[1] http://www.java.com/es/

Página oficial de java.

[2] http://www.josso.org/

Web oficial de Josso.

[3] http://www.openldap.org/

LDAP - Lightweight Directory Access Protocol.

[4] http://losanimaleskate.blogspot.com.es/2012/02/ciclos-de-vida-del-software.html

Blog donde se ha extraído la imagen.

[5]http://ciclodevidasoftware.wikispaces.com/CICLO+DE+VIDA+INCREMENTAL

Web con información sobre modelos de ciclo de vida.

[6] http://ganttproject.sourceforge.net

Web de descarga de software libre para la realización de diagrama de Gantt.

[7] http://plandecarrera.infojobs.net/

Web Infojobs.

[8] http://www.dipucadiz.es/portal\_del\_personal\_empleado/

Acceso corporativo de los empleados de la Diputación de Cádiz.

[9] https://ws024.juntadeandalucia.es/ae/adminelec/areatecnica/portafirmas

Web oficial de la Consejería de Hacienda y Administración Pública de la Junta de Andalucía.

[9.1] http://xprogramming.com/index.php

Web de metodología de programación tipo XP, eXtreme Programming.

[10] http://live.gnome.org/Dia

Web oficial de aplicación Día.

[11] E. Bascón Pantoja. El patrón de diseño modelo-vista-controlador (MVC) y su implementación en java swing. Revista Acta Nova, 2(4), 2011.

[12] http://www.monografias.com/trabajos89/poo-y-mvc-php/poo-y-mvc-php2.shtml

Web de consulta de modelo vista controlador.

[13] http://struts.apache.org/development/2.x/

Sitio oficial del framework Struts 2.

[14] www.springframework.com

Sitio oficial del framework Spring.

[15] http://hibernate.org/

Sitio web oficial del framework Hibernate.

[16] http://axis.apache.org/axis2/java/core/

Sitio web oficial de Axis2.

[17] http://www.eclipse.org/subversive/

Subversion Eclipse.

[18] http://www.redmine.org/]

Web oficial Redmine.

[A]http://www.estudioteca.net/

Proceso licitatorio.

[B] http://www.gnu.org/copyleft/fdl.html

Web de descarga de documentación de licencia GNU Free documentation.

[C] https://contrataciondelestado.es/

Plataforma de Contratación del Sector Público.

Otras Webs de consulta

[19] http://stackoverflow.com/

Web de consultas y respuestas a dudas sobre programación.

[20]http://www.chuidiang.com/

Web de programación extrema.

[21]http://www.lawebdelprogramador.com

Web diseñada para programadores aficionados y profesionales.

[22]https://www.pdftoword.com/es/

Web para pasar PDF a formato de texto.

[23]http://buscon.rae.es/diccionario/drae.htm

Web diccionario real academia española.

[24]http://www.uv.es/grimo/teaching/SpringMVCv4PasoAPaso/part3.html

Curso de MVC.

[25]https://ws024.juntadeandalucia.es/ae/adminelec/areatecnica/portafirma

Web de consulta de la Conserjería de Haciendo y Administración Pública: Port@firma.

Bibliografía

[Booc00] Booch, G. & Jacobson, I. & Rumbaugh, J.M. UML. El lenguaje unificado de modelado. Manual de Referencia.

Addison Wesley, 1999.

[CELMA03]Celma, M.; Casamayor, J.C.; Mota, L.

Bases de Datos Relacionales

Pearson-Prentice Hall, 2003.

[DATE01]Date, C.J.

Introducción a los sistemas de bases de datos.

Addison-Wesley Publishing Company, Ed. 7, 2001.

[ELMASRI02]Elmasri & Navathe

Fundamentos de Sistemas de Bases de Datos.

Addison-Wesley Publishing Company, Ed. 3, 2002 .

[DRAE]

Real Academia Española de la Lengua

Diccionario de la Lengua Española

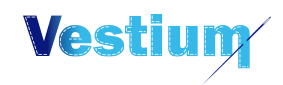
Espasa, 2001 .

R. Pressman, Ingeniería del Software. Un enfoque práctico, McGraw Hill,

(2002).

Anexo A: Manual de usuario

A continuación se incluye el manual de usuario siguiendo la plantilla utilizada en EPICSA para documentos con este fin.

****

**Vestium, gestión de vestuario de la**

**Diputación de Cádiz**

**Manual de usuario/a**

**Fecha:** 30 de mayo de 2017

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de EPICSA, S.A.

ÍNDICE

[1 INTRODUCCIÓN 3](#1.INTRODUCCIÓN|outline)

[1.1 Alcance 3](#1.1. Alcance|outline)

[1.2 Visión general y arquitectura 3](#1.2. Visión general y arquitectura|outline)

[2 PUESTA EN MARCHA Y OPERATIVA 6](#2. PUESTA EN MARCHA Y OPERATIVA |outline)

[2.1 Requerimientos mínimos de hardware y software 6](#2.1. Requerimientos mínimos de hardware y software|outline)

[2.1.1 Servidores 6](#2.1.1. Servidores|outline)

[2.1.2 Estaciones cliente 6](#2.1.2. Estaciones cliente|outline)

[2.2 Acciones previas: instalación y configuración de la solución 7](#2.2. Acciones previas: instalación y configuración de la solución|outline)

[2.3 Procedimiento de digitalización de documentos 11](#2.3. Procedimiento de digitalización de documentos|outline)

[2.4 Búsqueda de documentos 15](#2.4. Búsqueda de documentos|outline)

[2.5 Revisión de documentos pendientes 17](#2.5. Revisión de documentos pendientes|outline)

[2.6 Gestión de metadatos 18](#2.6. Gestión de metadatos|outline)

[2.6.1 Definición de constraints del modelo documental 19](#2.6.1. Definición de constraints del modelo documental|outline)

[2.6.2 Definición de tipos documentales 21](#2.6.2. Definición de tipos documentales|outline)

[2.6.2.1 Gestión de metadatos 22](#2.6.2.1. Gestión de metadatos|outline)

[2.6.2.2 Gestión de estructura de carpetas del tipo documental 24](#2.6.2.2. Gestión de estructura de carpetas del tipo documental|outline)

[2.6.2.3 Gestión de campos de formulario del tipo documental 25](#2.6.2.3. Gestión de campos de formulario del tipo documental|outline)

[3 ACCIONES DEL MANTENIMIENTO PREVENTIVO 27](#3. ACCIONES DEL MANTENIMIENTO PREVENTIVO|outline)

[3.1 Operaciones sobre la aplicación 27](#3.1. Operaciones sobre la aplicación|outline)

[3.2 Operaciones sobre periféricos 27](#3.2. Operaciones sobre periféricos|outline)

[3.3 Entorno de trabajo 28](#3.3. Entorno de trabajo|outline)

[3.4 Cuadro resumen de las acciones y periodicidad del mantenimiento 29](#3.4. Cuadro resumen de las acciones y periodicidad del mantenimiento|outline)

[4 MANTENIMIENTO CORRECTIVO 31](#4. MANTENIMIENTO CORRECTIVO|outline)

[4.1 Contingencias durante el proceso de digitalización 31](#4.1. Contingencias durante el proceso de digitalización|outline)

[4.2 Contingencias durante el procedimiento de archivo 31](#4.2. Contingencias durante el procedimiento de archivo|outline)

1. INTRODUCCIÓN
   1. Alcance

Versión: Vestium v1.0

Fecha: 30 de mayo de 2017

Se pretende informatizar la totalidad del proceso de gestión de vestuario en la medida de lo posible, para ello la aplicación permite realizar el mantenimiento de Centros, Oficios, Prendas y grupos pertenecientes a la Diputación de Cádiz, así como gestionar las necesidades para cada uno de ellos y dar de alta la actuación en el aplicativo Contrat@ para dar comienzo así el proceso licitatorio.

Figura 1: Pantalla principal

* 1. Visión general y arquitectura

Vestium es una aplicación desarrollada con tecnología web. En líneas generales la gestión del vestuario consiste principalmente en, una vez realizadas las negociaciones entre los Sindicatos y la Diputación, y definido el catálogo de prendas para el año en cuestión, se realiza el mantenimiento de los objetos que componen dicho catálogo en la aplicación. Tras el mantenimiento se generan las necesidades para los diferentes oficios que forman parte de los centros pertenecientes a la Diputación. Una vez definidas estas necesidades, se da de alta una actuación en el aplicativo de Contrat@ para que dé comienzo el proceso licitatorio.

Vestium está dividido en 3 módulos:

* **Módulo de gestión de necesidades .**
* Realiza la tarea de generar las necesidades para el ejercicio.
* Permite realizar consultas de las anteriores necesidades generadas
* Da de alta en Contrat@ la actuación con las necesidades del suministro.
* Permite realizar extracción en un documento Excel del listado de las necesidades para el ejercicio.
* **Módulo de tallaje.**
* Realiza tareas de generación de tallaje.
* Obtiene un listado de los tallajes para el ejercicio en cuestión además permite realizar una extracción en documento Excel de dicho listado.
* Permite indicar para cada prenda el valor que tiene para permitir realizar el presupuesto estimado del año próximo.
* **Módulo mantenimiento.**
* Se realiza el mantenimiento de Centros, Oficios, Prendas y Grupos. Así como las asociaciones que existan entre ellos.
* Permite al usuario realizar la copia de todos los objetos anteriormente nombrados para la preparación del vestuario del año próximo.
* Se obtiene el listado con el catálogo de las prendas que se ha establecido entre la Diputación y los sindicatos.
* Permite realizar exportaciones a Excel de los listados de Centros, Oficios, Prendas, Grupos y catálogo.

1. PUESTA EN MARCHA Y OPERATIVA
   1. Requerimientos mínimos de hardware y software
      1. Servidores

|  |  |
| --- | --- |
| **Dato** | **Valor** |
| **SO** | Windows, Linux |
| **Kernel** | - |
| **Memoria RAM** | 1GB o más |
| **Tamaño Almacenamiento** | 60MB necesarios (sin contar logs generados posteriormente) |
| **Otros** | - Servidor de aplicaciones Apache Tomcat 5.0.28 o superior |

* + 1. Estaciones cliente

|  |  |
| --- | --- |
| **Dato** | **Valor** |
| **SO** | Versiones Windows |
| **Procesador** | - |
| **Memoria RAM** | 1GB o más |
| **Tamaño Almacenamiento** | - |
| **Otros** | - Navegador web: Internet Explorer 6.0 o Mozilla Firefox 2.0.20 (o superiores)  - Versión java: JRE 1.6.0\_45 o superior |

|  |  |
| --- | --- |
| **Dato** | **Valor** |
| **SO** | Distribuciones Linux |
| **Procesador** | - |
| **Memoria RAM** | 1GB o más |
| **Tamaño Almacenamiento** | - |
| **Otros** | - Navegador web: Mozilla Firefox v3.6.10  - Visor de ficheros PDF con capacidad para consultar datos de firma y propiedades del documento.  - Versión Java: JRE 1.6.0 |

* 1. Acciones previas: instalación y configuración de la solución

**Configuración en servidor tras el despliegue**

El administrador de sistemas de la entidad adquiriente deberá configurar los siguientes ficheros de propiedades de la aplicación para adaptarlos al entorno de trabajo donde se despliega la aplicación:

|  |  |
| --- | --- |
| **Fichero de Configuración** | **Detalle de la modificación, el alta o la baja** |
| /WEB-INF/classes/  resources.properties | VESTIUM  vestium.url = http://localhost:8080/vestuario/  web.dipucadiz = http://www.dipucadiz.es  JOSSO Producción  josso.host=http://172.22.8.196:6060  josso.url=https://www3.dipucadiz.es/josso/signon/login.do  josso.logout.url=https://www3.dipucadiz.es/josso/signon/logout.do  url.entrada.general=https://www3.dipucadiz.es/imdica/  PROVIDER\_URL=ldap://172.22.8.173:389/dc=dipucadiz,dc=es  INITIAL\_CONTEXT\_FACTORY = com.sun.jndi.ldap.LdapCtxFactory  SECURITY\_AUTHENTICATION = simple  SECURITY\_PRINCIPAL = cn=Manager,dc=dipucadiz,dc=es  SECURITY\_CREDENTIALS = secret  SECURITY\_PROTOCOL = ssl  ORACLE Contratación  oracle.url = jdbc:oracle:thin:@172.22.8.243:1521:OLIBRIOD2  oracle.user = CONTRATA  oracle.pass = CONTRATA |
| /WEB-INF/classes/log4j.properties | Los valores del mismo son los que vienen a continuación. Es necesario revisar la profundidad de la traza, recomendable ALL en pruebas e INFO en explotación, y que la ruta del archivo log en el servidor de aplicaciones sea correcta.  log4j.rootLogger=DEBUG, vestium  log4j.logger.org.apache=ERROR  log4j.logger.org.displaytag=ERROR  log4j.appender.vestium=org.apache.log4j.DailyRollingFileAppender  log4j.appender.vestium.File=/Logs/Vestium.log  log4j.appender.vestium.layout=org.apache.log4j.PatternLayout  log4j.appender.vestium.DatePattern='.'yyyy-MM-dd  log4j.appender.vestium.layout.ConversionPattern=%d{[yyyy-MM-dd HH:mm:ss,SSS]} %5p (%F:%L) - %m%n# |
| /WEB-INF/classes/ authenticator.properties | *authenticator.impl=josso*  authenticator.impl.josso=es.guadaltel.framework.authenticator.impl.JossoAuth  enticatorImpl  authenticator.josso.endpoint.url=<url\_gjosso>/gjosso/signon  authenticator.josso.endpoint.ws=<url\_gjosso>/gjosso  authenticator.josso.session.attribute.name=JOSSO\_SESSIONID\_josso  authenticator.josso.requesterid=Rode  authenticator.josso.fixed.backto.url=<url\_rode>/rode  #Establece si se activa el sistema de login de usuarios (true o false)  authenticator.josso.filter.active=false  #Nombre del usuario conectado cuando esta desactivada la autenticación  login.disabled.user=DEFAULT  #Mensaje indicador de que la autenticación no esta activa (No obligatorio)  login.disabled.message=  #Mensaje de aviso de que la autenticación no esta activa (No obligatorio) |

* 1. Mantenimiento de vestuario

En este apartado se realiza el mantenimiento de todos los componentes que forman parte el catálogo estipulado previamente por los Sindicatos y la Central de Contratación. A continuación se explica cómo realizar ese mantenimiento.

* + 1. Creación, edición y eliminación de elementos de vestuario.

Para un correcto mantenimiento de los elemento se debe seguir el siguiente orden.

1. **GRUPO:** Una serie de prendas que reúnen unas características comunes es lo que se llamará a partir de ahora un grupo. Por ejemplo, el grupo calzado lo forman las prendas zapato ejecutivo, zapato seguridad, zanco blanco, etc.
2. **PRENDA**: Cada una de las partes que son necesarias para formar un uniforme. Por ejemplo, camisa blanca, pantalón negro, calcetines grises y zapatos ejecutivo.
3. **OFICIO**. Un oficio es una ocupación a la que se le va a suministrar un vestuario específico. Por ejemplo, entre los oficios se incluyen: auxiliar de clínica, conductor/a, jardinero/a, cocinero/a, profesor/a, etc.
4. **CENTRO**. Un centro es un lugar físico perteneciente a la Diputación de Cádiz donde se van a desarrollar una serie de oficios. Independientemente del centro donde se desempeñe un oficio, este va a tener el mismo conjunto de prendas.

Se ha establecido este orden porque los elementos están relacionados entre sí, de manera que no podremos asociar una prenda a un grupo si anteriormente no hemos creado el grupo en cuestión y así sucesivamente.

**IMPORTANTE**: Es necesario tener conocimiento del ejercicio en el que estamos trabajado ya que las modificaciones y consultas que realicemos se harán sobre dicho ejercicio. Por ello en la pantalla principal, sobre la imagen de bienvenida podremos ver un desplegable para seleccionar el año con el que vamos a trabajar. Esta selección se podrá realizar nuevamente en todas las pantallas de listado, como la que vemos en la imagen 2.



Imagen 2: Listado de grupos

Grupos

Para crear un nuevo grupo tenemos que dirigirnos al punto de menú "Mantenimiento" y pinchar sobre "Grupos". Lo primero que se verá será un listado con todos los grupos del año en cuestión. Si pinchamos sobre el botón "nuevo" aparecerá el formulario para el mantenimiento de grupos, como podemos ver en la figura 3. Debemos indicar el nombre que va a tomar el grupo y la fecha de baja si la tuviera y el IVA del grupo.

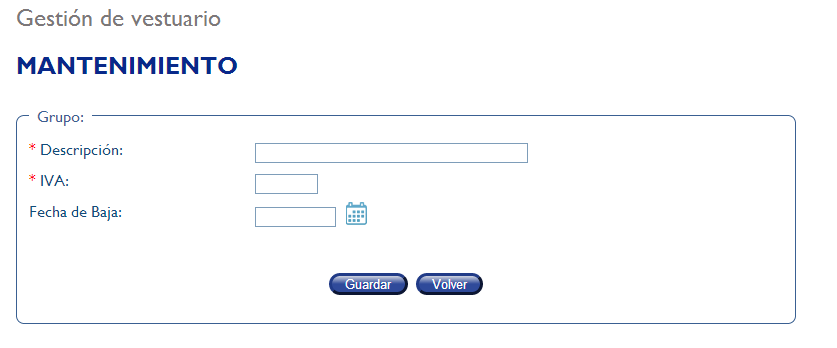


Figura 3: Nuevo grupo

Una vez completado el formulario pinchamos en el botón "guardar" para completar la acción.

Aparecerá un mensaje verde en la pantalla indicando que el grupo se ha guardado correctamente.

La edición de un grupo es análoga a la creación, sólo que en este caso en vez de pinchar sobre el botón guardar lo realizaremos sobre el "lápiz" que se puede ver en la figura 2 en la columna "acciones" y los datos del grupo que queremos editar serán cargados automáticamente en la imagen 3.

Aparecerá un mensaje verde en la pantalla indicando que el grupo se ha guardado correctamente.

Si queremos eliminar un grupo previamente creado tenemos dos formas distintas de realizarlo:

1. Eliminar con fecha de baja la que marque el sistema (el día actual).
2. Eliminar con fecha de baja que indique el usuario.

**IMPORTANTE**: En ambos casos si damos de baja un grupo las prendas asociadas a ese grupo serán dadas de baja con la misma fecha siempre y cuando esas prendas no tuvieran fecha de baja o fuera posterior a la fecha que indiquemos. Lo mismo pasaría con todas las asociaciones de oficios que tengan prendas pertenecientes a ese grupo.

Por ejemplo:

Prenda1 pertenece al grupo1

Escenario 1: La prenda 1 no tiene fecha de baja. Si damos de baja el grupo 1 a 2/8/2014 la prenda 1 tendrá la fecha de baja a día 2/8/2014

Escenario 2: La prenda 1 tiene fecha de baja a 1/8/2014. Si damos de baja el grupo 1 a 2/8/2014 la prenda 1 mantendrá su fecha de baja ya que esta es anterior

Escenario 3: La prenda 1 tiene fecha de baja a 3/8/2014. Si damos de baja el grupo 1 a 2/8/2014 la prenda 1 cambiará su fecha de baja a 2/8/2014 ya que esta es anterior.

En el primer caso la fecha de baja se realiza pinchando sobre el botón "X" de la columna "acciones" que se puede observar en la figura 2. El segundo de los casos se realiza entrando en la edición de grupo, pinchando el "lápiz", e introducir la fecha de baja deseada en el espacio con tal fin. Pulsamos "guardar".

Aparecerá un mensaje en la pantalla indicando que el grupo se ha guardado correctamente.

Prendas

La eliminación, creación y edición de prendas es análoga a la de grupos. Es importante crear los grupos antes ya que al crear y modificar la prenda es obligatorio indicar a qué grupo pertenece.

Oficios

Para crear un nuevo oficio tenemos que dirigirnos al punto de menú "Mantenimiento", "Oficios". Lo primero que se verá será un listado con todos los oficios disponibles en ese momento. Si pinchamos sobre el botón "nuevo" aparecerá el formulario para el mantenimiento de oficios, como podemos ver en la figura 4.

En esta ocasión, además de la información sobre el oficio se asocian las prendas necesarias para desempeñarlo y la cantidad y periodicidad con la que se suministran dichas prendas. Para crear asociaciones de prendas con el oficio pulsamos "añadir". Las asociaciones se pueden eliminar y editar pulsando en la "X" y en el "lápiz" de la columna de las "acciones".

Una vez añadidas las asociaciones y el introducido los datos del oficio pulsamos "guardar". Aparecerá un mensaje verde en la pantalla indicando que la edición se ha realizado correctamente.



Figura 4: Nuevo de oficio

La edición de un oficio es análoga a la creación, sólo que en este caso en vez de pinchar sobre el botón guardar lo realizaremos sobre el "lápiz" que está en la columna "acciones" cuando entremos en el listado. Los datos del oficio que queremos editar serán cargados automáticamente como podemos ver en la imagen 4. Pulsamos "guardar" cuando realicemos las modificaciones oportunas.

Aparecerá un mensaje verde en la pantalla indicando que el oficio se ha guardado correctamente.

Si queremos eliminar un oficio previamente creado tenemos dos formas distintas de realizarlo:

1. Eliminar con fecha de baja la que marque el sistema (el día actual).
2. Eliminar con fecha de baja que indique el usuario.

**IMPORTANTE**: En ambos casos si damos de baja un oficio las asociaciones de las prendas con ese oficio serán dadas de baja con la misma fecha. Además también se darán de baja las asociaciones con los centros donde se desempeñe ese oficio. En ambos casos se realizará la acción siempre y cuando no tengan fecha de baja o esta sea posterior a la que queremos introducir. Ver ejemplo de eliminación de grupos.

En el primer caso la fecha de baja se realiza pinchando sobre el botón "X" de la columna "acciones" en el listado de oficios. El segundo de los casos se realiza entrando en la edición de oficio, pinchando el "lápiz", e introducir la fecha de baja deseada en el espacio con tal fin y pulsando "guardar".

Aparecerá un mensaje en la pantalla indicando que el grupo se ha guardado correctamente.

Centros

La eliminación, creación y edición de centros es análoga a la de oficios. Es importante crear los oficios antes ya que al crear y modificar el centro es necesario indicarle que oficios están asociados a dicho centro.

* + 1. Copia de ejercicio

De un año a otro el catálogo de prendas no sufre grandes cambios, por lo que con este punto de menú se realizará una copia de los elementos de un ejercicio a otro que no haya sido creado con anterioridad.

En la copia de los elementos se tienen en cuenta la fecha de baja, así, si un elemento tiene fecha de baja el día siguiente de realizar la copia, dicho elemento conservará esa fecha de baja. Por otro lado si el elemento tiene fecha de baja el día anterior a realizar la copia, dicho elemento no se copiará para el siguiente ejercicio.

Para la realización de la copia del ejercicio pinchamos en "Copia Ejercicio" en el menú "Mantenimiento". Aparecerá una pantalla como la que se muestra en la figura 5.

Ejercicio origen: Ejercicio que va a ser copiado

Ejercicio destino: Ejercicio donde se va a copiar el ejercicio origen.

Para cambiar el ejercicio de origen en la parte superior de la pantalla aparece un desplegable donde seleccionamos el año deseado y pulsamos "Seleccionar año".



Figura 5: Copia de ejercicio

*Nota: Este punto de menú no permite sobrescribir un ejercicio que ya ha sido empezado.*

* + 1. Consulta catálogo

Una vez realizado el mantenimiento de todos los componentes, en el catálogo podremos ver todos ellos ordenados por oficio y prenda de cada oficio. Indicando la periodicidad y la cantidad en cada caso que se haya estipulado. Para ello nos dirigiremos al punto de menú de "Mantenimiento" y pincharemos sobre Catálogo de vestuario.

* 1. Gestión de necesidades

Todos los centros no van a tener los mismos oficios, ni tampoco todos los oficios utilizan las misma prendas, por lo que se debe conocer e indicar para cada oficio y en cada centro que prendas son las que se van a utilizar, teniendo en cuenta para ello la cantidad de personas que desempeñan el oficio, la periodicidad y la cantidad con la que se suministra cada prenda anualmente.

Por ejemplo, En el centro “Parque Móvil” sólo se tiene el oficio de conductor, desempeñado por 3 trabajadores. Dicho oficio tiene como prendas estipuladas camisa blanca (2 unidades anuales), pantalón negro (2 unidades anuales), calcetín gris (2 unidades trimestrales) y zapato de ejecutivo (1 unidad anual). Las necesidades de ese centro consistirán en:

* 3 trabajadores X 2 unidades de camisa blancas al año = 6 camisas blancas.
* Por la misma cuenta se sabe que serán 6 pantalones negros.
* Para los calcetines se tienen 3 trabajadores X 2 unidades de calcetines X 4 trimestres que tiene el año = 24 Calcetines grises.
* Consistiría en 3 zapatos de ejecutivo.
* Las necesidades de ese centro por lo tanto consistirán en 6 camisas blancas, 6 pantalones negros, 24 calcetines grises y 3 zapatos ejecutivos anuales.
  + 1. Generación de necesidades y alta de actuación

Para la generación de necesidades pinchamos sobre el menú "Gestión de Necesidades", "Generar necesidades". Aparecerá una pantalla como la que podemos ver en la figura 6 donde se debe indicar el año al que pertenece las necesidades que queremos generar. Pulsamos "generar necesidades"

IMPORTANTE: Cuando generamos las necesidades, si hay necesidades generadas para ese año, éstas se eliminan y se vuelven a crear.

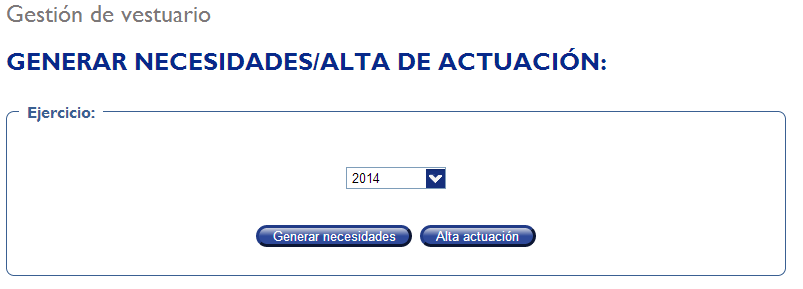


Figura 6: Generar necesidades

Una vez realizadas las necesidades deberos de realizar el alta de actuación en Contrat@para que comience el proceso licitatorio. **IMPORTANTE:** Una vez realizado el alta de la actuación, la opcion de alta actuación quedará bloqueada hasta que el administrador la desbloquee. Ver 2.4.3 Deshacer alta de actuación.

Para realizar el alta de actuación, indicamos el año de las necesidades que queremos enviar y pulsamos sobre "alta de actuación".

Aparecerá un mensaje verde indicando que se ha realizado el alta con éxito.

* + 1. Deshacer alta de actuación

La acción de realizar el alta de una actuación es irreversible, por lo que una vez realizado el alta no tiene vuelta atrás. En el caso de que haya algún error en las necesidades y por lo tanto la actuación haya sido errónea, inmediatamente se debe de comunicar a la persona encargada del registro de actuaciones de Contrat@.

Una vez se le haya indicado que el alta es errónea seguimos los siguientes pasos ordenadamente:

1. Realizamos las modificaciones en los centros, oficios, prendas o grupos oportunas.
2. Volvemos a generar las necesidades para el año en cuestión.
3. Reactivamos el alta. Desde el punto de menú "Gestión de necesidades", "Rehacer alta actuación". Indicamos el año que ha sido erróneo y pulsamos "activar alta". Aparecerá un mensaje indicando que la operación se ha realizado con éxito.
4. Volvemos a realizar el alta de la actuación como en el apartado 2.4.1
   * 1. Consulta de necesidades

Para la consulta de necesidades de un ejercicio pulsamos sobre "Gestión de necesidades", "Consulta de necesidades" y aparecerá la pantalla como la figura 7. Desde ahí realizaremos la selección para aplicar el filtro de las necesidades que queremos ver de manera que podremos realizarlo por centro, grupo o ambas. Pulsamos "seleccionar" y aparecerá la tabla con la consulta solicitada.



Figura 7: Filtro de necesidades

* 1. Gestión de tallaje

Una vez completado el proceso licitatorio, la empresa adjudicataria realizará el tallaje de las prendas in situ y le indicará a Contratación el resultado de ese tallaje que a su vez nos lo facilitará para realizar la gestión del tallaje.

El tallaje consiste en comprobar las necesidades reales in situ, verificando el número de prendas y empleados.

* + 1. Generación de tallaje

En el punto de menú "Gestión de necesidades", "Gestión de tallaje", una vez indicado el ejercicio con el que queremos trabajar pulsamos "seleccionar" y aparecerá una pantalla como la de la figura 8.

Si es la primera vez que realizamos el tallaje para ese ejercicio en la tabla se cargarán las necesidades de ese año para la generación del tallaje con las modificaciones oportunas. Si el tallaje ya se ha realizado con anterioridad aparecerán esos datos para ser modificados.



Figura 8: Tallaje

La "cantidad" indica el numero de prendas para ese centro y esa prenda que hay en base de datos. En la columna de "cantidad real" debemos indicar el resultado del tallaje realizado por la empresa adjudicataria.

Pulsamos "guardar" y aparecerá un mensaje verde confirmando que ha sido guardado con éxito.

* + 1. Consulta de tallaje

Para consultar el tallaje de un año pinchamos sobre consulta de tallaje del menú "Gestión de tallaje", "Consulta de tallaje". Aparecerá una pantalla para seleccionar el año con el que queremos trabajar. Una vez indicado el año aparecerá una pantalla como la de la figura 8 pero en la que no se pueden editar elementos.

* + 1. Generar precio prendas

Es importante realizar el mantenimiento del precio de las prendas ya que al dar de alta la actuación en Contrat@ se realiza un presupuesto aproximado basándose en el coste de las prendas del año anterior. Si la prenda es nueva o nunca se ha realizado este mantenimiento tendrá un coste de 0. Por lo que el presupuesto aproximado no será muy fiable.

Para generar el precio de las prendas nos vamos al punto de menú "Generar precios prendas" en "Gestón de tallaje". Indicaremos el año con el que queremos trabajar y aparecerá un listado como el de la figura 9. Realizamos las modificaciones oportunas en el precio y pulsamos "guardar".

Aparecerá un mensaje verde indicando que la acción se ha ejecutado correctamente.



Figura 9: Mantenimiento de precio de prendas

* 1. Extracción de listado a documento Excel

Cada vez que estemos observando un listado podremos realizar una exportación a un documento Excel, para ello pulsaremos sobre la palabra "Excel" en la parte inferior izquierda de los listados.

1. ACCIONES DEL MANTENIMIENTO PREVENTIVO
   1. Operaciones sobre la aplicación

Es responsabilidad del usuario el control de la fecha y hora de los sistemas donde se utiliza la aplicación (tanto servidores como estaciones cliente) siendo recomendable la sincronización por el protocolo NTP de la fecha y hora del sistema, de forma que la introducción de datos como fecha de bajase realice de forma correcta.

* 1. Entorno de trabajo

El puesto del operador debe estar compuesto por un equipo informático con las características descritas en el apartado “2.1.2 Estaciones cliente”.

El equipo deberá tener conexión al servidor donde se despliega Vestium así como un navegador web

compatible con la aplicación.

* 1. Cuadro resumen de las acciones y periodicidad del mantenimiento

Se deberán realizar, como mínimo las siguientes tareas de revisión y mantenimiento con la periodicidad marcada.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea** | **Objetivo** | **Acción** | **Periodicidad** |
| Limpieza y mantenimiento de equipos | Mantener los equipos hardware en perfecto estado de funcionamiento. | Realizar las tareas de limpieza y mantenimiento de los equipos (escáner y PC) siguiendo las indicaciones del fabricante del equipo. | Mensual |
| Revisión de puesto de operador | Asegurar unas condiciones ambientales óptimas del lugar donde tiene lugar el proceso. | Asegurar que el operador mantiene su entorno de trabajo correctamente ordenado, libre de exposiciones ambientales (líquidos, …) y manteniendo la información confidencial visible únicamente durante el tiempo requerido | Diaria |
| Revisión del certificado digital | Mantener el certificado digital empleado activo, asegurándose de que el mismo no se encuentra en estado caducado ni revocado. | Revisar periódicamente que el certificado a emplear se encuentre absolutamente vigente | Semanal |
| Revisión de confidencialidad | Asegurar el secreto de las claves de acceso a las aplicaciones incluidas en la solución ni la contraseña de acceso e instalación del certificado digital. | Revisar periódicamente que el operador no mantenga la información de acceso de forma accesible a terceros | Mensual |
| Hora del sistema | Asegurar el mantenimiento de una hora cierta en el sistema de digitalización | Revisar periódicamente que la fecha y hora del equipo que realiza la digitalizaciones es correcta , para lo cual se recomienda la sincronización diaria a través del protocolo NTP. | Semanal |
| Calidad de la imagen obtenida | Asegurar la calidad y la corrección de la imagen y del proceso en su conjunto para detectar posibles fallos del sistema. | Revisar periódicamente mediante la obtención de muestras de la calidad y legibilidad de las imágenes resultantes | Diaria |
| Actualizaciones | Mantener el software actualizado y realizar las actualizaciones oportunas siempre y cuando esta acción no suponga un cambio de versión tal que impida el funcionamiento con cualquiera de los productos incluidos en la solución. | Verificar la necesidad de realización actualizaciones del software, sistema operativo o aplicaciones dependientes recomendadas por el fabricante del mismo. | En función de las actualizaciones del fabricante. Comprobación diaria |
| Back up | Respetar la política de back up y recuperación. | Realizar una prueba completa de back up y recuperación del sistema siguiendo las pautas del fabricante | Mensual |
| Sistema de alertas | Mantener activo el sistema de alertas en lo relativo al espacio del disco y problemas de consistencia o fallos en la base de datos empleada. | Realizar pruebas de evaluación del sistema de alertas | Mensual |
| Logs | Mantener un sistema de información de eventos de sistema adecuado | Revisar que se están registrando los eventos relevantes | Mensual |
| Recuperación del sistema | Asegurar que el entorno responde ante un cierre inesperado | Realizar una prueba periódica de reinicio de sistema y verificar que todos los servicios se activan correctamente | Mensual |

Anexo B: Proceso licitatorio

Un proceso licitatorio es un proceso de selección de un contratante particular [A]. Se pueden distinguir tres tipos de proceso licitatorios:

* Licitación Privada: Los participantes a la licitación son los que selecciona la administración, invitándolos personalmente.
* Licitación pública: El acto mediante el cual la administración selecciona la lista de ofertantes a invitar.
* Licitación Restringida: Hay licitación restringida cuando el llamado a realizar las ofertas está limitado por los requisitos o características que la administración impone como condición para realizar la oferta.

1. La Licitación Pública

Es el procedimiento de selección en que la Administración Pública invita públicamente y de forma general a recibir propuestas sobre un objeto y condiciones, con la finalidad de seleccionar la oferta más adecuada o conveniente para realizar un contrato con el autor de dicha oferta.

Son tres principios que rigen la licitación pública: Oposición o libre recurrencia, publicidad e igualdad.

Figura 75: Esquema proceso licitación pública

1. Oposición o libre concurrencia

A mayor concurrencia de ofertantes mayor es la posibilidad que se obtenga las mejores condiciones y el mejor precio, o al menos el más conveniente.

En caso de presentarse una sola oferta, puede contratarse con ese oferente solitario solo si ha cumplido con los requisitos previstos.

1. Publicidad

La publicación se ha de realizar en el “boletín Oficial” y en un diario de amplia circulación, según el objeto. Esto permite que se conozca el proceso licitatorio por todo aquel que estuviera interesado. La publicidad además asegura la transparencia en el acto.

1. Igualdad

Este principio es considerado el más importante. Los procesos licitatorios existen para poder primar la igualdad de los oferentes en el procedimiento. No se pueden incluir conductas que ponga a un oferente en una posición más ventajosa frente a los demás.

1. Etapas del procedimiento licitatorio
2. Elaboración del pliego de bases y condiciones

Cláusulas donde se especifica el suministro, obra o servicio que se licita, y donde se estableen las condiciones del contrato, determinando el trámite a seguir en el procedimiento de licitación.

1. El llamado a licitación

Es la invitación que se realiza públicamente para que se presenten ofertas con el fin de contratar la prestación en las condiciones que se definen. Debe ser un llamado amplio para lograr una buena oferta. Los que concurran al llamado se llaman licitadores u oferentes.

1. Oferta y presentación del pliego

La oferta es el acto jurídico de propuesta que presenta el oferente con la aspiración de ser el contratante de la administración, siempre ajustada a las exigencias del pliego de la licitación.

* Presentación: las ofertas deben ser presentadas dentro del término que fija el licitante en el llamado a licitación. La presentación puede tener requisitos formales que deber ser cumplidos.
* Requisitos: son numerosos los requisitos que debe llenar la oferta. Existen requisitos que hacen al monto ofertado, que debe ser claro, determinado, escrito y firmado. también se imponen requisitos relativos a la persona del oferente.

Garantía precontractual: juntamente con la propuesta, el oferente debe acompañar una garantía de mantenimiento de oferta. Asegura al licitante que el oferente no se va a arrepentir de su oferta antes del cumplimiento. Puede ser dinero en efectivo, títulos públicos, etc.

1. **Apertura de los sobres**

En el lugar y día que se determine para la celebración del acto, seabriran las puertas y se procederá aa abrir las propuestas. Presentes estarán los funcionarios designados por la dependencia y todos aquellos que lo quieran presenciar. Es un acto público.

De lo acontecido se labra un acta y en ella se puede incluir todas las observaciones que los presentes quieran formular.

A partir del acto de apertura se produce el análisis de las ofertas por parte de los distintos integrantes del licitante.

1. **Adjudicación**

La adjudicación es el acto administrativo donde el órgano competente de la administración licitante elige de entre todos los ofertantes que se han admitido, a el que tras la valoración y la comparativa presenta la mejor opción.

1. **Perfeccionamiento**

Una vez seleccionado el oferente, se debe notificar en forma fehaciente dentro del plazo estipulado para el mantenimiento de las ofertas. Una vez notificado el oferente y habiéndose perfeccionado el contrato, la administración procede a la devolución de la fianza a aquellos que no fueron elegidos.

1. **La mejora de propuesta**

Se da en el caso en que existan dos o más oferentes con una misma propuesta. Tiene por objeto poder seleccionar al mejor oferente.

Apéndice A: GNU Free Documentation License v 1.3



Version 1.3, 3 November 2008 Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.[[3]](#footnote-3)

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. [B]

**PREAMBLE**

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

**VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

**COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

**MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

**COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

**COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

#### AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

#### TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

#### TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

#### FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

#### RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

#### ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document

under the terms of the GNU Free Documentation License, Version 1.3

or any later version published by the Free Software Foundation;

with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU

Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the

Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Apéndice B: Documentación Estándar de Licitación Pública

#### Regulación específica de la Plataforma de Contratación del Estado[C]

* **Disposición adicional tercera de la Ley 20/2013, de 9 de diciembre, de garantía de la unidad de mercado.**

La Plataforma de Contratación del Estado regulada en el artículo 334 del texto refundido de la Ley de Contratos del Sector Público, aprobado por el Real Decreto Legislativo 3/2011, de 14 de noviembre, pasará a denominarse Plataforma de Contratación del Sector Público.

En la Plataforma se publicará, en todo caso, bien directamente por los órganos de contratación o por interconexión con dispositivos electrónicos de agregación de la información de las diferentes administraciones y entidades públicas, la convocatoria de licitaciones y sus resultados de todas las entidades comprendidas en el apartado 1 del artículo 3 del texto refundido de la Ley de Contratos del Sector Público.

* **Artículo 334 del Texto Refundido de la Ley de Contratos del Sector Público**
  1. La Junta Consultiva de Contratación Administrativa del Estado, a través de sus órganos de apoyo técnico, pondrá a disposición de todos los órganos de contratación del sector público una plataforma electrónica que permita dar publicidad a través de internet a las convocatorias de licitaciones y sus resultados y a cuanta información consideren relevante relativa a los contratos que celebren, así como prestar otros servicios complementarios asociados al tratamiento informático de estos datos. En todo caso, los perfiles de contratante de los órganos de contratación del sector público estatal deberán integrarse en esta plataforma, gestionándose y difundiéndose exclusivamente a través de la misma. En las sedes electrónicas de estos órganos se incluirá un enlace a su perfil del contratante situado en la Plataforma de Contratación del Estado.
  2. La plataforma deberá contar con un dispositivo que permita acreditar fehacientemente el inicio de la difusión pública de la información que se incluya en la misma.
  3. La publicación de anuncios y otra información relativa a los contratos en la plataforma surtirá los efectos previstos en la Ley.
  4. El acceso de los interesados a la plataforma de contratación se efectuará a través de un portal único. Reglamentariamente se definirán las modalidades de conexión de la Plataforma de Contratación con el portal del «Boletín Oficial del Estado».
  5. La Plataforma de Contratación del Estado se interconectará con los servicios de información similares que articulen las Comunidades Autónomas y las Entidades locales en la forma que se determine en los convenios que se concluyan al efecto.
* **Orden EHA/1220/2008, de 30 de abril, por la que se aprueban las instrucciones para operar en la Plataforma de Contratación del Estado**.
* **Dictamen 31/11 de la Junta Consultiva de Contratación Administrativa sobre la obligatoriedad de integrar el perfil de contratante de los organismos y entidades del sector público estatal en la Plataforma de Contratación del Estado así como de las informaciones a publicar en la misma.** El dictamen responde al escrito dirigido por la Dirección General del Patrimonio del Estado a la Junta Consultiva de Contratación Administrativa solicitando informe sobre la obligatoriedad de que los organismos y entidades del sector público estatal integren sus perfiles de contratante en la Plataforma de Contratación del Estado así como de las informaciones a publicar en la misma.
* **Resolución de 3 de marzo de 2010, de la Dirección General del Patrimonio del Estado, por la que se publica la Recomendación de la Junta Consultiva de Contratación Administrativa sobre el envío de anuncios a la Comisión Europea**. Se recuerda a los usuarios de la Plataforma de Contratación del Estado, que los anuncios a los que se refiere la resolución anterior pueden publicarse directamente desde la Plataforma de Contratación del Estado.

1. Free Documentation License [↑](#footnote-ref-1)
2. Free Software Foundation [↑](#footnote-ref-2)
3. <http://fsf.org/> [↑](#footnote-ref-3)