

UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA APLICADA



Escola Tècnica Superior
d'Informàtica Aplicada

**DESARROLLO DE UNA APLICACIÓN
PARA LA GESTIÓN DE LOS RECURSOS
DE IMPRESIÓN DE LA UNIVERSIDAD
UCH-CEU**

ELABORADO POR:

Marcos Pitarque

DIRECTOR DEL PROYECTO:

Eliseo Marzal Calatayud

INDICE

1.- Introducción	4
1.1 Motivación	4
1.2 Estructura de la memoria.....	5
2.- ESPECIFICACIÓN DE REQUISITOS.....	6
2.1.1 Introducción.....	6
2.1.2 Descripción general.....	8
2.1.3 Requerimientos específicos	10
3.- ANÁLISIS	18
3.1 Casos de uso.....	18
3.2 Diagrama de clases	21
4.- DISEÑO.....	21
4.1 Estilo arquitectónico.....	21
4.2 Nivel de presentación.....	24
4.2.1 Mapas navegacionales.....	25
4.3 Nivel lógico.....	42
4.4 nivel de persistencia.....	43
4.4.1 Modelo de datos Entidad-Relación.....	43
5.- IMPLEMENTACIÓN.....	44
5.1 Patrones de traducción.....	44
5.2 Esquema de clases.....	48

5.2.1 Nivel de presentación.....	48
5.2.2 Nivel lógico.....	52
5.2.3 Nivel de persistencia.....	52
5.3 Tecnologías utilizadas.....	52
5.4 Notas adicionales.....	56
6.- EVALUACIÓN.....	61
7.- CONCLUSIONES.....	64
8.- BIBLIOGRAFÍA.....	66
9.- ANEXOS.....	68
9.1 Manuales de instalación.....	68
9.1.1 Prerrequisitos.....	68
9.1.2 Instalación del portal Web.....	71
9.1.3 Instalación de las aplicaciones de escritorio.....	74
9.2 Manual del usuario.....	75
9.2.1 Manual aplicación cliente.....	76
9.2.2 Manual aplicación servidor.....	77
9.2.3 Manual aplicación Web.....	80

1.- INTRODUCCIÓN

Este documento pretende describir el trabajo realizado en el Proyecto Final de Carrera de Ingeniería Técnica en Informática de Gestión, cursada en la Escuela Técnica Superior de Informática Aplicada de la Universidad Politécnica de Valencia.

El proyecto se titula “Desarrollo de un Gestor de Impresiones para un aula informática”, ha sido dirigido por Eliseo Marzal y realizado por Marcos Pitarque Tomás.

Objetivos Motivación

Actualmente, en la universidad dónde llevo a cabo las prácticas se dispone de varias aulas de informática dónde se puede encontrar una impresora compartida en red para todos los ordenadores de las mismas.

Éste sistema, útil para un hogar o un pequeño comercio, no es eficiente para la gran cantidad de clientes simultáneos que pueden querer acceder a este servicio en una universidad provocando colapsos en la cola de impresión y largos tiempos de espera para obtener una impresión por parte de los usuarios.

Es por ello, que con este proyecto he querido diseñar una herramienta que subsane todos los problemas anteriormente mencionados proporcionando una interfaz que mejore la interacción entre los usuarios y el sistema de impresión.

Con esta idea, he desarrollado una aplicación que utilizarán tanto alumnos como personal de la universidad y que permitirá, principalmente, una gestión más optimizada de las impresiones por parte de los alumnos permitiendo enviar trabajos, eliminarlos, decidir cuáles imprimir, etc. y un control del sistema de impresión por parte del personal permitiendo conocer estadísticas de uso, duración de los cartuchos de tinta, etc.

Este proyecto, que consta de dos aplicaciones de escritorio (una para los clientes y otra para el servidor de impresión) y otra aplicación Web (utilizada por el personal para gestionar el servicio) puede ser implantada en cualquier sistema de impresión con una o múltiples impresoras con el único requisito de que exista un servidor conectado a la impresora compartida y los clientes se encuentren en la misma red.

Se ha pretendido proporcionar un entorno amigable, teniendo en cuenta el amplio abanico de perfiles de usuario, existiendo desde poco experimentados hasta profesionales. Por este motivo, he intentado que la aplicación sea muy sencilla de manejar, con controles claramente definidos y operaciones intuitivas.

En este conjunto de aplicaciones podemos encontrar a grandes rasgos dos clases de usuarios, los alumnos y profesores y el personal del Servicio de Informática. Los primeros, desde la aplicación cliente podrán enviar sus trabajos al servidor de impresión, obteniendo de antemano el precio unitario y total y la cantidad de páginas. Por otro lado, desde la aplicación instalada en el servidor, podrán seleccionar los trabajos a imprimir así como pagarlos. En cuanto al personal, desde la aplicación servidor podrán modificar los parámetros de impresión como por ejemplo el precio de las copias, la configuración de las impresoras, etc.

La solución de este proyecto se ha centrado en dar a los usuarios todas las herramientas necesarias para gestionar todo lo necesario para llevar adelante la gestión de los trabajos de impresión que en este caso, en la universidad, se puedan desempeñar.

Estructura de la memoria

1. En primer lugar se explica de forma global de qué trata el proyecto y los objetivos de la aplicación.
2. El siguiente punto será una descripción detallada de la especificación de requisitos software del proyecto, desarrollada siguiendo el estándar IEEE 830-1998.
3. En este apartado se va a detallar el análisis de la aplicación mediante el modelo UML. Para representar gráficamente la funcionalidad que tendrá la aplicación se utilizará los diagramas de caso de uso, en los que podremos ver las acciones que puede efectuar cada usuario; y para saber qué clases intervienen en el sistema y como están relacionadas entre sí, se utilizará el diagrama de clases.
4. En este punto se va a explicar el diseño de la aplicación. En él se presentan las tres capas de nuestra aplicación, mediante mapas navegacionales se representará la visión global de una aplicación según un perfil de usuario, se describirá la estructura conceptual de la aplicación y se desarrollará el diagrama Entidad-Relación para obtener la base de datos.
5. En este capítulo se explica cómo se ha realizado la implementación de la aplicación y qué archivos se han creado para que la aplicación funcione correctamente.
6. En el apartado de Evaluación se llevarán a cabo una serie de pruebas para asegurar el correcto funcionamiento de la aplicación.
7. Los últimos puntos del proyecto son: Conclusiones, Bibliografía y Anexos, en los que se hablará de las conclusiones obtenidas con el desarrollo de dicho proyecto, la bibliografía utilizada y en la parte de Anexos aparecerán los siguientes manuales: de instalación, de usuario y de administración.

2.- Especificación de Requisitos

2.1 Especificación de Requerimientos Software

Esta especificación de requerimientos pertenece al desarrollo del Proyecto Final de Carrera “Desarrollo de un Gestor de Impresiones para un aula informática”, desarrollado siguiendo las directrices **IEEE Std. 830-1998**, para la Especificación de Requisitos Software.

2.1.1 Introducción

2.1.1.1 Propósito

El propósito de esta Especificación de Requisitos Software (ERS) es exponer cuales son los requerimientos que tendrá una aplicación que se encargará de gestionar los trabajos de impresión de un aula informática así como, junto con el cliente, formalizar su funcionalidad.

2.1.1.2 Ámbito

El proyecto a desarrollar está orientado a la gestión de las colas de impresión de las aulas informáticas de la universidad. Permitirá organizar de manera más eficiente a la actual los trabajos de impresión de los usuarios minimizando los tiempos de espera y la congestión ofreciendo la siguiente funcionalidad:

APLICACIÓN CLIENTE

1.- Gestión de los trabajos de impresión

El usuario podrá añadir o eliminar documentos a su “bolsa de impresión” siendo informado del número de páginas y del precio tanto de cada trabajo como el total.

2.- Enviar trabajos de impresión al servidor

El usuario podrá enviar los trabajos que considere oportunos a la “bolsa de impresión” del servidor.

APLICACIÓN SERVIDOR

1.- Gestión de los trabajos

El usuario podrá seleccionar los trabajos que finalmente serán impresos. También podrá borrar los que no le interesen.

2.- Gestión de impresoras

El personal del departamento de informática podrá añadir o eliminar puntos de impresión pudiendo seleccionar cualquiera de las impresoras instaladas en el equipo servidor.

3.- Opciones de impresión

El personal del departamento de informática podrá modificar los parámetros de impresión tales como el precio de los folios, la ubicación de la “bolsa de impresión” dentro del servidor, etc.

APLICACIÓN WEB

1.- Muestreo de datos y estadísticas del sistema de impresión.

2.1.1.3 Definiciones, acrónimos y abreviaturas

Definiciones

Bolsa de Impresión: Al cambiar la lógica de impresión con esta aplicación, que en vez de imprimir por orden de envío como hace la cola de impresión, imprime lo que el usuario selecciona al llegar al servidor.

Documento: Cualquier archivo con formato compatible con la aplicación, es decir formatos de office 97 y 07 (doc, docx, xls, xlsx, ppt, pptx) y pdf.

Trabajo de Impresión: Documento que además incluye metadatos referentes al sistema de impresión como por ejemplo usuario que lo ha enviado, rango de páginas a imprimir, precio, etc.

Usuario impresor nulo: Mientras este metadato contenga valor nulo, significará que el trabajo de impresión continúa en el servidor y nadie lo ha impreso.

Acrónimos

UCH-CEU: Universidad dónde se pretende aplicar el proyecto.

BI: Bolsa de Impresión.

TI: Trabajo de Impresión

2.1.1.4 Referencias

Para el desarrollo de esta memoria se han tenido en cuenta las siguientes fuentes:

- 1) IEEE Std. 830- IEEE Guide to Software Requirements Specifications. IEEE-SA Standards Board.
- 2) <http://es.wikipedia.org/wiki/IEEE>
- 3) www.ieee.org
- 4) Apuntes de la asignatura “Técnicas Avanzadas para el Desarrollo de Software”
- 5) Universidad Cardenal Herrera

**Avda. Seminario s/n
46113 Moncada (Valencia)
Tel. 96 136 90 00**

2.1.2 Descripción General

Este apartado abarca los factores que afectan al desarrollo del proyecto así como a sus requerimientos.

2.1.2.1 Perspectiva del producto

El proyecto a desarrollar debe ser capaz de gestionar el flujo de impresiones de cualquier aula informática que cuente con un servidor para tal efecto, proporcionando una interfaz amigable e intuitiva.

Las aplicaciones de escritorio están desarrolladas para ejecutarse sobre Windows y la aplicación Web deberá ser compatible con cualquier navegador actual estándar que soporte las últimas tecnologías web.

2.1.2.2 Funcionalidad de la aplicación

La funcionalidad del conjunto de aplicaciones, en función de los tipos de usuarios será la siguiente:

- a) Alumnos y Personal perteneciente a la universidad:

App. Cliente:

- Añadir trabajos a la bolsa de impresión.
- Modificar las propiedades del trabajo de impresión.
- Eliminar trabajos de la bolsa de impresión.
- Consultar precio unitario y total de sus trabajos de impresión.

- Consultar historial diario de trabajos enviados a la bolsa de impresión.
- Enviar trabajos de impresión al servidor.

App. Servidor:

- Consultar trabajos de impresión clasificados por usuario.
- Consultar historial de impresiones diario.
- Gestionar trabajos a imprimir.
- Imprimir los trabajos seleccionados.

b) Personal del departamento de informática de la universidad

App. Servidor:

- Consultar trabajos de impresión clasificados por usuario.
- Consultar datos de la tarjeta de alumno o personal de la universidad.
- Consultar historial de impresiones diario.
- Gestionar trabajos a imprimir.
- Imprimir los trabajos seleccionados.
- Modificar las preferencias de la impresión tales como precio, impresora, etc.

App. Web

·Permitirá la consulta de los datos almacenados en la BD útiles para mostrar estadísticas de uso y para controlar la facturación, etc.

2.1.2.3 Características del Usuario

Los principales usuarios que van a utilizar esta aplicación van a ser: el alumno y el integrante del servicio informático.

El alumno es un usuario que utilizará la aplicación instalada en el cliente para enviar sus documentos al servidor para posteriormente, utilizando la aplicación instalada en el mismo imprimir sus documentos.

El personal del servicio informático utilizará la aplicación instalada en el servidor para gestionar las propiedades de las impresoras tales como el precio por página y también utilizará la aplicación web para monitorizar la utilización del sistema y además, modificar algunas propiedades también ofertadas en la aplicación del servidor.

2.1.2.4 Obligaciones Generales

Para mantener un correcto funcionamiento del sistema, la BD se ha de actualizar de una manera correcta tanto ya que los datos son accedidos tanto desde las aplicaciones escritorio como desde la aplicación web.

2.1.2.5 Asunciones y Dependencias

Las aplicaciones de escritorio están diseñadas para ser utilizadas en Windows (XP, Vista ó 7). El sitio web podrá ser alojado en cualquier tipo de plataforma de servidor y podrá ser visitada desde cualquier navegador aunque en un principio está diseñada para Internet Explorer.

2.1.3. Requerimientos Específicos

2.1.3.1. Requerimientos Funcionales

A.-En el cliente

A.1 Añadir un trabajo a la lista

- Introducción:** Se selecciona un documento para añadirlo a la lista de documentos a enviar.
- Entradas:** Path completo del documento a insertar.
- Proceso:** Se genera un nuevo documento .pdf con el contenido del documento original y posteriormente se obtienen datos de interés.
- Salida:** Visualización en la lista de documentos pendientes mostrada en el formulario principal del cliente.

A.2 Eliminar un trabajo de la lista

- Introducción:** Se selecciona un documento de la lista de documentos pendientes para ser eliminado de la misma.
- Entradas:** Identificador del trabajo a eliminar.
- Proceso:** Se obtiene de la BD el path del documento asociado al identificador obtenido y se elimina el pdf generado en la carpeta del cliente.

•**Salida:** El documento seleccionado desaparece de la lista de documentos pendientes y el archivo pdf es eliminado de la carpeta de cliente de servidor.

A.3 Enviar documentos al servidor

•**Introducción:** Tras hacer clic sobre el botón correspondiente del formulario principal, los trabajos de la lista se envían al servidor de impresión.

•**Entradas:** Identificador de cada trabajo a enviar.

•**Proceso:** Se obtiene el path asociado a cada identificador obtenido de la lista, y cada documento es movido a la carpeta compartida del servidor.

•**Salida:** Visualización por pantalla del éxito de la acción.

B. – En la aplicación servidor

B.1 Cambiar bolsa de impresión

•**Introducción:** Tras hacer clic sobre el botón correspondiente del formulario principal, se muestra el formulario de preferencias y con él, se ofrece la posibilidad de cambiar el directorio de la bolsa de impresión.

•**Entradas:** Nuevo directorio para la bolsa de impresión.

•**Proceso:** Se actualiza en la BD el directorio para la bolsa de impresión y a continuación se mueven los documentos que se encontraban en el antiguo directorio al nuevo.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.2 Cambiar precio por página de la impresora

•**Introducción:** Tras hacer clic sobre el botón correspondiente del formulario principal, se muestra el formulario de preferencias y con él, se ofrece la posibilidad de cambiar precio por página de cada impresora.

•**Entradas:** Nuevo precio por página para una impresora.

•**Proceso:** Se actualiza en la BD el campo precio de la tabla de impresoras.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.3 Cambiar precio contraseña de acceso a las preferencias.

•**Introducción:** Tras hacer clic sobre el botón correspondiente del formulario principal, se muestra el formulario de preferencias y con él, se ofrece la posibilidad de cambiar la contraseña de acceso a las preferencias.

•**Entradas:** Nueva contraseña

•**Proceso:** Se actualiza en la BD el campo contraseñas con la fecha actual.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.4 Seleccionar todo

•**Introducción:** Selecciona todos los trabajos de la lista de trabajos pendientes.

•**Entradas:** Ninguna

•**Proceso:** Se recorre la lista de trabajos pendientes y se cambia su atributo "checked" a true y se actualiza el precio total.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.5 Deseleccionar todo

•**Introducción:** Deselecciona todos los trabajos de la lista de trabajos pendientes.

•**Entradas:** Ninguna

•**Proceso:** Se recorre la lista de trabajos pendientes y se cambia su atributo "checked" a false" y se actualiza el precio total.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.6 Invertir selección

•**Introducción:** Invierte la selección de todos los trabajos de la lista de documentos pendientes.

•**Entradas:** Ninguna

•**Proceso:** Se recorre la lista de trabajos pendientes y se cambia su atributo y se invierte el valor del atributo "checked" de cada trabajo y se actualiza el precio total.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.7 Seleccionar trabajos manualmente

•**Introducción:** Se ofrece la posibilidad de seleccionar los trabajos necesarios haciendo clic sobre ellos en la lista de trabajos pendientes.

•**Entradas:** Ninguna

•**Proceso:** Se recorre la lista de trabajos pendientes y se cambia su atributo “checked” y se actualiza el precio total.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.8 Cambiar impresora seleccionada.

•**Introducción:** Se ofrece la posibilidad de seleccionar otra impresora que no sea la predefinida.

•**Entradas:** Nombre de la nueva impresora seleccionada.

•**Proceso:** Se actualiza el precio de cada documento calculándolo a partir del precio por página de la nueva impresora y se recalcula el precio total.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.9 Seleccionar trabajos manualmente

•**Introducción:** Se ofrece la posibilidad de seleccionar los trabajos necesarios haciendo clic sobre ellos en la lista de trabajos pendientes.

•**Entradas:** Ninguna

•**Proceso:** Se recorre la lista de trabajos pendientes y se cambia su atributo “checked” y se actualiza el precio total.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.10 Seleccionar filtro usuario

•**Introducción:** Los trabajos se filtran por nombre de usuario

•**Entradas:** Nombre del usuario

•**Proceso:** Se recorre en la BD toda la lista de trabajos por imprimir y en una tabla del formulario principal, se muestran los trabajos por imprimir de dicho usuario.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.11 Eliminar trabajos seleccionados

•**Introducción:** Tras hacer clic en el botón correspondiente del formulario principal, los trabajos seleccionados serán eliminados de la lista de documentos pendientes y los archivos eliminados de la carpeta del servidor de impresión.

•**Entradas:** Identificadores de los trabajos a eliminar.

•**Proceso:** Se recorre la lista de trabajos pendientes y se obtiene el identificador de los seleccionados. A continuación se eliminan tanto de la BD como del directorio del servidor y se refresca la lista con los documentos pendientes.

•**Salida:** Visualización por pantalla del éxito de la acción.

B.12 Imprimir trabajos seleccionados

•**Introducción:** Tras hacer clic en el botón correspondiente del formulario principal, los trabajos seleccionados serán enviados a la impresora.

•**Entradas:** Identificadores de los trabajos a imprimir.

•**Proceso:** Se recorre la lista de trabajos pendientes y se obtiene el identificador de los seleccionados. A continuación se crea un proceso por cada trabajo y éstos son impresos.

•**Salida:** Visualización por pantalla del éxito de la acción.

C.- En la aplicación Web

C.1 Ver Volumen de Impresión

•**Introducción:** Se mostrará una gráfica con la evolución del volumen de impresión.

•**Entradas:** Navegación por la web

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.2 Ver Trabajos no Impresos

•**Introducción:** Se mostrará una gráfica con la evolución del volumen de trabajos cuya impresión ha fallado.

•**Entradas:** Navegación por la web

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.3 Ver Volumen de Facturación

•**Introducción:** Se mostrará una gráfica con la evolución del volumen de facturación.

•**Entradas:** Navegación por la web

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.4 Ver utilización de Impresoras

•**Introducción:** Se mostrará una gráfica con la utilización de las impresoras.

•**Entradas:** Navegación por la web.

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.5 Ver Contenido de la Tabla Envíos.

•**Introducción:** Se mostrará una gridview con el contenido de la tabla "Envíos"

•**Entradas:** Navegación por la web.

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.6 Ver Contenido de la Tabla Passwords.

•**Introducción:** Se mostrará una gridview con el contenido de la tabla "passwords"

•**Entradas:** Navegación por la web.

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.7 Ver Contenido de la Tabla Directorios.

•**Introducción:** Se mostrará una gridview con el contenido de la tabla “paths”

•**Entradas:** Navegación por la web.

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.8 Ver Contenido de la Tabla printers.

•**Introducción:** Se mostrará una gridview con el contenido de la tabla “printers”

•**Entradas:** Navegación por la web.

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada.

C.9 Modificar preferencias de las impresoras.

•**Introducción:** Se mostrará una lista con las impresoras instaladas en el sistema y se ofrecerá la posibilidad de modificar algunos de sus parámetros.

•**Entradas:** Navegación por la web y nuevas preferencias.

•**Proceso:** Se recorre la BD en función de los parámetros solicitados.

•**Salida:** Visualización por pantalla de la información solicitada y confirmación de la operación.

2.1.3.2 Requerimientos de interfaces externos

•**Interfaces de Usuario**

En este apartado se va a hablar sobre los mecanismos mediante los cuales los usuarios podrán interactuar con la aplicación.

En cuanto a la interacción visual, los formularios de esta aplicación tendrán un aspecto estándar, lo cual facilitará la utilización por parte del usuario.

Actualmente, todas las entradas al sistema se llevan a cabo mediante los dispositivos clásicos, tales como teclado y ratón aunque se prevé la utilización de un lector de tarjetas “smart card” para la autenticación y cobro de las impresiones.

Para la salida de documentos, se utilizarán las impresoras instaladas en el sistema.

•Interfaces Hardware

Para la utilización de este sistema de impresión no se necesita más que ordenadores corrientes con conexión a una misma red.

Desde el punto de vista del alojamiento de la Web, se necesitará los mismos dispositivos que utiliza Windows ya que está basada en ASP.NET.

•Interfaces Software

La parte de escritorio de este sistema está desarrollado con tecnología c#.net, por tanto, se necesitará un ordenador con Windows XP o superior con FrameWork 3.5.

Para alojar la web se necesitará un equipo dotado del servidor web de Windows (ISS) 6.0 o superior.

Finalmente, el cliente también necesitará un navegador para acceder al apartado web de esta aplicación. Cualquier navegador actual puede ser válido, aunque la Web se optimizará para Internet Explorer 6 o superior, una resolución de pantalla de 1024x768 y un tamaño de fuente de sistema normal.

2.1.3.4 Restricciones de diseño

•Estándares cumplidos

El producto cumple las reglas de diseño de interfaces de usuario mediante la utilización de estilos y colores estándar, ubicación correcta y secuencial de los controles todo ello útil para asegurar la familiarización de los usuarios con la aplicación.

Como lenguaje se utiliza el castellano, con notación y comentarios concisos y fáciles de comprender, que no den lugar a ambigüedades o confusión.

•Limitaciones Hardware

Las limitaciones hardware de los equipos dotados de la aplicación cliente se reducen a las limitaciones de cualquier sistema Windows y el Hardware mínimo para manipular PDF's.

En cuanto a los equipos con el servidor de impresión, se necesita un Hardware mínimo para manipular PDF's y los drivers de la impresora a utilizar.

2.1.3.5 Atributos

•Seguridad

En cuanto a la parte de escritorio de la aplicación, la seguridad en cuanto a acceso a zonas de datos limitadas por el cliente se realiza mediante validación en la misma aplicación.

Respecto a la correcta entrada de información, la aplicación contiene sistemas de validación que facilitarán la consistencia de la base de datos del sistema instalada en el equipo servidor.

•Mantenimiento

Al tratarse en parte de una aplicación de escritorio, será necesario instalar la aplicación cliente en cada equipo que vaya a enviar trabajos a imprimir. Además, se ha de instalar una impresora virtual que genere documentos PDF de los trabajos a imprimir.

Puesto que no es necesaria ninguna instalación referente al apartado web, el mantenimiento del mismo se reduce a una posible mejora de la funcionalidad que se efectuaría en el servidor de la web.

Por otro lado, se prevé la utilización de lectores de tarjetas con chip "smart card" como método de validación lo que supondría un cambio en el código fuente de la aplicación utilizada en el servidor de impresión.

3.- Análisis

En esta fase, la de análisis del proyecto, vamos a analizar toda la aplicación a desarrollar y a describir su estructura y funcionalidad mediante diagramas que nos ayudarán a comprender cómo funciona la aplicación.

Con estos diagramas vamos a describir, de forma gráfica, cada una de las funcionalidades que lleva a cabo la aplicación (casos de uso) y cada uno de los objetos de los que se compone la misma (diagrama de clases), utilizando en ambos casos la notación UML.

3.1 Casos de uso

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de uso se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final.

•Definiciones:

•Una **relación** es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones.

•Se le llama **Actor** a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos así como a entidades abstractas como el tiempo.

Las siguientes figuras muestran los diagramas de uso de las diferentes aplicaciones que componen el proyecto. Podemos encontrar principalmente dos tipos de actores: Alumno, persona que utiliza el sistema para enviar sus trabajos a la impresora y el personal de informática de la universidad, que es una persona que utiliza el sistema para obtener estadísticas de uso y para modificar las preferencias de las impresoras.

Actualmente los alumnos acceden al sistema sin validación, mostrándoseles todos los trabajos enviados al servidor. En un futuro se prevé la utilización de las tarjetas con chip para la autenticación.

En el caso del personal, existe un password para acceder a la configuración del sistema. Este login también será sustituido por la autenticación mediante tarjeta con chip.

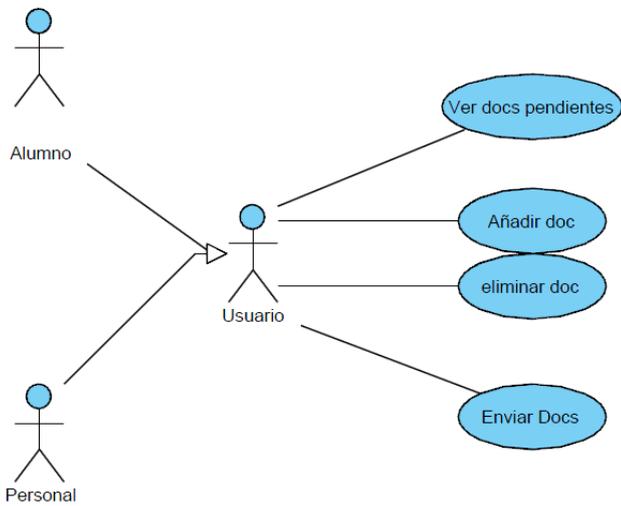


Figura 1. Aplicación Cliente

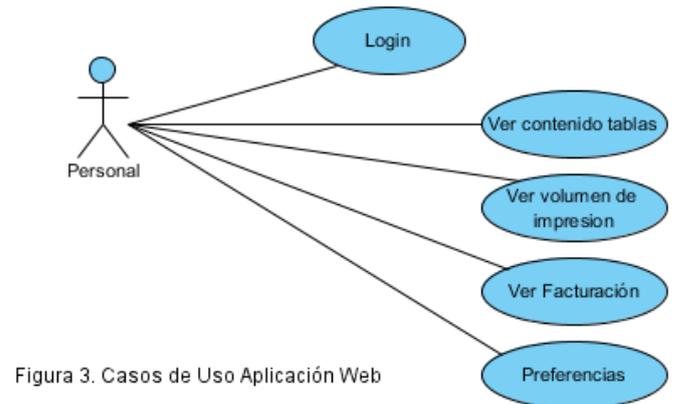
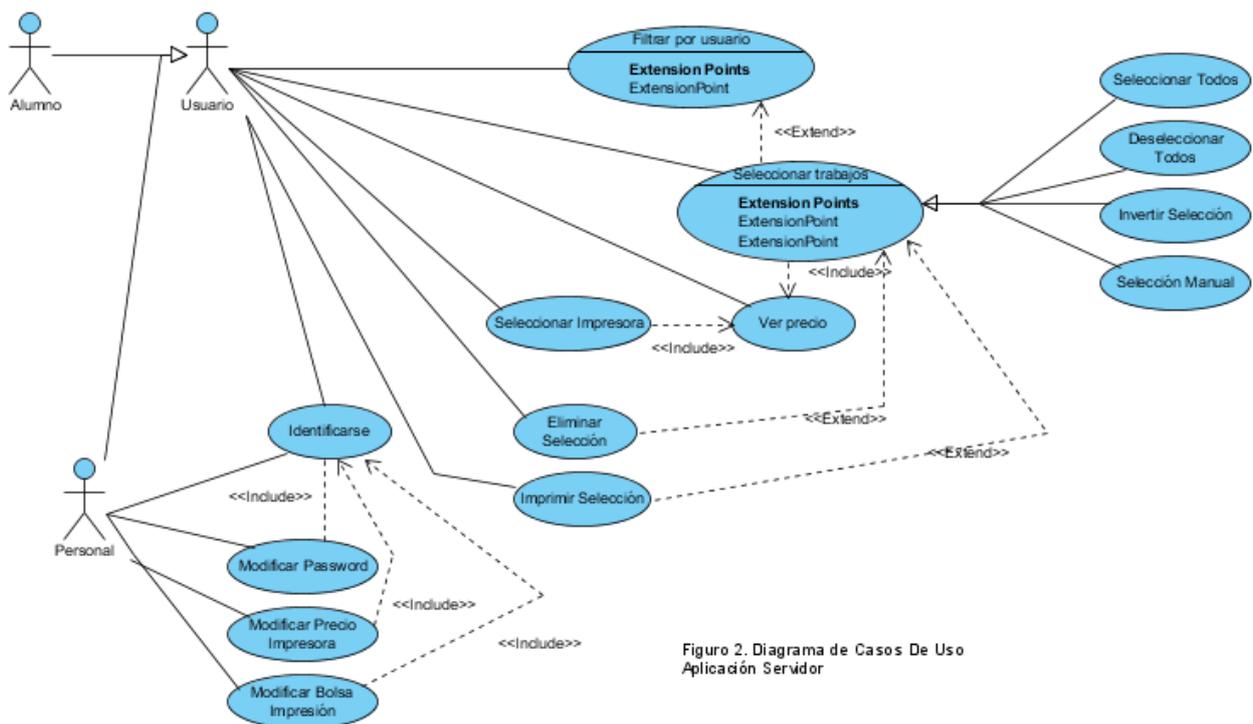


Figura 3. Casos de Uso Aplicación Web



Figuro 2. Diagrama de Casos De Uso Aplicación Servidor

3.2 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

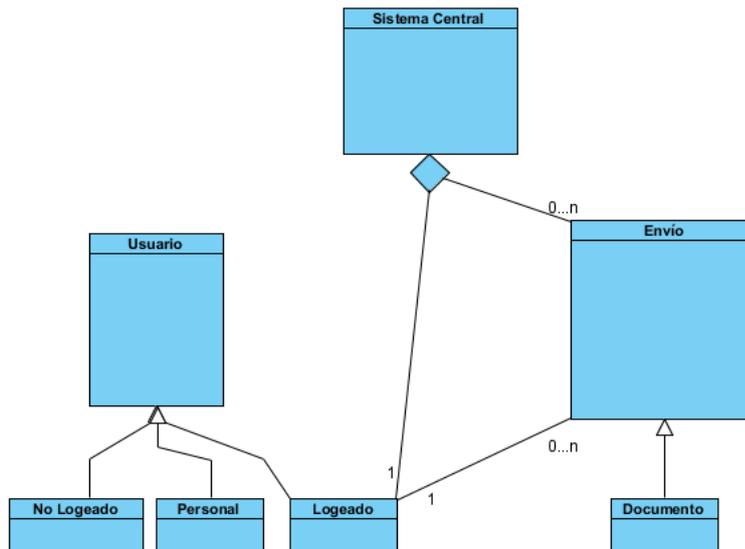


Figura 1. Diagrama de Clases

Dónde:

- Sistema central: Es la clase principal de la aplicación dónde se almacena toda la información sobre usuarios activos y envíos de trabajos de impresión. Además, contiene toda la información necesaria para la conexión con la BD.

- Usuario: Contiene el nombre del usuario y puede ser de tres tipos; usuario no logeado, un usuario logeado y un componente del personal informático que posee más permisos para llevar a cabo cambios en las preferencias, etc.

- Documento: Es la clase que almacena información sobre el documento que se quiere enviar. Véase la ruta completa, número de páginas, etc

- Envío: Es la clase que contiene toda la información sobre un documento, y además, contiene información sobre la impresora utilizada o el usuario que lo envía o ha impreso.

4.- Diseño

4.1. Estilo Arquitectónico

El estilo arquitectónico utilizado para el desarrollo de las tres aplicaciones que componen este proyecto está basado en el modelo cliente/servidor.

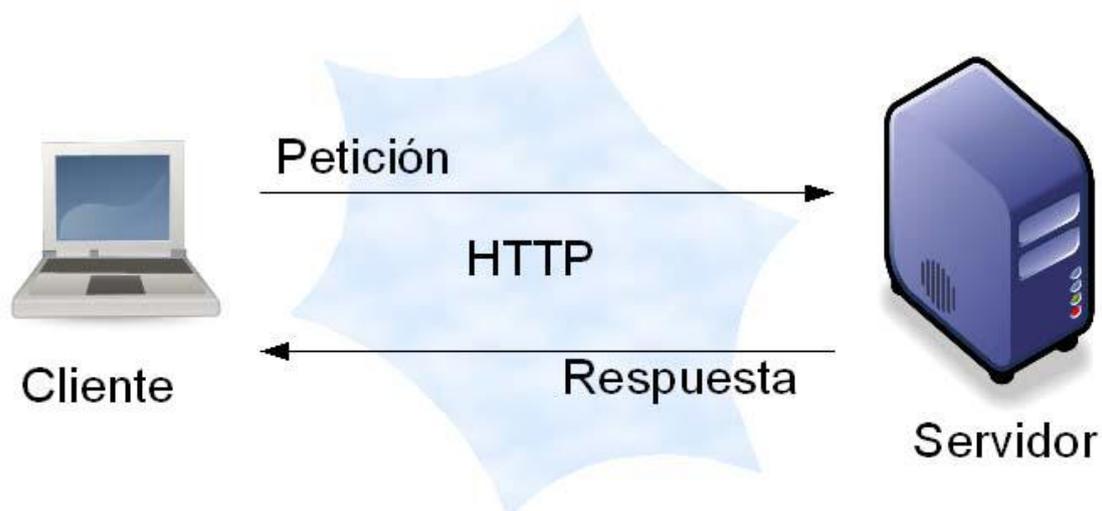
Esta arquitectura consiste básicamente en que un programa, el cliente, que realiza peticiones a otro programa, el servidor, que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan en un solo ordenador es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de ordenadores.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas *multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes ordenadores aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.



En la arquitectura de cliente/servidor el **remite de una solicitud** es conocido como cliente.

Sus características son:

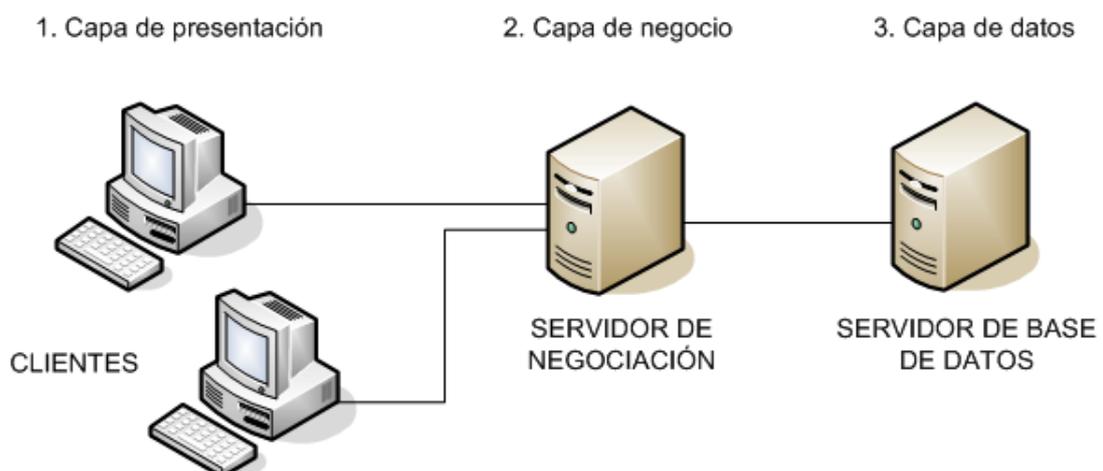
- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

En los sistemas cliente/servidor el **receptor de la solicitud** enviada por el cliente se conoce como servidor.

Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

La arquitectura del sistema se compone de 3 capas o niveles como se muestra a continuación:



- 1. Capa de presentación:** es la que ve el usuario (también denominada “capa de usuario”), presenta el sistema de usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser “amigable” (entendible y fácil de usar) para el usuario.
- 2. Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- 3. Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

A continuación se detallan un poco más cada uno de estos niveles o capas.

4.2. Nivel de Presentación

Definición: “Componentes software que implementan la interacción con los usuarios a través de una representación visual de la aplicación, proporcionando a los usuarios una forma de acceder y controlar los datos y los servicios de los objetos”.

Para implementar la Interfaz de Usuario se ha utilizado el lenguaje c#.net y framework 3.5 acompañado de algún componente externo para la representación de datos del apartado web de la aplicación.

La capa de presentación, es la encargada de interactuar con el usuario y en el caso concreto de este proyecto está formado por el conjunto de formularios o ventanas que constituyen las aplicaciones de escritorio y por otro lado las páginas web a través de las cuales se muestra información al usuario.

Para profundizar un poco más en la descripción de dicho nivel, se va a utilizar el modelo de navegación, el cual representa la semántica navegacional de una aplicación Web en base a un Modelo de Objetos y en requisitos de navegación.

Este modelo utiliza una notación basada en UML y considera el punto de vista de cada perfil de usuario identificado previamente en el Modelo de Objetos.

El modelo de navegación define y estructura el acceso al sistema para los diferentes tipos de usuario mediante los **Diagramas de Usuario** y los **Mapas Navegacionales** y además introduce primitivas de abstracción que permiten:

- Representación de la navegación
- Especificación de búsquedas
- Tratamiento de la visualización de la información (presentación)
- Ejecución de Servicios
- Personalización de información de los distintos usuarios

4.2.1. Mapas Navegacionales

Los mapas navegacionales son diagramas de navegación que nos permiten describir los aspectos de la interfaz de usuario tales como la misma presentación de pantallas o la navegación a través de las páginas Web. En ellos se especifican las rutas o caminos de acceso a la información para cada uno de los usuarios que interactúan con el sistema.

Un mapa navegacional está compuesto por contextos navegacionales y vínculos navegacionales.

Cada contexto especifica una vista para el usuario. Las clases que aparecen en el interior hacen referencia a qué partes de las clases descritas en el diagrama de clases serán visibles por los usuarios.

Una clase navegacional se define como una proyección (vista) que se especifica sobre una clase del Modelo de Objetos. Incluye un dado subconjunto de atributos y métodos.

Los vínculos navegacionales indican la navegación entre contextos (relación de alcanzabilidad entre contextos de navegación). Esta navegación se define a partir de relaciones navegacionales definidas dentro de los contextos.

Para empezar, es necesario detectar los distintos tipos de usuarios que harán uso tanto de las aplicaciones escritorio como la aplicación Web y las relaciones existentes entre ellos.

Los usuarios serán los siguientes:

- Alumno (A)
- Personal Informático (PI)



Figura 4: Diagrama de usuarios

Los mapas navegacionales se han realizado utilizando la metodología OOWS. En ella podemos encontrar agentes (usuarios que realizan la navegación), contextos de navegación y los vínculos que existen entre ambos. Los contextos de exploración (E) son aquellos a los que se puede acceder desde cualquier punto de la aplicación y los contextos de secuencia (S), en cambio, sólo pueden ser accedidos habiendo explorado antes otros contextos.

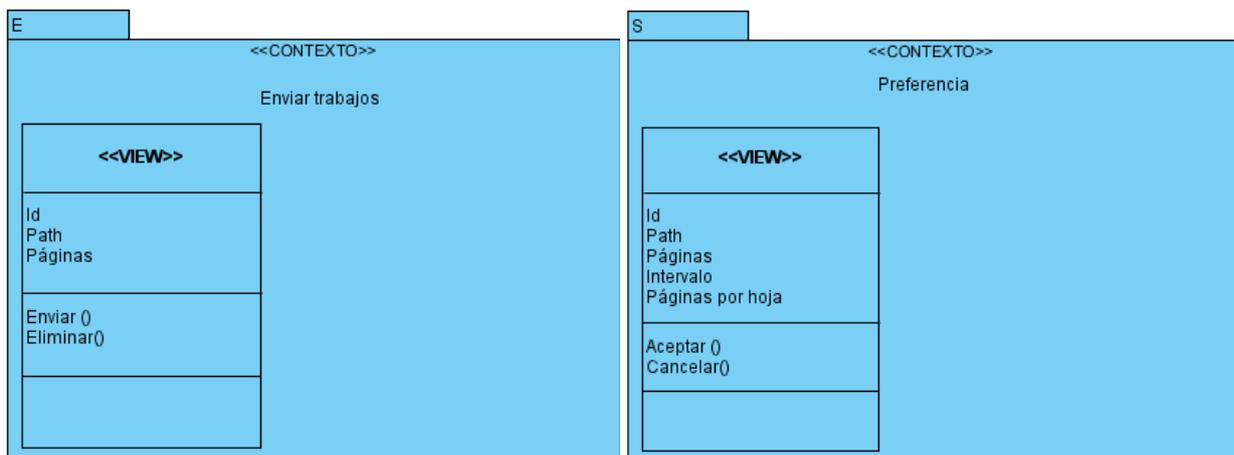
A continuación, se procederá con la descripción de los mapas navegacionales asociados a cada uno de los usuarios mostrados en la Figura 5.

Para la aplicación cliente tanto el alumno como el personal de informática tienen las mismas posibilidades:



Figura 6: Mapa Navegacional para la aplicación cliente

Ahora, se procede a la descripción más detallada de cada uno de los contextos que aparecen en la imagen anterior:



Los escenarios para la aplicación cliente son los siguientes:

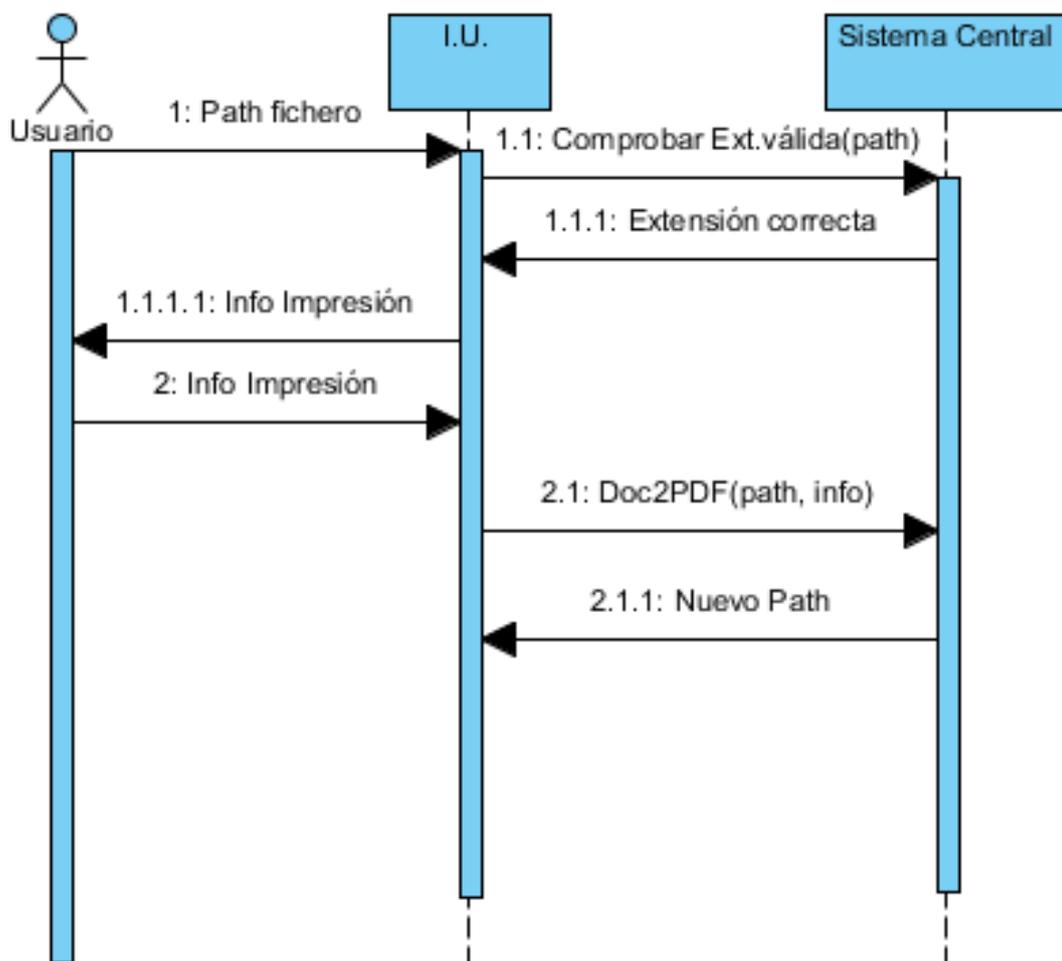
Escenario 1: Añadir un trabajo de OFFICE mediante Drag & Drop

Descripción General: El usuario añade un trabajo a la lista de trabajos pendientes.

Pasos:

1º- El usuario ha de posicionarse en la carpeta dónde se haya el documento que desea imprimir.

2º- El usuario arrastra el fichero que desea imprimir a la lista que aparece en la ventana principal del programa de impresión.



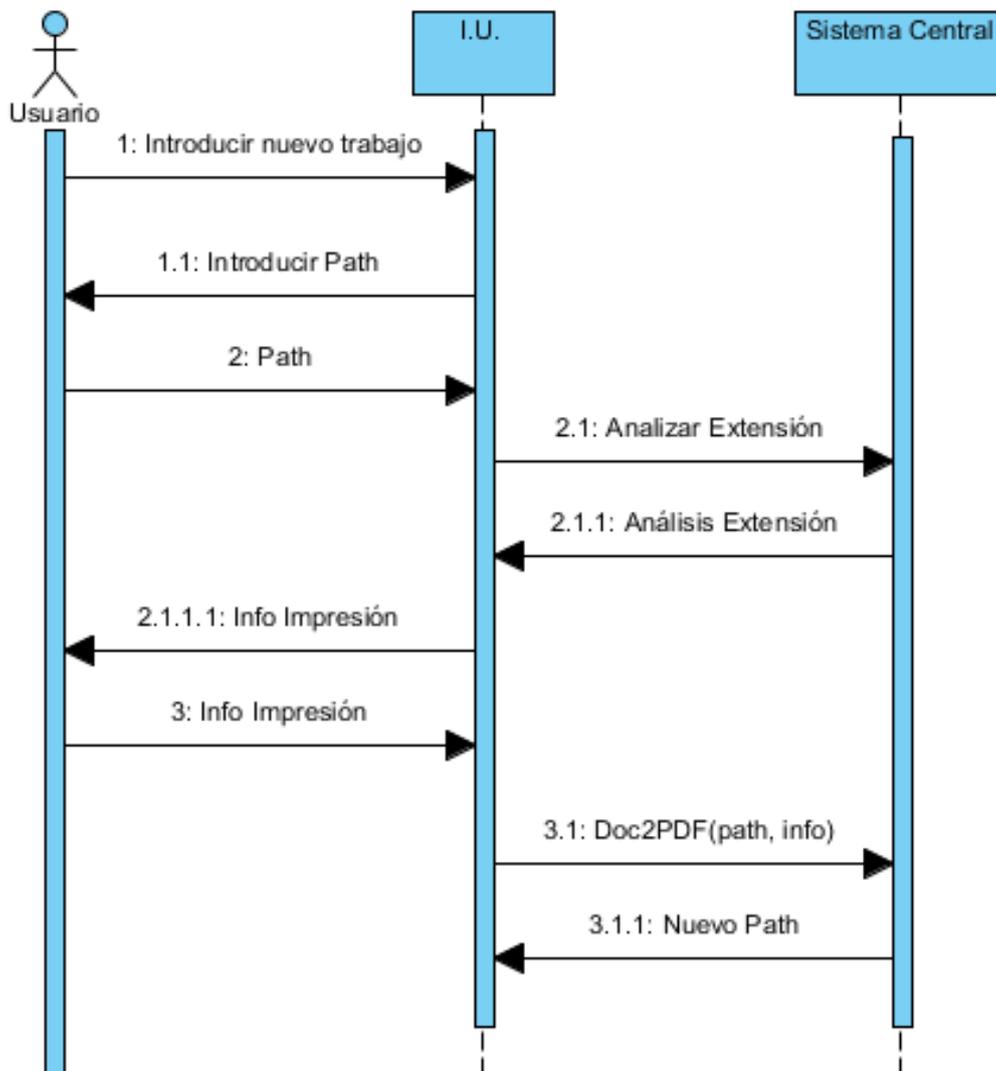
Escenario 2: Añadir un trabajo de OFFICE mediante “openFileDialog”

Descripción general: El usuario añade un trabajo a la lista de trabajos pendientes.

Pasos:

1º- El usuario hace click sobre el botón “Añadir”.

2º- En el diálogo que aparece, el usuario tendrá que buscar el archivo y tras seleccionarlo, hacer click en el botón aceptar para así añadirlo a la lista de trabajos pendientes.



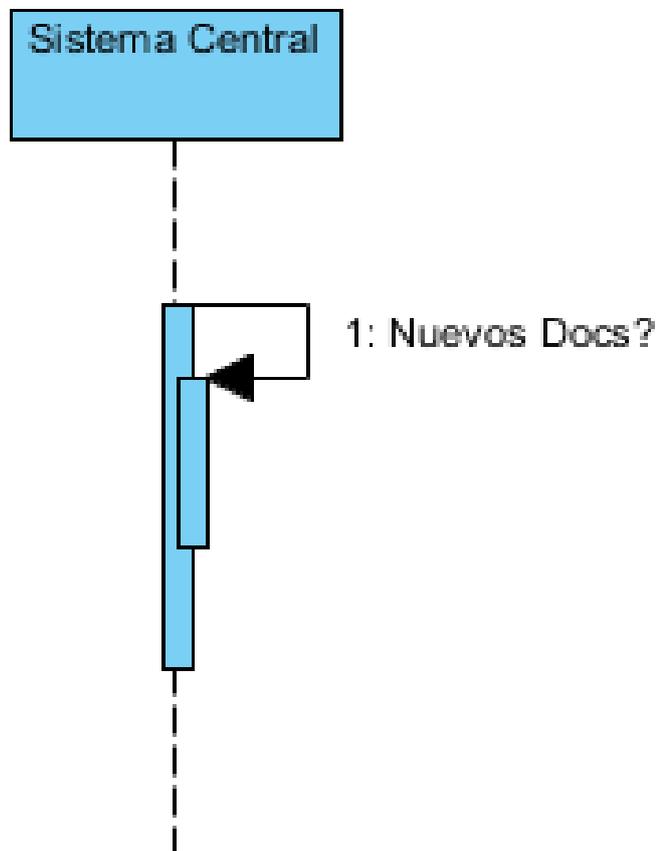
Escenario 3: Añadir un trabajo de CUALQUIER ORIGEN

Descripción general: El usuario añade un trabajo a la lista de trabajos pendientes desde otras aplicaciones.

Pasos:

1º- El usuario manda un trabajo a imprimir desde la misma aplicación con la que lo ha generado.

2º- La aplicación lo capturará y en vez de imprimirlo lo mostrará en la lista de documentos pendientes.



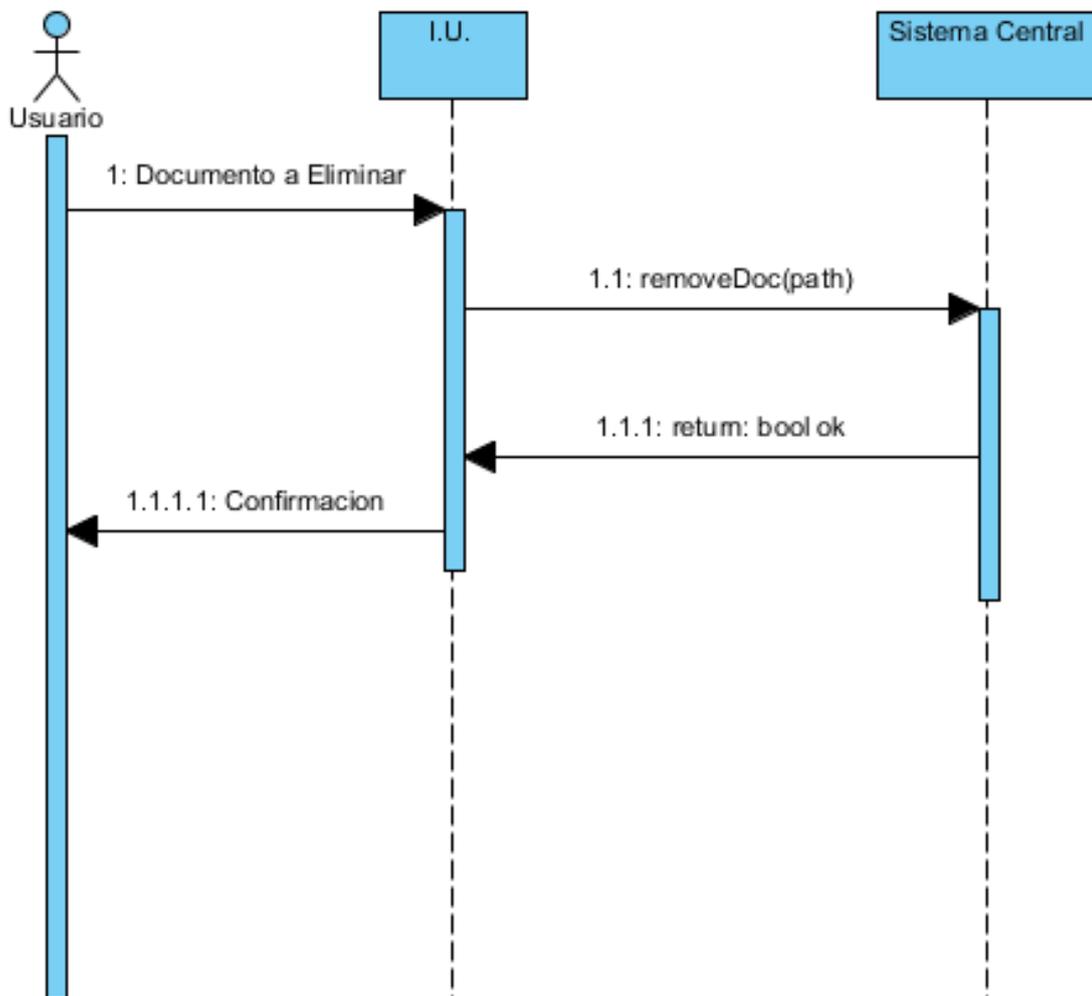
Escenario 4: Eliminar un trabajo de la lista

Descripción general: El usuario borra un trabajo de la lista para no enviarlo al servidor.

Pasos:

1º- El usuario selecciona el trabajo a eliminar de entre todos los que conforman la lista de documentos pendientes.

2º- Con el trabajo de la lista seleccionado, el usuario hace click sobre el botón "Eliminar" y el documento será eliminado de la lista.



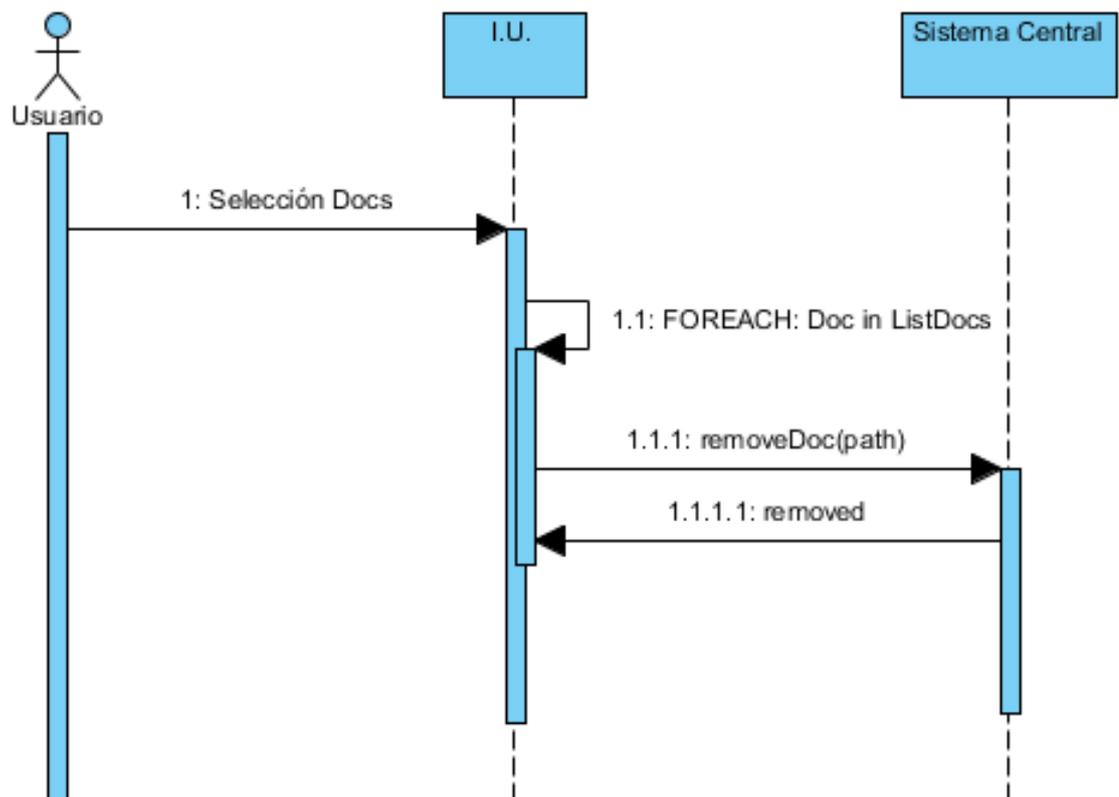
Escenario 5: Enviar documentos al servidor de impresión

Descripción general: El usuario envía los documentos de la lista de trabajos pendientes al servidor de impresión.

Pasos:

1º- El usuario hace click sobre el botón “Enviar”.

2º- Aparecerá un diálogo mostrando el número de trabajos que se van a enviar y pidiendo confirmación. El usuario deberá hacer click sobre “Aceptar” o “Cancelar” para confirmar o no la operación respectivamente.



Escenario 6: Salir de la aplicación

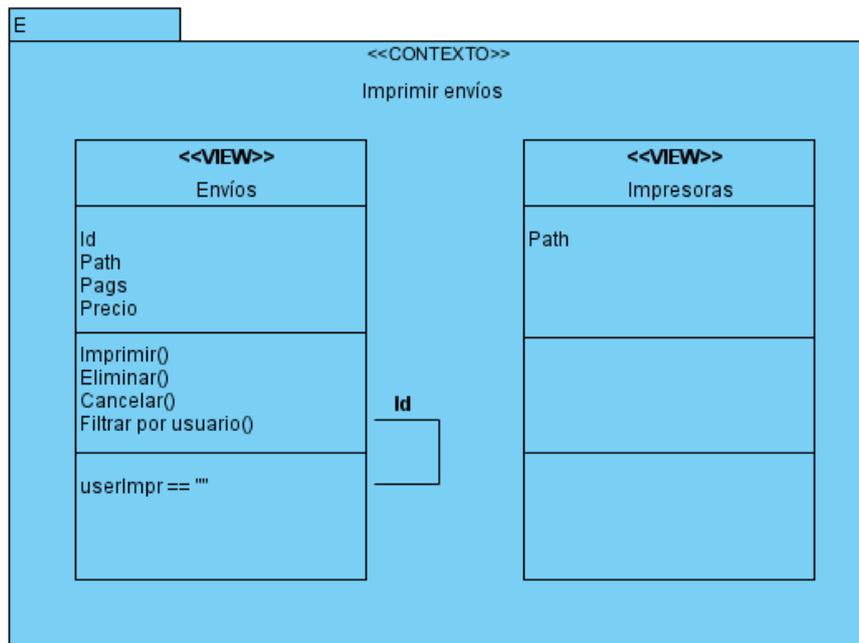
Descripción general: El usuario cierra la ventana principal de la aplicación cliente cerrando así la aplicación.

Pasos:

1º- El usuario hace click sobre el botón "Cerrar".

2º- Aparecerá un diálogo mostrando el número de trabajos pendientes de enviar y pidiendo confirmación. El usuario deberá hacer click sobre "Aceptar" o "Cancelar" para confirmar o no la operación respectivamente.

Ahora, se presentarán los mapas navegacionales para la aplicación servidor, esta vez desglosados por los diferentes usuarios ya que tienen diferente funcionalidad, empezando por el Alumno:

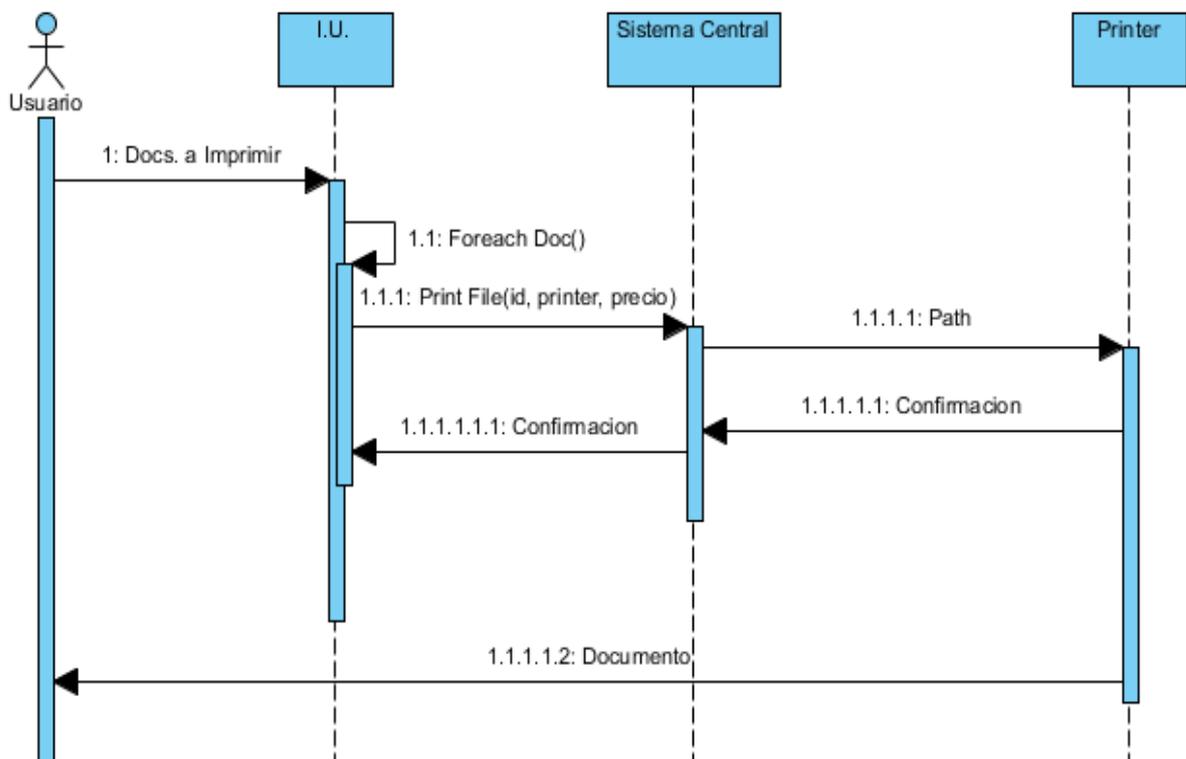


Escenario 7: Imprimir envíos

Descripción general: El usuario ordena la impresión de los documentos.

Pasos:

- 1º: El usuario ha de seleccionar marcando el checkbox correspondiente a cada trabajo de impresión.
- 2º: El usuario ha de hacer click sobre el botón imprimir.
- 3º: Se pedirá confirmación antes de proceder. Para enviar definitivamente los trabajos a la cola de impresión, el usuario deberá aceptar.



Escenario 8: Eliminar trabajos de la lista

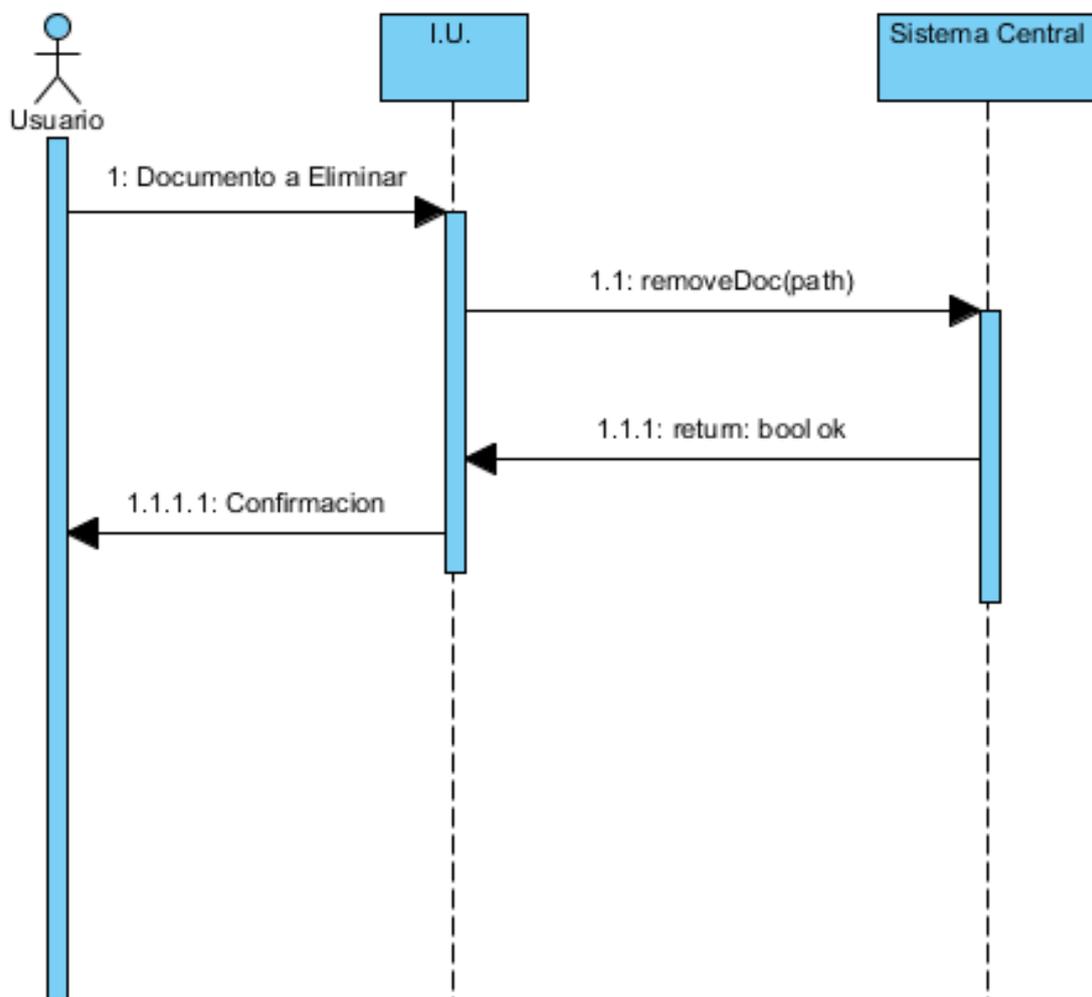
Descripción general: El usuario decide eliminar ciertos trabajos de la lista.

Pasos:

1º: El usuario ha de seleccionar marcando el checkbox correspondiente a cada trabajo de impresión.

2º: El usuario ha de hacer click sobre el botón “Eliminar selección”.

3º: Se pedirá confirmación antes de proceder. Para eliminar definitivamente los trabajos de la lista, el usuario deberá aceptar.



Escenario 9: Gestión de la selección de trabajos

Descripción general: El usuario puede gestionar la selección de trabajos rápidamente gracias a los botones de gestión rápida.

Pasos:

1º: El usuario puede realizar rápidamente, gracias a los botones destinados a ello las siguientes acciones:

- Seleccionar todo
- Deseleccionar todo
- Invertir selección

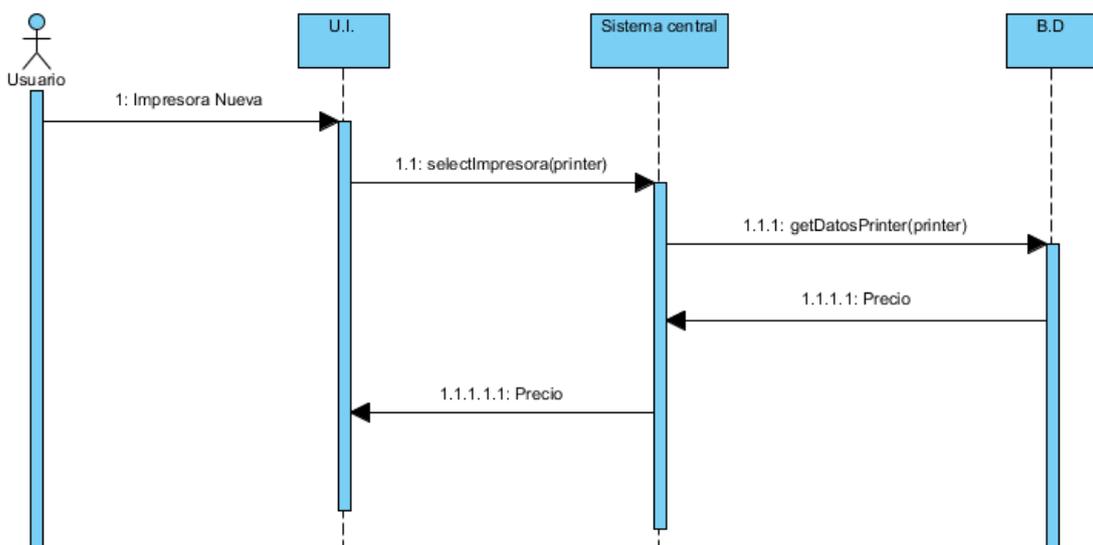
Escenario 10: Cambiar la impresora seleccionada.

Descripción general: El usuario puede elegir cuál será el dispositivo que imprimirá sus trabajos basándose en la ubicación y precio.

Pasos:

1º: El usuario deberá hacer click sobre el botón “cambiar...” situado en la parte superior derecha del formulario.

2º: Acto seguido aparecerá un combo box listando todas las impresoras disponibles. El usuario deberá seleccionar al que mejor se adecúe a sus necesidades.



Ahora, para el personal de informática...

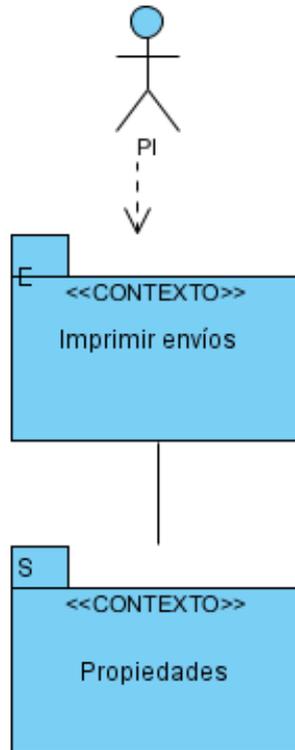
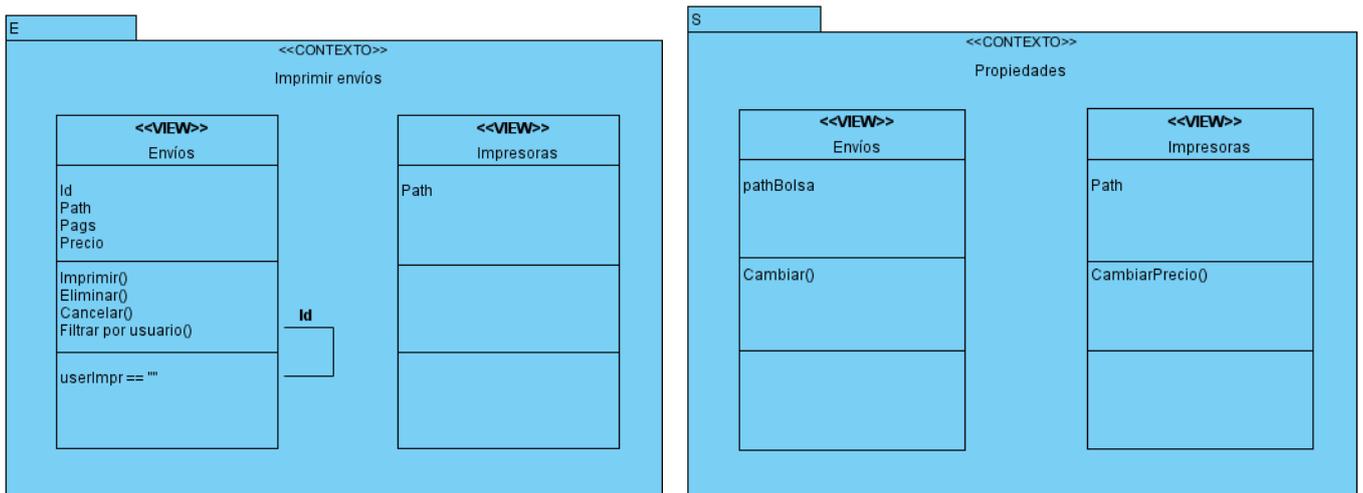


Figura 8: Mapa Navegacional del PI para la aplicación del servidor



Escenario 11: Cambio bolsa de impresión.

Descripción general: Un usuario con derechos de administrador puede cambiar la ruta de la bolsa de impresión.

Pasos:

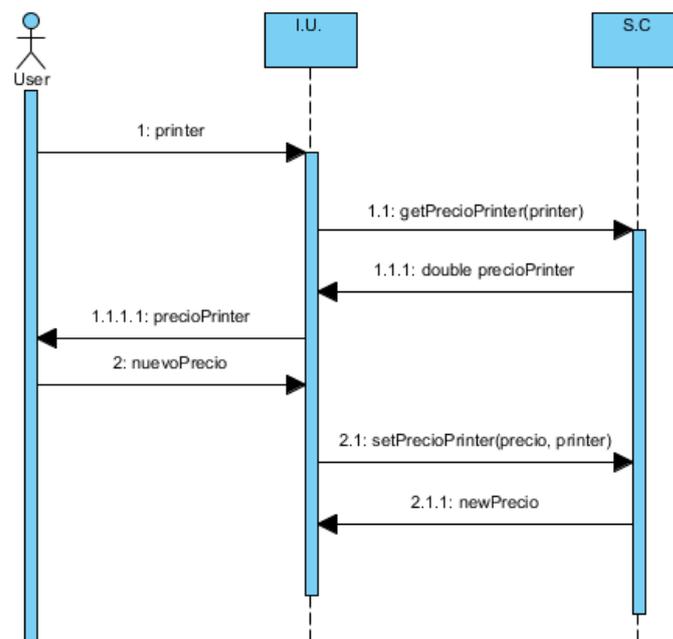
- 1º: Hacer click en el apartado “Herramientas” del menú superior de la ventana principal.
- 2º: Introducir la contraseña de administrador.
- 3º: Introducir la nueva ruta de la nueva bolsa de impresión ya sea a mano o mediante el botón “Explorar...”.

Escenario 12: Editar precio de las impresoras

Descripción general: El usuario puede editar el precio de las impresiones para cada impresora instalada en el sistema.

Pasos:

- 1º: Hacer click en el apartado “Herramientas” del menú superior de la ventana principal.
- 2º: Introducir la contraseña de administrador.
- 3º: Seleccionar la impresora deseada de la lista y hacer click sobre el botón “Editar precio”.
- 4º: Introducir el precio en el textbox que aparece y hacer click sobre el botón aceptar.



Escenario 13: Cambiar contraseña de administrador

Descripción general: El usuario tiene la posibilidad de cambiar el password requerido para acceder al apartado de administración y herramientas.

Pasos:

1º: Hacer click en el apartado “Herramientas” del menú superior de la ventana principal.

2º: Introducir la contraseña de administrador.

3º: Introducir la nueva contraseña en el apartado indicado y hacer click sobre el botón aceptar.

Finalmente, se van a presentar los mapas navegacionales para la aplicación web de este sistema. En este caso, solamente se hablará del personal informático ya que el alumnado no tiene acceso a este apartado.

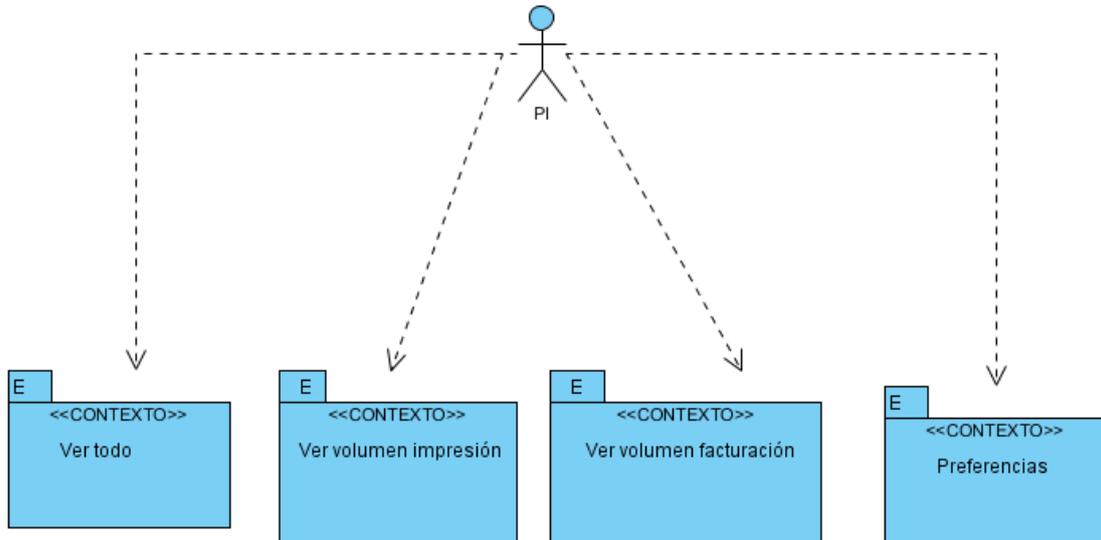
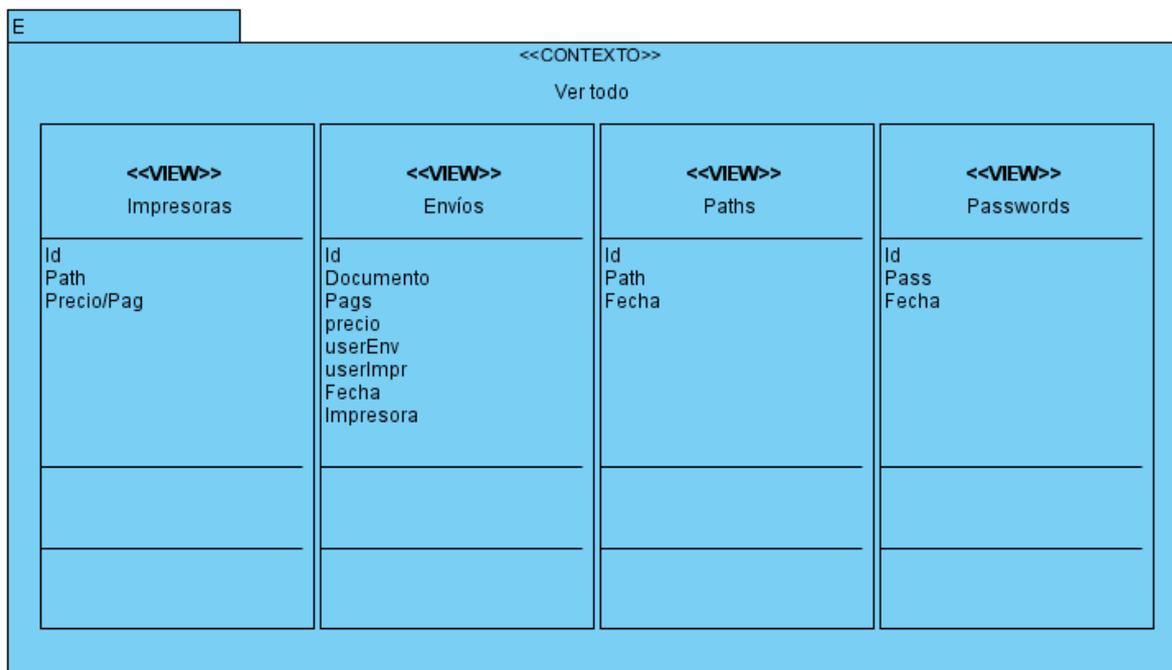
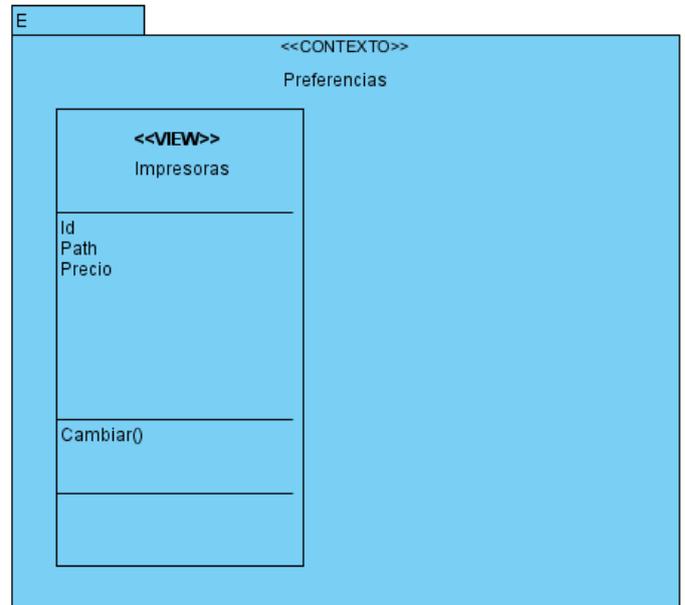
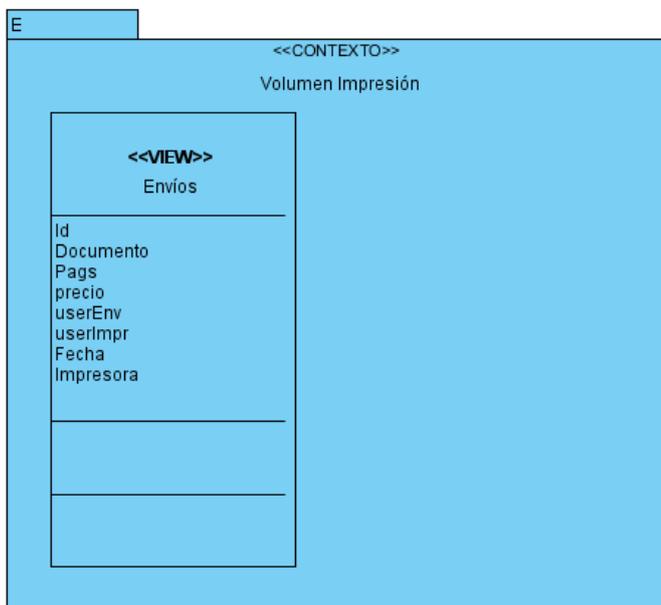
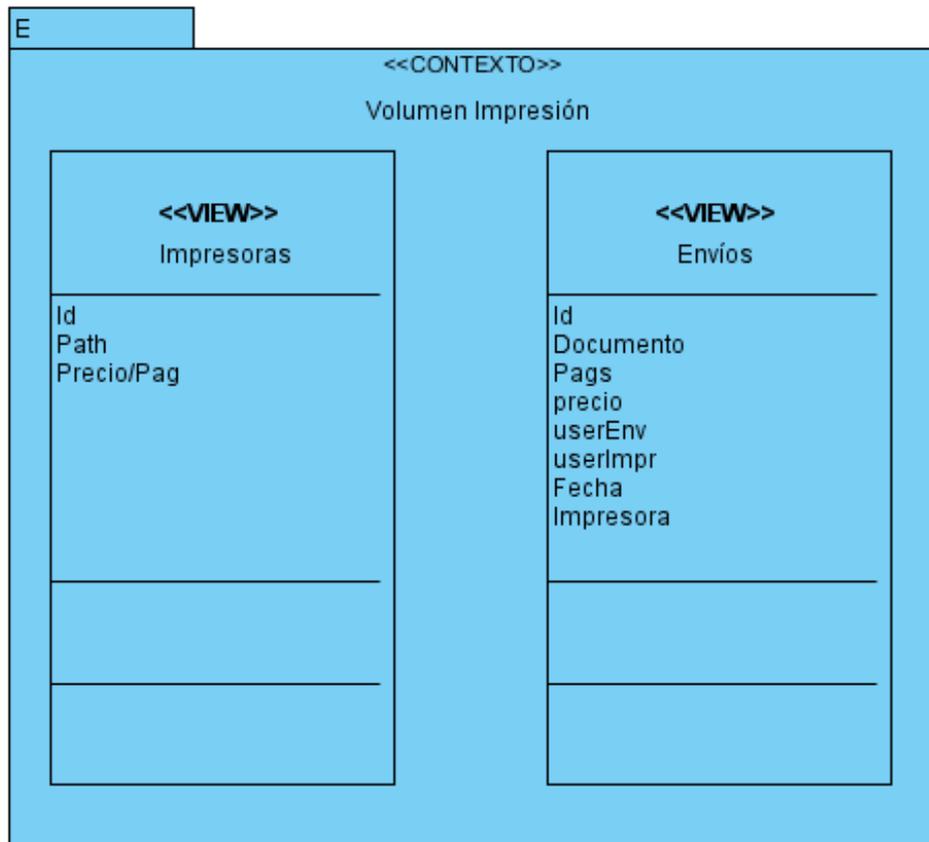


Figura 9: Mapa Navegacional del PI para la aplicación Web

Y con algo más de detalle...





4.3 Nivel Lógico

El nivel lógico es el encargado de comunicar el nivel de presentación con el nivel de persistencia. Desde el nivel de presentación se solicita la información necesaria para mostrarla al usuario, de manera que el nivel lógico se encarga de procesar dicha solicitud y conectar con el nivel de persistencia para obtener la información necesaria, procesarla y enviarla al nivel de presentación para que la muestre al usuario.

El nivel lógico está formado por todos aquellos componentes software que implementan la funcionalidad descrita. Estos, se componen de funciones que se encargan de llevar a cabo las operaciones necesarias para resolver las solicitudes del nivel anterior.

Tanto en las aplicaciones escritorio cliente y servidor como en la aplicación Web podemos encontrar ciertos componentes en común:

- Archivos de conexión a la BD: útiles para la conexión con la base de datos de la aplicación y se encarga de la solicitud de la información requerida en las operaciones.
- Archivos de configuración: contienen la configuración de la aplicación.
- Archivos de funciones: contienen la funcionalidad de la aplicación.

Además, la aplicación Web contiene:

- Archivos CSS: contienen los distintos estilos que se utilizan en la aplicación.
- Componente APN Soft DataGrid: Componente del estilo DataGrid de Microsoft útil para representar el contenido de una tabla de una BD en una lista de manera más eficiente.

4.4 Nivel de persistencia

El nivel de persistencia es el nivel que se encarga de gestionar la información de la aplicación. Para describir dicho nivel se muestra a continuación el modelo de datos entidad relación.

4.4.1 Modelo de datos entidad-relación

Un **diagrama o modelo entidad-relación** es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

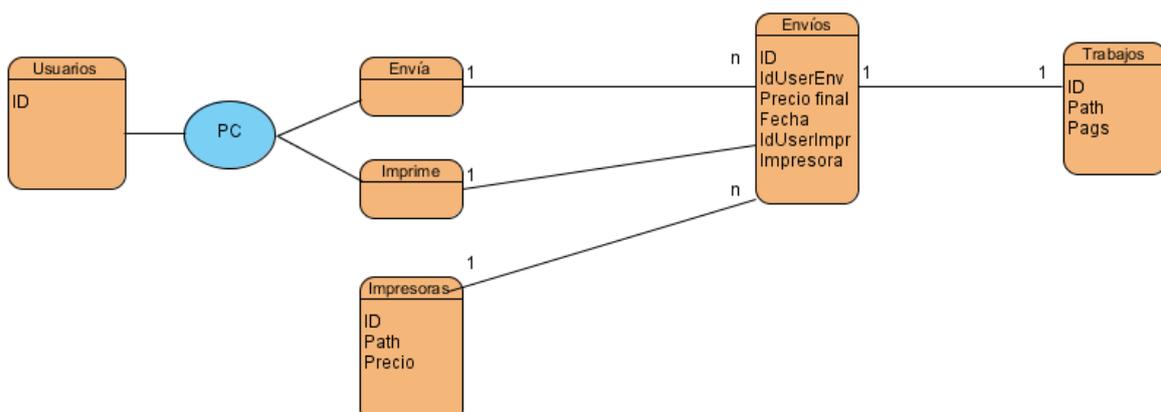
El término **entidad**, es cualquier objeto sobre el que se tiene información. Una entidad está descrita por sus características. Por ejemplo, la entidad Envío lleva consigo las características de: Id, Documento, UserEnv, UserImpr, Precio, etc. Se representa mediante un rectángulo o “caja” etiquetada en su interior mediante un identificador.

Una **relación** describe cierta dependencia entre entidades. Se representa mediante un rombo etiquetado en su interior con un verbo, este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona.

Los **atributos** son propiedades relevantes propias de una entidad y/o relación. Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior.

Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

En la siguiente figura, se muestra el diagrama Entidad-Relación para la aplicación desarrollada:



5. - Implementación

5.1 Patrones de traducción

Llamamos patrones de traducción al conjunto de reglas que se utilizan para “traducir” todo lo especificado en la fase de diseño, a la fase de implementación.

Los mapas navegacionales definidos en el diseño de la aplicación, se traducen aplicando los siguientes patrones de traducción:

Alumno

·**Mapa Navegacional:** Define el modo de acceder a cada uno de los formularios o páginas web de la aplicación referente a cada tipo de usuario.

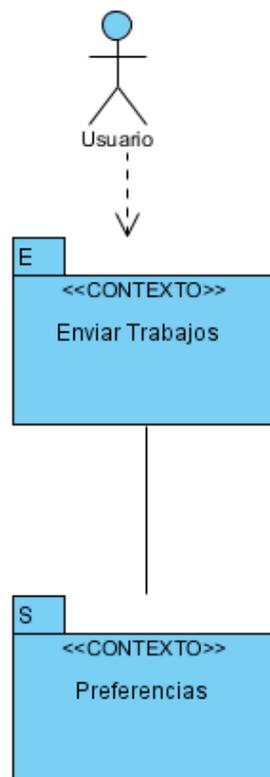


Figura 15: Mapa Navegacional de “Alumno”.

Como se puede observar en la figura anterior, el Alumno puede acceder en cualquier momento al contexto de exploración (E) y a los contextos de secuencia (S) a través de los correspondientes contextos de exploración.

A continuación, se va a mostrar el resultado de la traducción de los contextos que aparecen en la figura anterior:

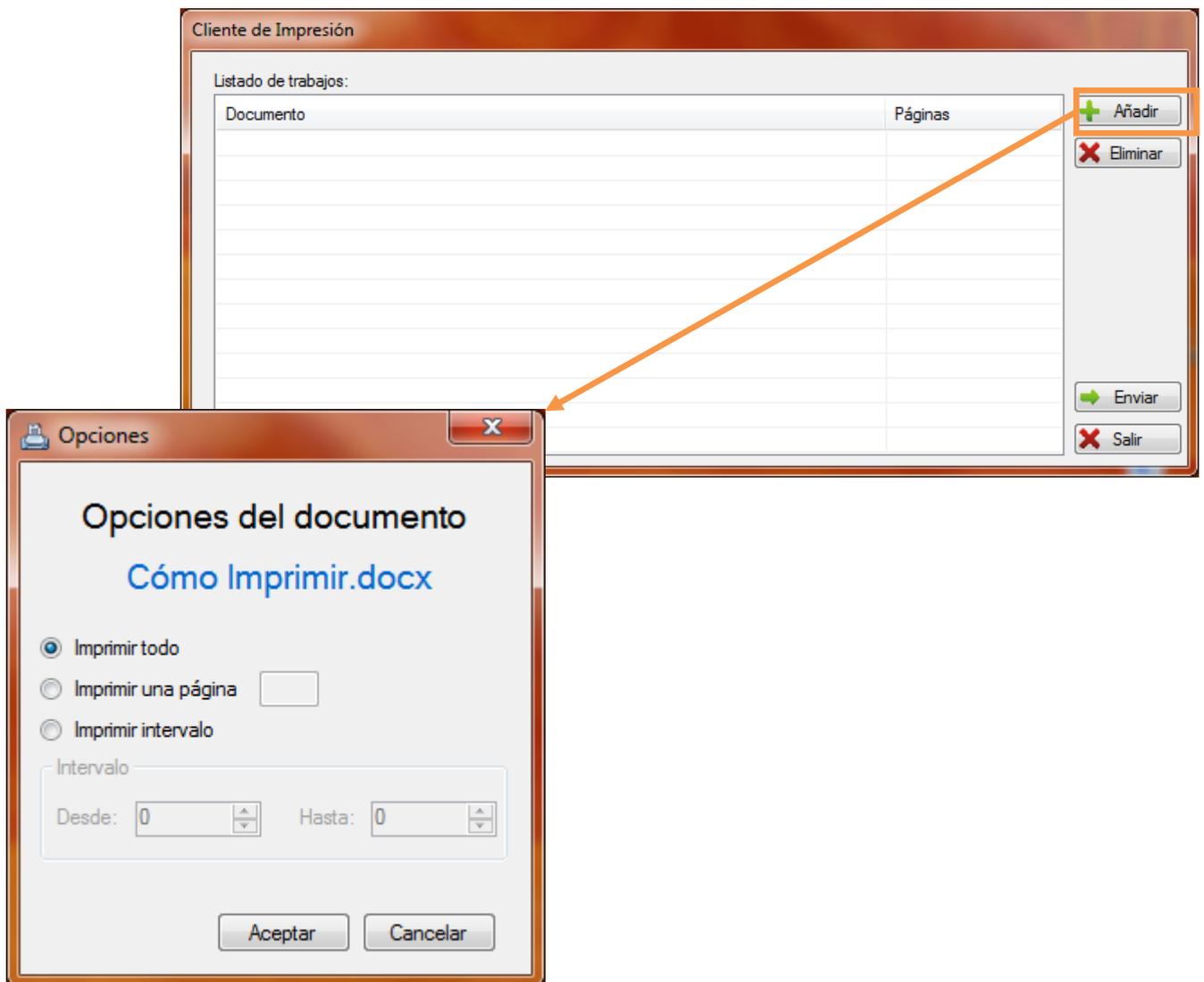


Figura 16: Añadir trabajo a la lista

·**Contexto Navegacional:** Cada contexto representado en el diagrama navegacional se traduce en un formulario para las aplicaciones de escritorio y en una página Web para la aplicación Web.

Los contextos de exploración (E) estarán siempre accesibles desde cualquier otro formulario o página Web. Por el contrario, los contextos de secuencia (S) son aquellos que requieren navegar a través de otros contextos para llegar a ellos.

·**Vistas de los contextos:** Cada vista representada dentro de los contextos navegacionales se representa como una clase de la base de datos, en la que sólo se muestra la información necesaria. Dentro de cada contexto pueden haber varias vistas y cada una de ellas tener que cumplir una serie de condiciones, ofrecer diferentes métodos, etc. Una vista se traduce en una lista de información que aparecerá en los formularios o páginas Web, entendiéndose las condiciones como la información que aparecerá en dicha lista y los métodos, en la funcionalidad que se le ofrecerá al usuario.

Tal y como se puede apreciar en la figura 6, dónde se representa el contexto “Enviar Trabajos” para cualquier usuario se ofrecen dos métodos: Enviar (), Eliminar ().

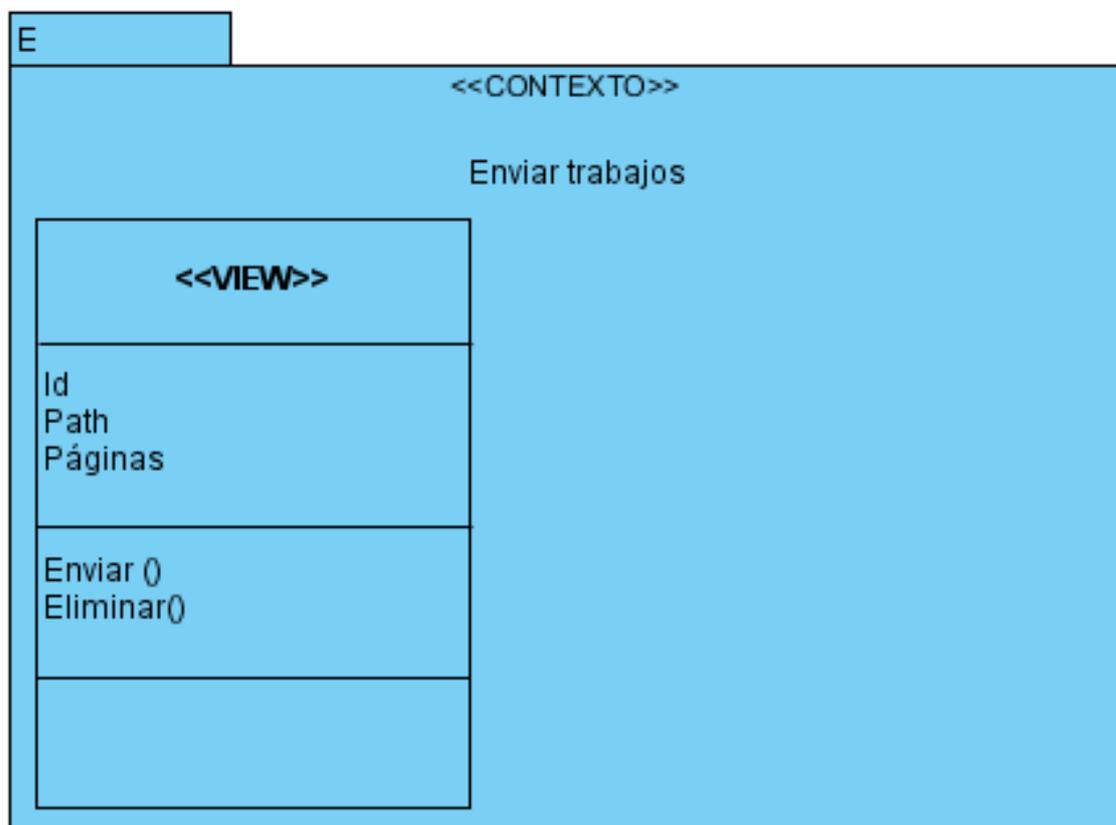


Figura 17: Enviar trabajos

Los métodos que se pueden observar en la figura anterior (Añadir (), Eliminar (), Enviar (), Salir ()) son los que luego aparecerán implementados en el formulario correspondiente tal y como se muestra en la siguiente figura:

The screenshot shows a window titled "Cliente de Impresión". Inside, there is a section labeled "Listado de trabajos:" containing a table with two columns: "Documento" and "Páginas". The table is currently empty. To the right of the table, there are four buttons: "Añadir" (with a green plus icon), "Eliminar" (with a red X icon), "Enviar" (with a green right-pointing arrow icon), and "Salir" (with a red X icon). The "Añadir" and "Eliminar" buttons are highlighted with an orange border.

Figura 18: Formulario principal Cliente

5.2 Esquema de clases

En este apartado, se va a mostrar las clases que permiten el funcionamiento de la aplicación, diferenciando entre los tres niveles arquitectónicos de la misma.

5.2.1 Nivel de presentación

El nivel de presentación provee a la aplicación una interfaz de usuario en la que el sistema presenta la información a los usuarios y acepta entradas o respuestas del usuario para usar la aplicación.

Aplicación Cliente

En la siguiente figura, se muestra todas aquellas clases y distintos tipos de archivos (imágenes, hojas de estilo etc.) que forman el nivel de presentación de la aplicación cliente.



Figura 19: Nivel de presentación de la aplicación cliente.

En la siguiente figura, se muestra todas aquellas clases y distintos tipos de archivos (imágenes, hojas de estilo etc.) que forman el nivel de presentación de la aplicación servidor.

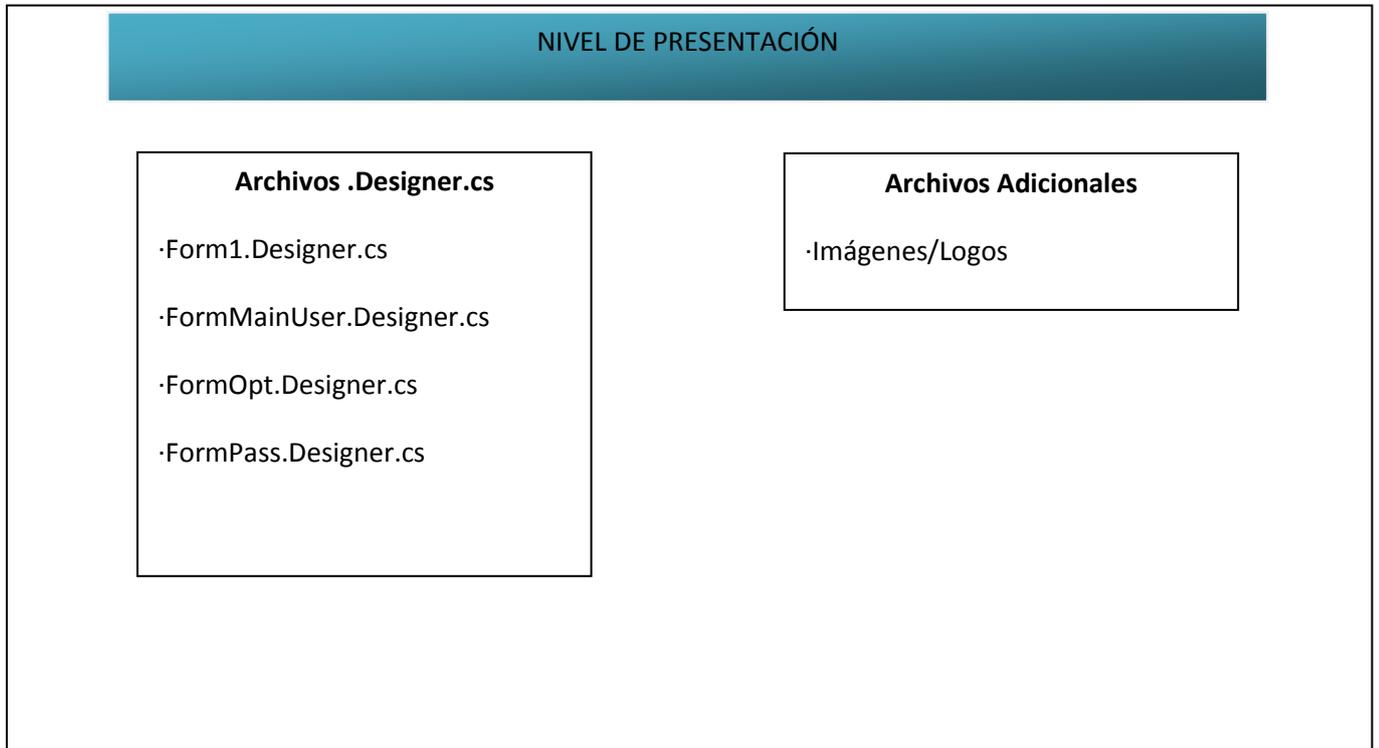


Figura 20: Nivel de presentación de la aplicación servidor.

En la siguiente figura, se muestra todas aquellas clases y distintos tipos de archivos (imágenes, hojas de estilo etc.) que forman el nivel de presentación de la aplicación Web.

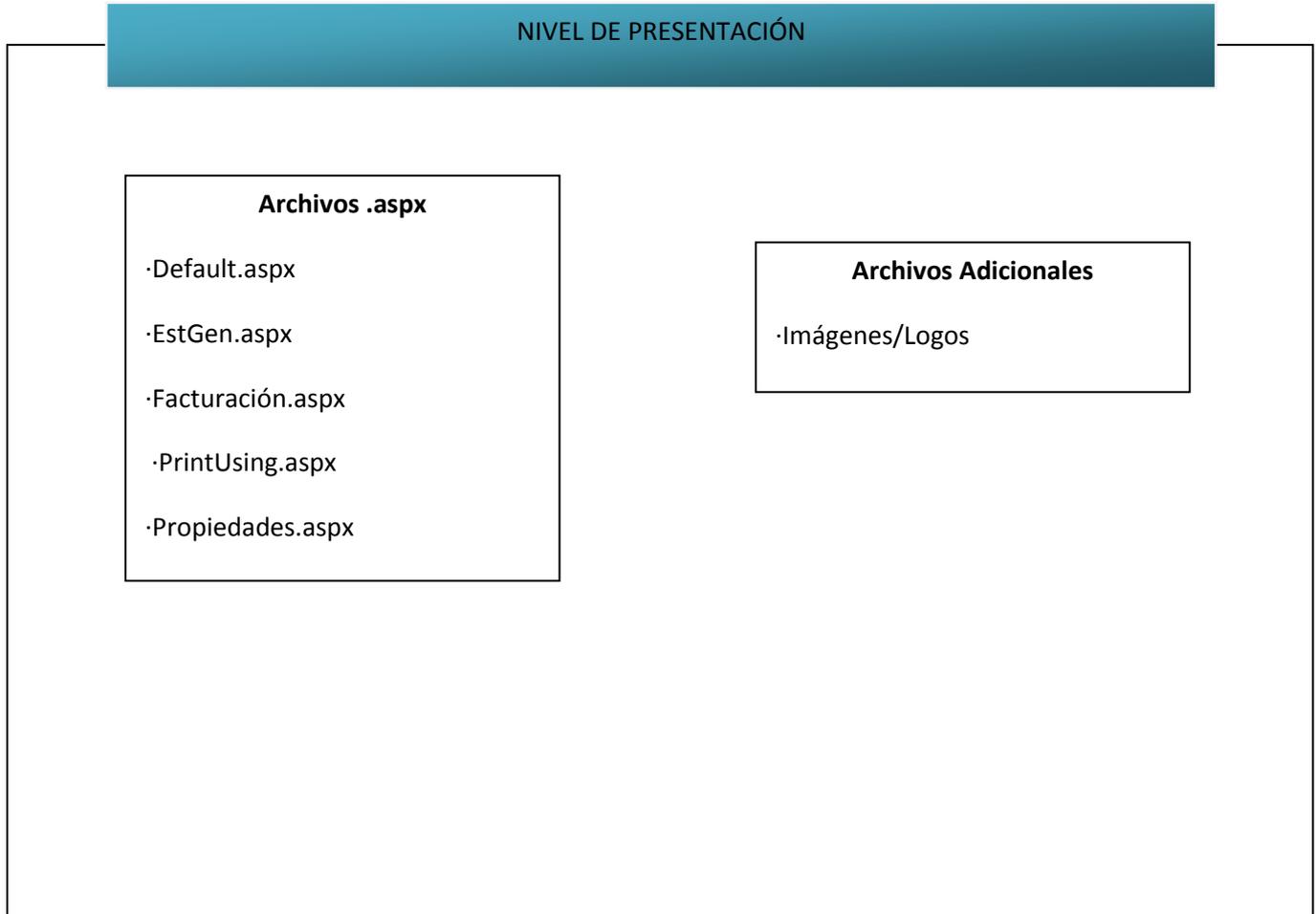


Figura 21: Nivel de presentación de la aplicación Web.

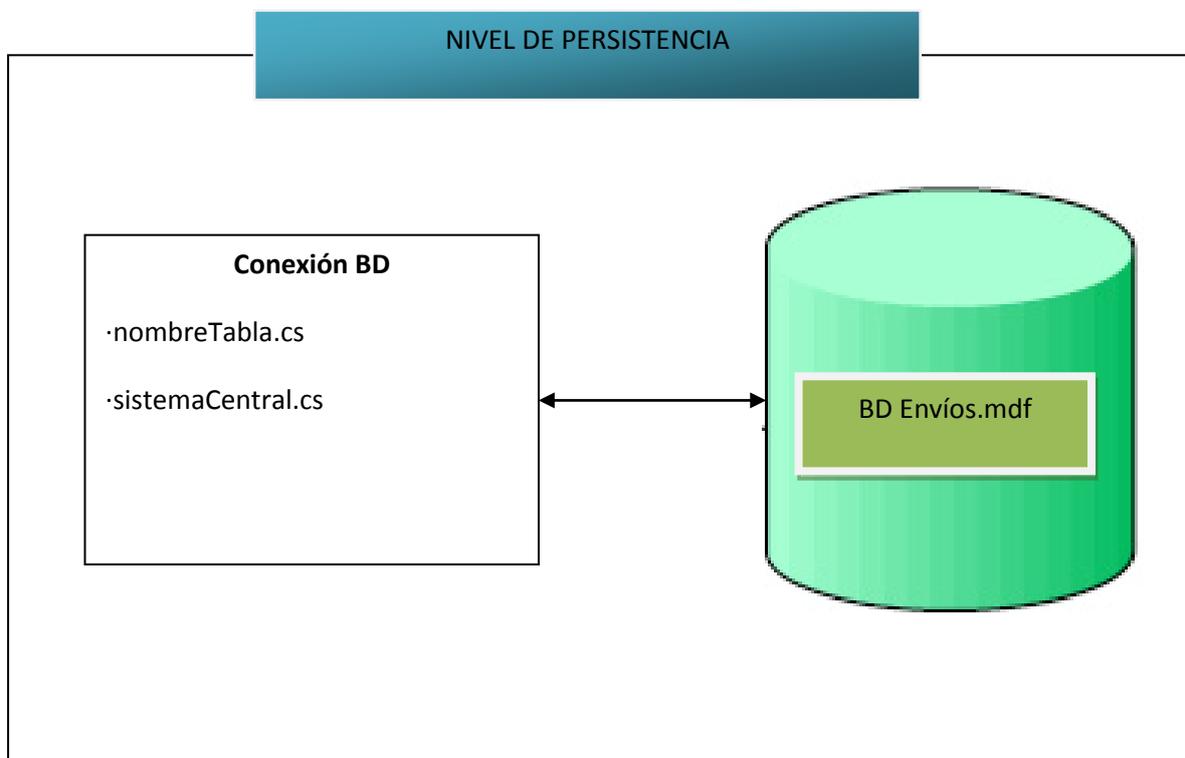
5.2.2 Nivel Lógico

El nivel lógico es aquel que guarda toda la funcionalidad de las entidades que se han definido en el sistema y aquellas que hacen de intermediarias entre el nivel de presentación y el nivel de persistencia de datos.

En el caso concreto de este proyecto, para cada aplicación existe una clase denominada "Sistema Central" que es la encargada tanto de realizar operaciones con la información obtenida de la capa de presentación y de recuperar datos de la capa de persistencia.

5.2.3 Nivel de Persistencia

El nivel de persistencia es aquel donde se almacenan todos los datos de la aplicación. En el caso de este proyecto se trata de una base de datos, pero podría estar formado por archivos de texto, documentos XML, etc.



5.3 Tecnologías utilizadas

A lo largo de este proyecto se han utilizado diferentes tecnologías, las cuales van a ser explicadas a continuación:

- **C#:** es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (más notablemente de Delphi y Java).

El símbolo # viene de sobreponer "+" sobre "+" y eliminar las separaciones, indicando así su descendencia de C++.

C#, como parte de la plataforma.NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 19 de noviembre de 2007 salió la versión 3.0 de C# destacando entre las mejoras los tipos implícitos, tipos anónimos y el LINQ (Language Integrated Query).

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.

- **SQL SERVER: Microsoft SQL Server** es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son **Oracle, Sybase ASE, PostgreSQL o MySQL**.

Características:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition, que se distribuye en forma *gratuita*.

Es muy común desarrollar completos proyectos complementando *Microsoft SQL Server* y *Microsoft Access* a través de los llamados **ADP** (Access Data Project). De esta forma se completa una potente base de datos (*Microsoft SQL Server*), con un entorno de desarrollo cómodo y de alto rendimiento (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

Para el desarrollo de aplicaciones más complejas (tres o más capas), *Microsoft SQL Server* incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor solo está disponible para Sistemas Operativos Windows.

•**HTML:** siglas de **HyperText Markup Language** (*Lenguaje de Marcas de Hipertexto*), es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar al comportamiento de navegadores Web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

Por convención, los archivos de formato HTML usan la extensión .htm o .html. HTML consiste de varios componentes vitales, incluyendo *elementos* y sus *atributos*, *tipos de datos* y la *declaración del tipo de documento*.

Elementos

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido el documento HTML. Un elemento generalmente tiene una etiqueta de inicio (<nombre-de-elemento>) y una etiqueta de cierre (</nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>). Algunos elementos, tales como
, no tienen contenido ni llevan una etiqueta de cierre.

Atributos

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual "=" y escritos en la etiqueta de comienzo de un elemento, después del nombre de éste. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML (pero no en XHTML). De todas maneras, dejar los valores sin comillas es considerado poco seguro. En contraste con los pares nombre-elemento, hay algunos atributos que afectan al elemento simplemente por su presencia (tal como el atributo *ismap* para el elemento *img*).

ASP.NET: es un framework para aplicaciones Web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios Web dinámicos, aplicaciones Web y servicios Web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Las páginas de ASP.NET, conocidas oficialmente como "*web forms*" (formularios Web), son el principal medio de construcción para el desarrollo de aplicaciones Web. Los formularios Web están contenidos en archivos con una extensión **ASPX**; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML

estático, y también etiquetas definiendo *Controles Web* que se procesan del lado del servidor y *Controles de Usuario* donde los desarrolladores colocan todo el código estático y dinámico requerido por la página Web. Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<% -- código dinámico --%>` que es muy similar a otras tecnologías de desarrollo como PHP (el cual es realmente gratuito), JSP y ASP, pero esta práctica es, generalmente, desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página.

Microsoft recomienda que para realizar programación dinámica se use el modelo **code-behind**, o de respaldo, que coloca el código en un archivo separado o en una etiqueta de script especialmente diseñada. Los nombres de los archivos *code-behind* están basados en el nombre del archivo ASPX tales como *MiPagina.aspx.cs* o *MiPagina.aspx.vb* (esta práctica se realiza automáticamente en Microsoft Visual Studio y otras interfaces de desarrollo). Cuando se usa este estilo de programación, el desarrollador escribe el código correspondiente a diferentes eventos, como la carga de la página, o el clic en un control, en vez de un recorrido lineal a través del documento.

El modelo *code-behind* de ASP.NET marca la separación del ASP clásico y alienta a los desarrolladores a construir aplicaciones con la idea de presentación y contenido separados en mente. En teoría, esto permite a un diseñador Web, por ejemplo, enfocarse en la creación del diseño con menos posibilidades de alterar el código de programación mientras lo hace.

·VISUAL STUDIO: Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones Web, así como servicios Web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas Web y dispositivos móviles.

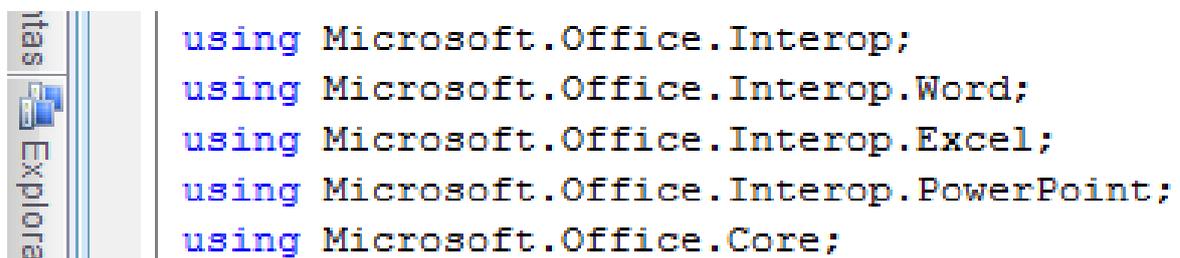
Para el desarrollo de dicho proyecto hemos utilizado dos versiones distintas del Visual Studio, que son el VS2005 y el VS2008.

5.4 Notas adicionales

En este apartado se va a comentar algunos componentes y librerías que han sido utilizadas para la implementación de las aplicaciones que componen este proyecto.

Para llevar a cabo la funcionalidad principal de las aplicaciones de escritorio de este proyecto, ha sido necesario utilizar librerías para el tratamiento de documentos de office y pdf ya que, a petición del cliente, se quería ofrecer la posibilidad de imprimir documentos de office (Word, Excel y Power Point) sin necesidad de abrirlos.

Para tratar con documentos de MS Office y obtener la información necesaria sobre los mismos se han utilizado las siguientes librerías:

The image shows a snippet of C# code in a code editor. On the left, there is a vertical sidebar with a search icon and the text 'Explora'. The code consists of five lines of 'using' statements, each on a new line, all in a blue monospace font. The statements are: 'using Microsoft.Office.Interop;', 'using Microsoft.Office.Interop.Word;', 'using Microsoft.Office.Interop.Excel;', 'using Microsoft.Office.Interop.PowerPoint;', and 'using Microsoft.Office.Core;'.

```
using Microsoft.Office.Interop;
using Microsoft.Office.Interop.Word;
using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.PowerPoint;
using Microsoft.Office.Core;
```

Figura 23: Librerías de office utilizadas

Gracias a estas librerías, además de obtener los metadatos relacionados con los documentos de MS Office, podemos acceder a cierta funcionalidad.

En este caso nos resulta muy útil la posibilidad de guardar cualquier documento de office como pdf, facilitándonos de este modo el recuento de de páginas y mayor flexibilidad para imprimir dichos documentos gracias a las siguientes librerías que trabajan con PDF's:

```
using iTextSharp.text;
using iTextSharp.text.pdf;
```

Figura 24: Librerías utilizadas para la manipulación de PDF's

iText es una librería que permite generar archivos PDF al vuelo. Es una librería ideal para desarrolladores que buscan mejorar sus aplicaciones con la generación dinámica de PDF's y/o su manipulación.

En resumen: las clases iText son muy útiles para generar documentos de solo lectura, independientemente de la plataforma y que contengan texto, tablas, listas o imágenes. También son útiles para manipular documentos PDF existentes.

Por otro lado, para generar PDF's a partir de cualquier tipo de documento que no sea de MS Office, se ha utilizado una impresora virtual a PDF llamada Bullzip PDF Printer que ofrece la posibilidad de automatizar el proceso de conversión y además hacerlo de forma transparente al usuario que ha de imprimir de la manera habitual.

A continuación, se van a exponer algunos ejemplos de utilización de las características mencionadas anteriormente.

•Librerías de MS Office:

```
private string word2pdf(string path, int ini, int fin)
{
    object fileName = path;
    object readOnly = true;
    object isVisible = false;
    object objDNS = Microsoft.Office.Interop.Word.WdSaveOptions.wdDoNotSaveChanges;

    #region Abrir Documento

    Microsoft.Office.Interop.Word.ApplicationClass WordApp = new Microsoft.Office.Interop.Word.ApplicationClass();
    // the way to handle parameters you don't care about in .NET
    object missing = System.Reflection.Missing.Value;
    WordApp.Visible = false;

    // Open the document that was chosen by the dialog
    Microsoft.Office.Interop.Word.Document aDoc = WordApp.Documents.Open(ref fileName,
        ref missing, ref readOnly, ref missing,
        ref missing, ref missing, ref missing,
        ref missing, ref missing, ref missing,
        ref missing, ref isVisible,
        ref missing, ref missing, ref missing, ref missing);

    #endregion

    #region Datos para la conversion|

    string paramExportFilePath = this.generatePath(path + ".pdf");
    WdExportFormat paramExportFormat = WdExportFormat.wdExportFormatPDF;
    bool paramOpenAfterExport = false;
    WdExportOptimizeFor paramExportOptimizeFor =
        WdExportOptimizeFor.wdExportOptimizeForPrint;
    WdExportRange paramExportRange = WdExportRange.wdExportAllDocument;
    int paramStartPage = ini;
    int paramEndPage = fin;
    WdExportItem paramExportItem = WdExportItem.wdExportDocumentContent;
    bool paramIncludeDocProps = true;
    bool paramKeepIRM = true;
    WdExportCreateBookmarks paramCreateBookmarks = WdExportCreateBookmarks.wdExportCreateWordBookmarks;
    bool paramDocStructureTags = true;
    bool paramBitmapMissingFonts = true;
    bool paramUseISO19005_1 = false;
    object paramMissing = Type.Missing;

    #endregion

    #region Conversion a PDF

    aDoc.ExportAsFixedFormat(paramExportFilePath,
        paramExportFormat, paramOpenAfterExport,
        paramExportOptimizeFor, paramExportRange, paramStartPage,
        paramEndPage, paramExportItem, paramIncludeDocProps,
        paramKeepIRM, paramCreateBookmarks, paramDocStructureTags,
        paramBitmapMissingFonts, paramUseISO19005_1,
        ref paramMissing);

    #endregion

    #region Cerrar documento y wordApp

    aDoc.Close(ref paramMissing, ref paramMissing,
        ref paramMissing);
    aDoc = null;

    WordApp.Quit(ref paramMissing, ref paramMissing,
        ref paramMissing);
    WordApp = null;

    GC.Collect();
    GC.WaitForPendingFinalizers();
    GC.Collect();
    GC.WaitForPendingFinalizers();

    #endregion

    return paramExportFilePath;
}
```

Donde:

- Primero se asocia el documento pasando como parámetro la ruta del fichero.
- A continuación se abre el documento incluyendo entre los parámetros el rango de páginas seleccionado y el formato del fichero al que queremos convertir, en este caso PDF.
- Finalmente se cierra el documento y se elimina la referencia para liberar las direcciones de memoria utilizadas. Éste paso es muy importante ya que si no se cierra la aplicación tras la conversión, las aplicaciones de MS Office dejan de funcionar debido a que no permiten múltiples instancias de los mismos.

Para el resto de aplicaciones de la suit MS Office (Excel y Power Point) el código desarrollado para la conversión y obtención de metadatos del archivo referenciado es muy similar al mostrado anteriormente con ligeras variaciones como se muestra a continuación:

Para Excel:

```
private string excel2pdf(string path, int ini, int fin)
{
    object fileName = path;
    object readOnly = true;
    object isVisible = false;
    object objDNS = Microsoft.Office.Interop.Excel.XlSaveAction.xlDoNotSaveChanges;

    #region Abrir el documento

    Microsoft.Office.Interop.Excel.ApplicationClass ExcelApp = new Microsoft.Office.Interop.Excel.ApplicationClass();
    object missing = System.Reflection.Missing.Value;
    ExcelApp.Visible = false;
    Microsoft.Office.Interop.Excel.Workbook aXLS = ExcelApp.Workbooks.Open(path,
        missing, readOnly, missing,
        missing, missing, missing,
        missing, missing, missing,
        missing, isVisible,
        missing, missing, missing);

    #endregion

    Datos para la conversion
}
```

Y para Power Point:

```
private string powerpoint2pdf(string path, int ini, int fin, int diapHoja)
{
    object fileName = path;
    MsoTriState readOnly = MsoTriState.msoTrue;
    object isVisible = false;
    object missing = System.Reflection.Missing.Value;

    #region Abrir el documento

    Microsoft.Office.Interop.PowerPoint.ApplicationClass PowerApp = PowerApp = new Microsoft.Office.Interop.PowerPoint.
    PowerApp.Visible = MsoTriState.msoTrue;
    Microsoft.Office.Interop.PowerPoint.Presentation aPPT = aPPT = PowerApp.Presentations.Open(path,
        readOnly, Microsoft.Office.Core.MsoTriState.msoFalse,
        Microsoft.Office.Core.MsoTriState.msoTrue);

    #endregion

    #region Datos para la conversion

    string paramExportFilePath = this.generatePath(path + ".pdf");
    object paramMissing = Type.Missing;
    PpFixedFormatType formatType = PpFixedFormatType.ppFixedFormatTypePDF;
    PpFixedFormatIntent formatIntent = PpFixedFormatIntent.ppFixedFormatIntentPrint;
}
```

Por otro lado, en la aplicación que se instalará en el servidor de impresión, ha sido necesario importar la librería “winspool.drv” para imprimir documentos tal y como se muestra en el siguiente fragmento de código:

```
//Datos en string para ser impreso
static string prnData = string.Empty;

//Añadir connexion para una impresora especifica
[DllImport("winspool.drv")]
public static extern bool AddPrinterConnection(string pName);

//Seleccionar la impresora dada como predeterminada
[DllImport("winspool.drv", CharSet = CharSet.Auto, SetLastError = true)]
public static extern bool SetDefaultPrinter(string name);
#endregion
```

A continuación se muestra el fragmento de código que hace posible la impresión de documentos pasando como parámetros únicamente el nombre de la impresora y el documento a imprimir:

```
//Instanciar el objeto ProcessStartInfo
Process objProcess = new Process();
try
{
    //Configurar la impresora
    AddPrinterConnection(printer);
    SetDefaultPrinter(printer);

    //Imprimir el archivo

    objProcess.StartInfo.FileName = this.BDGetPathBolsa()+ id+".pdf";
    //objProcess.StartInfo.FileName = @"C:\Users\Marcos\Desktop\Docs Prueba\Manual de ECO.pdf";
    objProcess.StartInfo.CreateNoWindow = true;
    objProcess.StartInfo.Verb = "Print";

    objProcess.StartInfo.WindowStyle = ProcessWindowStyle.Hidden;
    objProcess.StartInfo.UseShellExecute = true;
    objProcess.Start();

    //Actualizar en la BD el campo userImpr
    this.BDUpdateUserImpr(id, this.getUserName());
    //Actualizar en la BD el precio final del documento
    this.BDUpdatePrecioEnvio(precio, id);
    //Actualizar en la BD la impresora utilizada
    this.BDUpdatePrinterEnvio(printer, id);

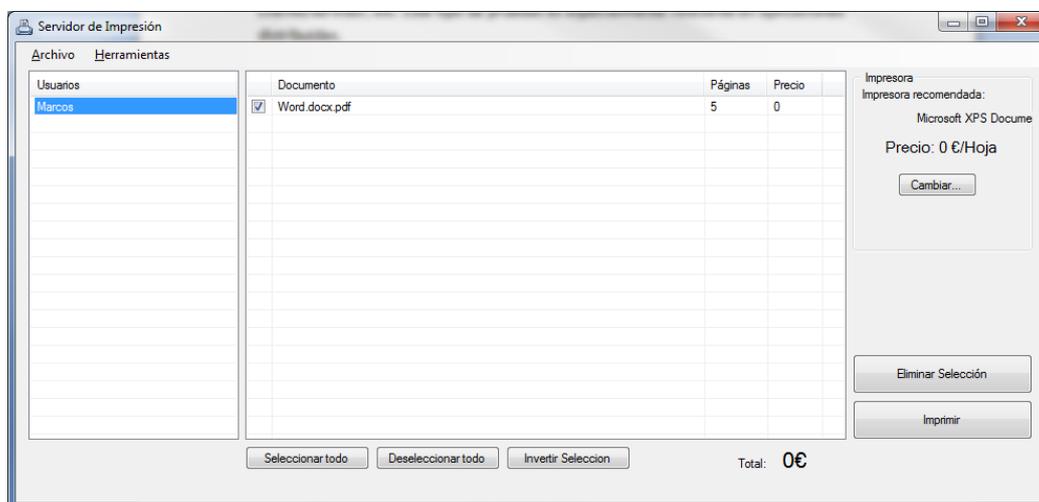
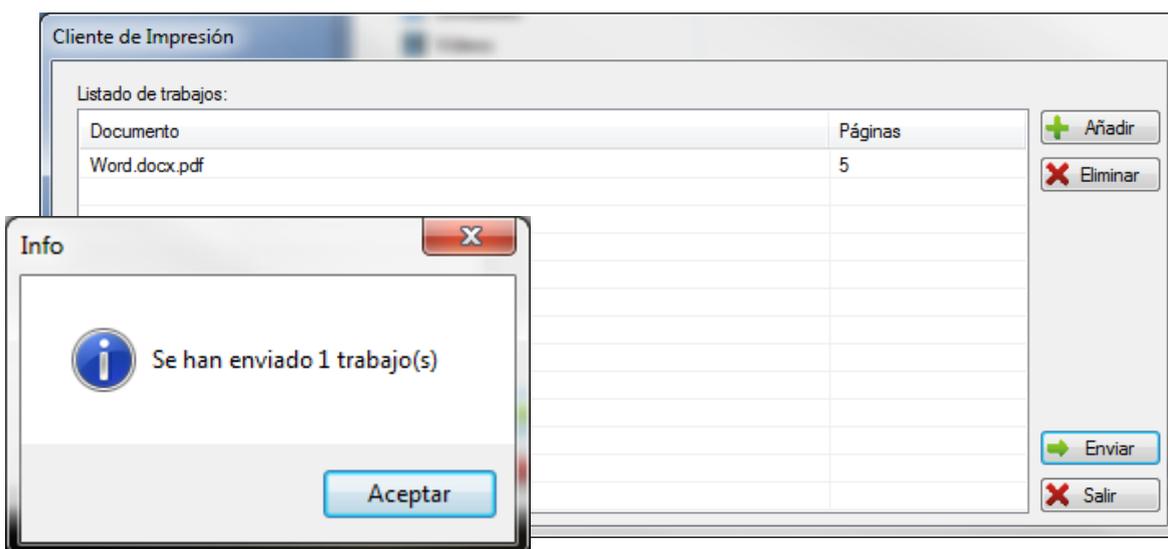
    //devuelve true si exito
    return true;
}
```

6. EVALUACIÓN

Para asegurar el correcto funcionamiento de las aplicaciones que componen este proyecto se han realizado diferentes pruebas, las cuales se van a describir a continuación:

• **Pruebas de Integración:** En este tipo de pruebas se comprueba la compatibilidad y funcionalidad de los interfaces entre las distintas 'partes' que componen un sistema, estas 'partes' pueden ser módulos, aplicaciones individuales, aplicaciones cliente/servidor, etc. Este tipo de pruebas es especialmente relevante en aplicaciones distribuidas.

En el caso concreto del proyecto a tratar, se ha comprobado la perfecta comunicación y sincronía entre la aplicación instalada en el ordenador cliente y la aplicación instalada en el servidor de impresión mediante pruebas de envío, eliminación y modificación de trabajos de impresión.



Además, se ha probado el sistema en diferentes ordenadores a diferentes resoluciones aunque se encuentra optimizado para 1280x1024.

Pruebas de sistema: el software ya validado se integra con el resto del sistema donde algunos tipos de pruebas a considerar son:

·**Rendimiento:** determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.

Hay que tener en cuenta que en épocas de gran actividad como el comienzo del curso o las épocas de exámenes pueden generar cantidad de información que deberá ser almacenada en la Base de Datos. Esto puede provocar lentitud y un tamaño importante de la misma por lo que quizás se debería contemplar algún sistema de copias de seguridad periódico.

Por otro lado, el tiempo de respuesta de la aplicación instalada en el servidor, viene determinada por dos factores:

·La velocidad de transmisión de la red en la que se emplea el sistema, según la cual los documentos a imprimir se transmitirán con mayor o menor brevedad

·El tiempo de refresco de la lista de trabajos de la aplicación servidor que se ha fijado en 5 segundos por mantener un compromiso entre la carga del sistema y el tiempo de respuesta.

·**Robustez:** Determinan la capacidad del programa para soportar entradas incorrectas.

Con el amplio abanico de usuarios que han probado el sistema, se ha comprobado la capacidad del mismo ante entradas incorrectas comprobando, por ejemplo, que el rango de páginas a imprimir es correcto.

·**Seguridad:** se determinan los niveles de permiso de usuarios, las operaciones de acceso al sistema y acceso a datos.

A decisión del cliente se deja la posibilidad de que un usuario pueda imprimir trabajos de otros usuarios o no. En caso de no permitirlo se implementa un simple login que filtre los trabajos en función del usuario que los ha enviado.

El acceso al apartado web y a la configuración de precios e impresoras está reservado al personal del departamento de informática de la universidad, por lo que se requiere el conocimiento de un password o en el caso de implantarlo en la universidad, usuario y contraseña dentro del dominio de personal.

•**Pruebas de Aceptación:** Son las que hará el cliente, se determina que el sistema cumple con lo deseado y se obtiene la conformidad del cliente.

Tras verificar todas las pruebas técnicas que aseguran el correcto funcionamiento de las aplicaciones que conforman este proyecto se han llevado a cabo una serie de pruebas de validación que demuestren que las aplicaciones sean sencillas de usar y funcionen como se espera.

Para estas pruebas se ha contado con la colaboración de varios usuarios con diferente familiarización con la informática que se han prestado a utilizar las aplicaciones y evaluarlo.

Las personas que han colaborado con las pruebas han sido 3 oficinistas, 3 componentes del departamento de informática de la universidad, dos personas mayores de 50 años y varios estudiantes. Para todos ellos el sistema ha sido sencillo de utilizar y se han obtenido los resultados esperados.

7. CONCLUSIONES

Ha resultado una experiencia muy interesante haber llevado a cabo este proyecto ya que para terminarlo ha sido necesario aplicar gran parte de los conocimientos adquiridos durante los estudios de informática y es muy gratificante ver la utilidad de los mismos a la hora de desarrollar un proyecto prácticamente real y completo.

Por otro lado, es excitante ir llevando a cabo todas las fases que posee un proceso software y que hemos visto en asignaturas como ISG. Comenzando por la captura de requisitos, y planificando reuniones bastante a menudo para que el “cliente” pudiera seguir el desarrollo del proceso, participe del mismo y vaya aportando sugerencias.

El desarrollo de este proyecto también ha supuesto todo un reto ya que gran parte ha implicado la investigación de nuevas tecnologías que no se habían tratado en profundidad a lo largo de la carrera tales como el desarrollo Web o el trabajo con librerías externas.

A pesar de haber realizado este proyecto sin compañeros, en la empresa dónde lo he realizado he podido obtener ayuda cuando por mi mismo no he hallado solución a ciertos problemas. Aún así, se han echado de menos ciertos aspectos típicos de los procesos software como el trabajo en equipo. Por otro lado, otros aspectos como la comunicación con el cliente, el cumplimiento de plazos de entrega y la realización de pruebas han estado fuertemente presentes.

Para finalizar, solo cabe mencionar que la realización de éste proyecto de final de carrera ha sido de un inestimable valor. Permite llevar a cabo un proyecto casi real, muy similar a uno en el mundo laboral y además le termina de dar sentido a los conocimientos aprendidos durante la carrera, que si bien en un principio pueden parecer un poco dispersos, tras la realización del proyecto todo cobra un mayor sentido.

8. Bibliografía

•Programación en Visual C# .NET de Francisco Charde Ojeda.

Anaya Multimedia

•Programación en Visual C#

Documentación de las asignaturas ISG, DSW y APW.

•Modelo OOWS

http://66.102.9.104/search?q=cache:kMgUA_PtDlwJ:www.ing.unlpam.edu.ar/icwe2002/tutoriales/opastor.pdf+oows&hl=es&ct=clnk&cd=2&gl=es

•Desarrollo de aplicaciones y Office

[http://msdn.microsoft.com/es-es/library/e7ezs44a\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/e7ezs44a(VS.80).aspx)

•iText Sharp

<http://www.codeproject.com/KB/graphics/iTextSharpTutorial.aspx>

<http://kuujinbo.info/cs/itext.aspx>

•Requisitos para servidor Web aspx

[http://msdn.microsoft.com/es-es/library/ms228041\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/ms228041(VS.80).aspx)

•Requisitos hardware para Windows

<http://www.trucoswindows.com/requisitos/requisitos.php?limite=8>

•Instalación y configuración de IIS

http://www.netveloper.com/contenido2.aspx?IDC=32_0&IDP=1&P=0

•Requisitos SQLServer

<http://www.microsoft.com/spain/sql/2000/productinfo/sysreq.aspx>

·**Descarga del framework 4.0**

<http://www.microsoft.com/downloads/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en>

·**IEEE STD 830- IEEE Guide to Software Requirements Specifications.**

·*msdn.microsoft.com/es-es/default.aspx*

·<http://www.codeproject.com/>

9. Anexos

9.1 Manuales de instalación

9.1.1 Prerrequisitos

Para dar soporte al portal Web, el sistema ha de cumplir una serie de requisitos en la máquina que alojará el portal y hará de servidor.

A continuación se procederá a comentar en detalle los requisitos mencionados anteriormente y las configuraciones necesarias tanto para dar soporte al apartado Web como a las aplicaciones de escritorio que componen este proyecto.

Apartado Web

Se recomienda una versión de Windows de tipo Server ya que dichas versiones están orientadas hacia el trabajo en red y ofrecen un mayor nivel de seguridad.

·Requisitos Hardware para sistemas Windows:

Windows server 2003 standard edition

- procesador a 133 Mhz, recomendado 550 Mhz o superior, soporta hasta 4 procesadores en un Server
- 128 megabytes (MB) de memoria ram, recomendado 256 MB, máximo 4 GB
- entre 1.25 y 2 GB de espacio en el disco duro
- unidad de CD-ROM o DVD-ROM
- VGA o hardware que soporte consola de redirección, soporta súper VGA a 800x600, recomendado una resolución mayor

Windows NT 4.0

- procesador de 32 bits basado en Intel x86: 80486 a 33 Mhz o superior
- procesador basado en Intel Pentium o en Pentium pro
- procesador basado en MIPS4 R4000
- procesador basado en digital alpha AXP
- procesador basado en powerpc compatible prep
- 16 megabytes (MB) de memoria ram
- equipos basados en Intel x86: 110 MB para Windows NT Workstation y 125 MB para Windows NT Server
- equipos basados en RISC: 110 MB para Windows NT Workstation y 160 MB para Windows NT Server
- adaptador de pantalla de video con resolución VGA o superior
- los equipos basados en Intel x86 requieren una unidad de disquetes de alta densidad de 3,5 pulgadas y una unidad de CD-ROM, en el caso de los equipos sin unidad de CD-ROM, instalaremos Windows NT a través de red, los equipos basados en RISC requieren una unidad de CD-ROM

Componentes opcionales:

- tarjeta de red
- ratón o cualquier otro dispositivo señalador

Windows XP professional

- procesador Pentium a 233 megahercios o superior, se recomienda 300 Mhz
- 64 megabytes (MB) de memoria ram, se recomienda 128 MB
- 1,5 GB de espacio en el disco duro
- unidad de CD-ROM o DVD-ROM
- teclado y Microsoft Mouse o dispositivo señalador compatible
- adaptador de vídeo y monitor con resolución Super VGA (800 x 600) o superior
- tarjeta de sonido
- altavoces o auriculares

Windows Server 2003 Standard edition

- procesador a 133 Mhz, recomendado 550 Mhz o superior, soporta hasta 4 procesadores en un Server
- 128 megabytes (MB) de memoria ram, recomendado 256 MB, máximo 4 GB
- entre 1.25 y 2 GB de espacio en el disco duro
- unidad de CD-ROM o DVD-ROM
- VGA o hardware que soporte consola de redirección, soporta súper VGA a 800x600, recomendado una resolución mayor

Windows Vista Ultimate

- Procesador de 32 bits (x86) o de 64 bits (x64) a 1 GHz
- 1 GB de memoria del sistema
- 40 GB de disco duro con al menos 15 GB de espacio disponible
- Compatibilidad con gráficos DirectX 9 con lo siguiente:
- Controlador WDDM
- 128 MB de memoria de gráficos (mínimo)
- Unidad de DVD-ROM
- Salida de audio

Los requisitos y la funcionalidad del producto pueden variar según la configuración del sistema.

Requisitos mínimos admitidos

Algunas características del producto no se encuentran disponibles con los requisitos mínimos admitidos.

- Procesador a 800 MHz y 512 MB de memoria del sistema
- 20 GB de disco duro con al menos 15 GB de espacio disponible
- Compatibilidad con gráficos Súper VGA
- Unidad de CD-ROM

Windows 7 Ultimate

- Procesador de 32 bits (x86) o de 64 bits (x64) a 1 GHz
- 1 GB de memoria del sistema
- 40 GB de disco duro con al menos 15 GB de espacio disponible
- Compatibilidad con gráficos DirectX 9 con lo siguiente:
 - Controlador WDDM
 - 128 MB de memoria de gráficos (mínimo)
 - Pixel Shader 2.0 en hardware
 - 32 bits por píxel
- Unidad de DVD-ROM
- Salida de audio
- Acceso a Internet (si procede, con tarifa aplicable)

.NET Framework 4 o superior

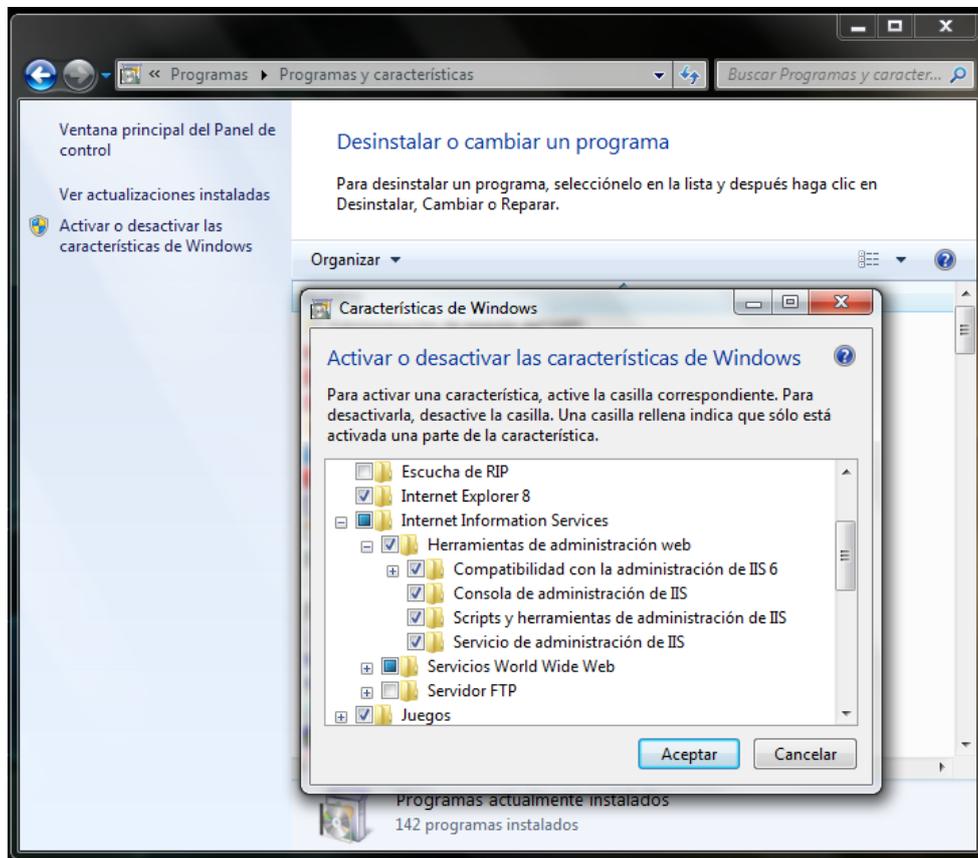
- **Sistemas operativos compatibles:** Windows 7; Windows Server 2003 Service Pack 2; Windows Server 2008; Windows Server 2008 R2; Windows Vista Service Pack 1; Windows XP Service Pack 3
 - Windows XP SP3
 - Windows Server 2003 SP2
 - Windows Vista SP1 o posterior
 - Windows Server 2008 (no admitido en el rol Server Core)
 - Windows 7
 - Windows Server 2008 R2 (no admitido en el rol Server Core)
- **Arquitecturas compatibles:**
 - x86
 - x64
 - ia64 (algunas características no se admiten en ia64; por ejemplo, WPF)
- **Requisitos de hardware:**
 - Mínimo recomendado: Pentium 1 GHz o superior con 512 MB de RAM o más
 - Espacio en disco mínimo:
 - x86: 850 MB
 - x64: 2 GB
- **Requisitos previos:**
 - [Windows Installer 3.1](#) o posterior
 - [Internet Explorer 5.01](#) o posterior

9.1.2 Instalación del Portal Web

A continuación se va a explicar todo el proceso mediante el cual se conseguirá que se pueda acceder al portal Web a través de internet.

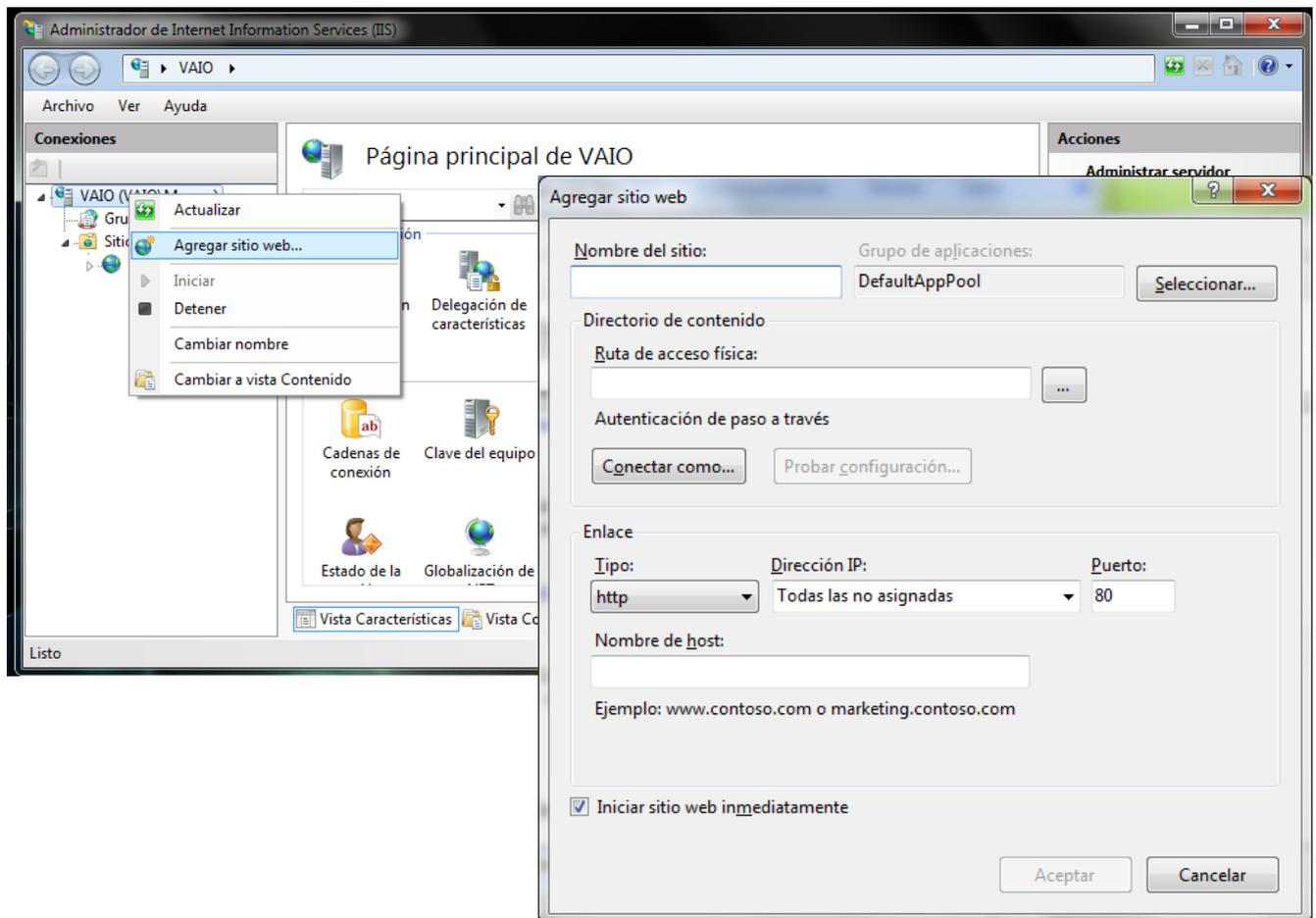
1º- Hay que comprobar que tengamos disponible el IIS (Internet Information Service) ya que, a pesar de tener instalado un sistema operativo compatible, es posible que no se encuentre instalado. En este caso, el modo de proceder para instalarlo será el siguiente:

En "Panel de control" -> "Desinstalar o cambiar un programa" -> "Activar o desactivar características de Windows", y elegiremos instalar IIS.



A continuación, hay que crear la aplicación en nuestra máquina para que esté disponible. Para ello se debe copiar los archivos de la aplicación en la carpeta que por defecto utiliza el IIS, es decir, **c:\inetpub\wwwroot**

A continuación, hay que crear un directorio virtual en IIS que haga referencia al portal. Para ello, se puede hacer click derecho sobre el sitio Web en el que queremos definirlo y seleccionar la opción “Agregar sitio Web” apareciendo a continuación un asistente que nos guiará a lo largo del proceso.



Donde hay que introducir:

- Nombre del sitio: Que es el nombre lógico que queremos darle al directorio.
- Ruta de acceso física: Que es la ruta del directorio en el disco duro o en la red local.

Una vez finalizado el asistente queda creado el directorio virtual al que se podrá acceder a través del nombre del sitio. Por ejemplo:

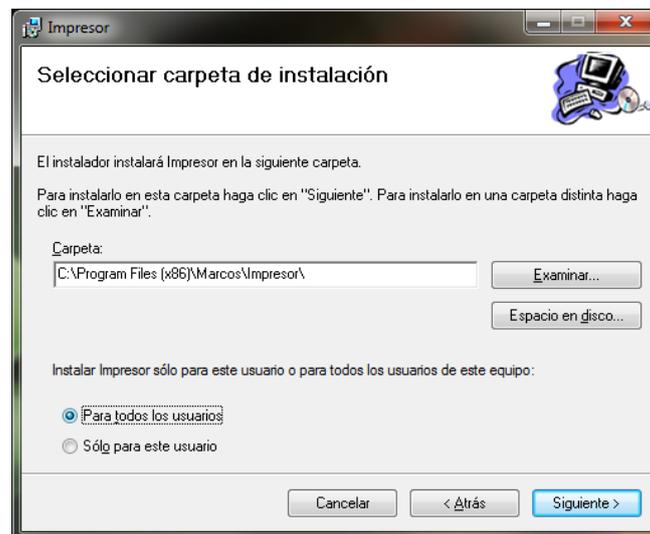
http://localhost/webImpr/pagina_de_inicio.aspx

9.1.3 Instalación de las aplicaciones de escritorio

Para instalar las dos aplicaciones de escritorio desarrolladas en este proyecto, el procedimiento es el habitual para instalar cualquier aplicación simple.

Cada una tiene su propio instalador, el cual hay que ejecutar para lanzar el asistente de instalación.

Éste comprobará si se cumplen los requisitos mínimos para el funcionamiento de las aplicaciones y pedirá una ruta donde instalarlas.



MUY IMPORTANTE: Este sistema de impresión, utiliza una impresora a PDF automatizada que ha de estar instalada en el sistema.

La impresora utilizada y que se adjunta con la documentación es la impresora virtual BullzipPDF Printer y se puede descargar gratuitamente de:

<http://www.bullzip.com>

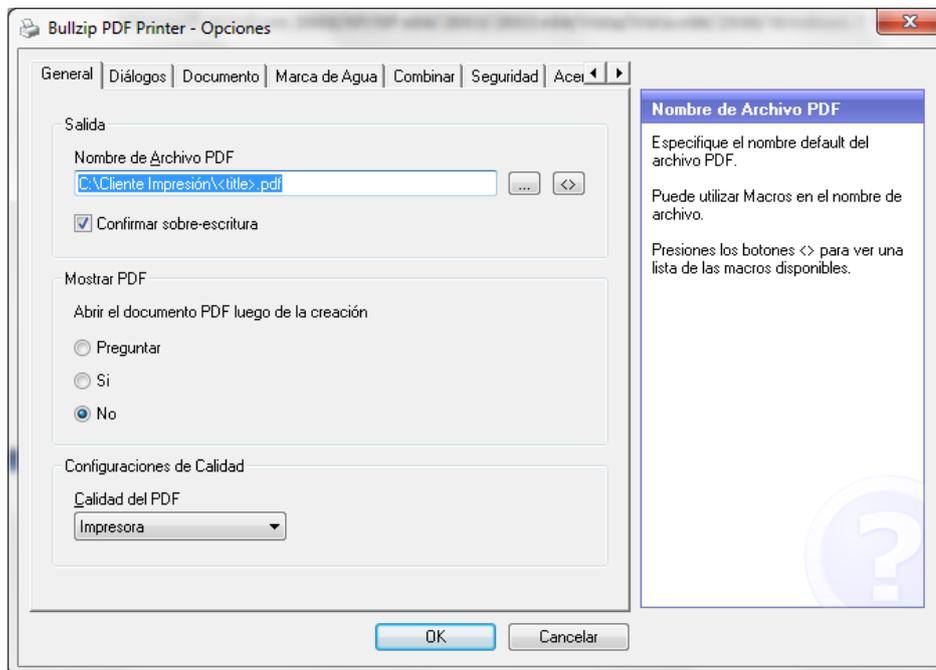
Requisitos:

- Microsoft Windows 2000/XP/XP x64/2003/2003 x64/Vista/Vista x64/2008/Windows 7 
- GPL Ghostscript 8.64 or later (<http://www.bullzip.com/download/gsl/gslite.exe>)

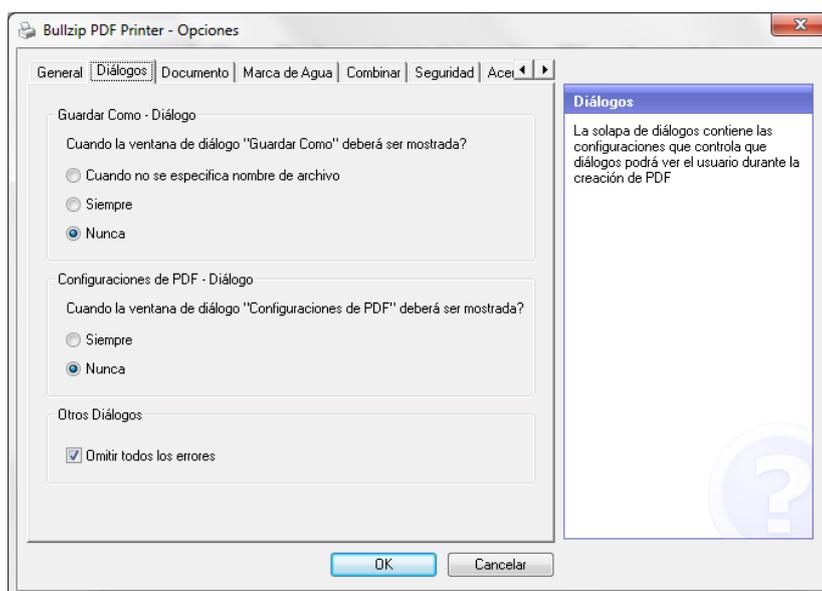
Una vez instalada la impresora, solo falta configurarla para que haga las conversiones de forma automática.

Para ello abrimos el gestor de la impresora y configuramos lo siguiente:

·En la pestaña “General” indicamos la ruta de los archivos PDF resultantes de la conversión y marcamos que no abra el documento tras la creación.



·En la pestaña diálogos, marcamos que no muestre el diálogo “Guardar como...” ni el de “Configuraciones de PDF...” para conseguir que todo el proceso se haga de manera transparente al usuario.



Así termina la configuración de la impresora virtual.

9.2 Manual de usuario

9.2.1 Aplicación de escritorio Cliente

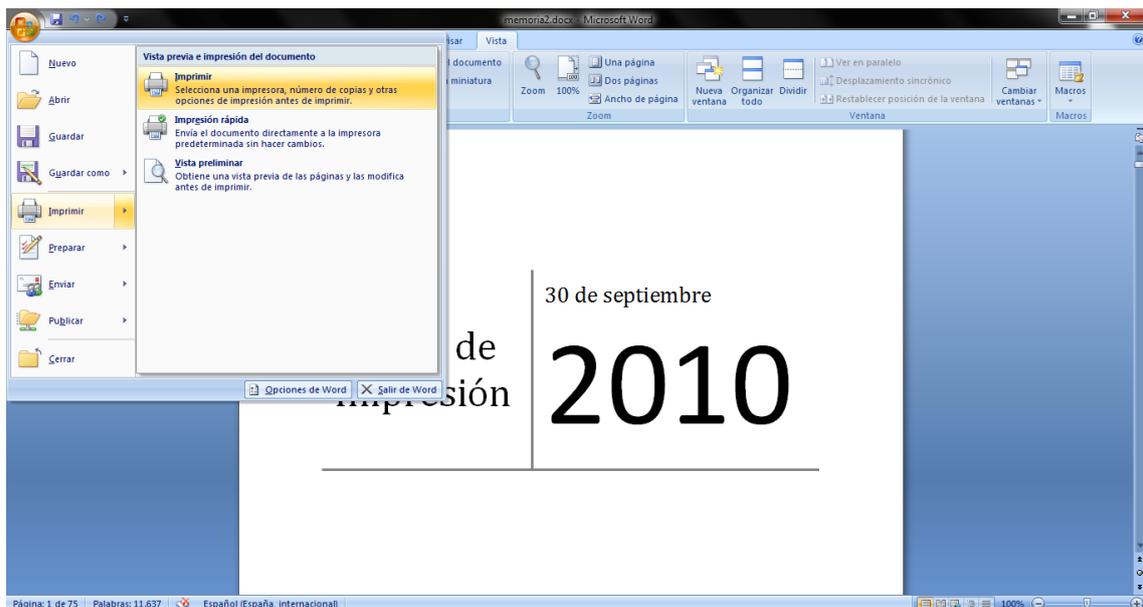
La aplicación cliente de este proyecto es muy fácil de usar ya que se encarga de una única tarea que se puede llevar a cabo de diferentes maneras.

El objetivo de esta aplicación es enviar los documentos que aparezcan en la lista a un repositorio existente en el servidor mediante unos sencillos pasos:

1º- Seleccionar los archivos a enviar

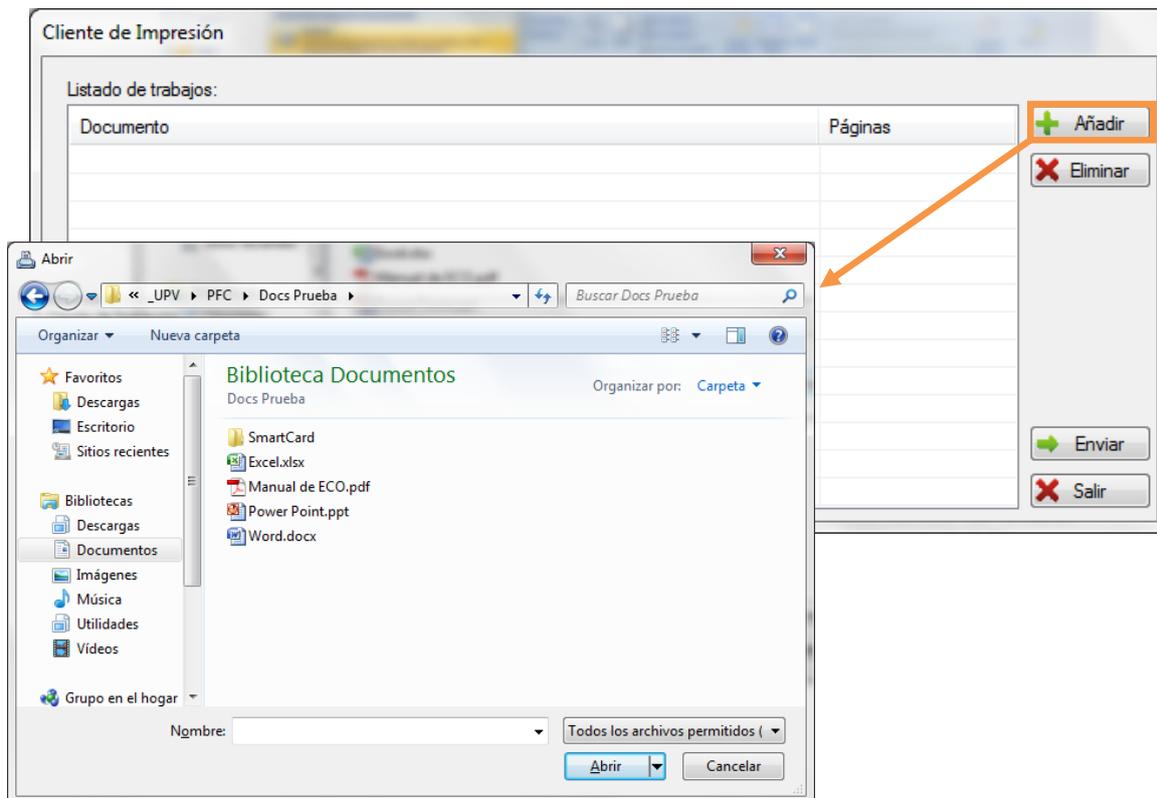
·El método más natural es enviar los documentos desde la misma aplicación con la que se trabaja con ellos, es decir, tal y como se enviaría un trabajo a imprimir de la forma habitual.

Lo que hay que tener en cuenta es que la impresora seleccionada deberá ser la impresora virtual a PDF y en vez de imprimirse el documento, se añadiría a la lista de documentos pendientes.



·Existen otros dos métodos de impresión exclusivamente para documentos de la suite **Microsoft Office** que permiten la impresión de documentos sin la necesidad de abrirlos.

Uno de los métodos es haciendo click sobre el botón “Añadir”, apareciendo un “FileBrowseDialog” en el que deberemos indicar el documento a añadir:

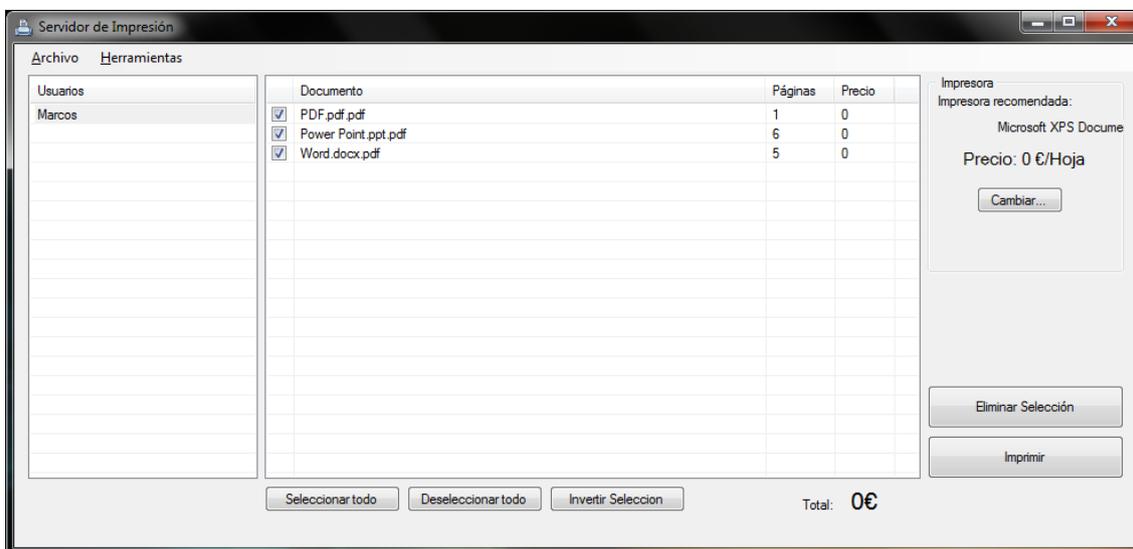


El otro método es mediante “Drag & Drop”. Los documentos de office pueden ser arrastrados a la lista de la aplicación y serán analizados y añadidos automáticamente.

Finalmente, cuando en la lista se encuentren los documentos que deseemos imprimir, solo falta hacer click en “Enviar” para enviarlos al repositorio ubicado en el servidor de impresión.

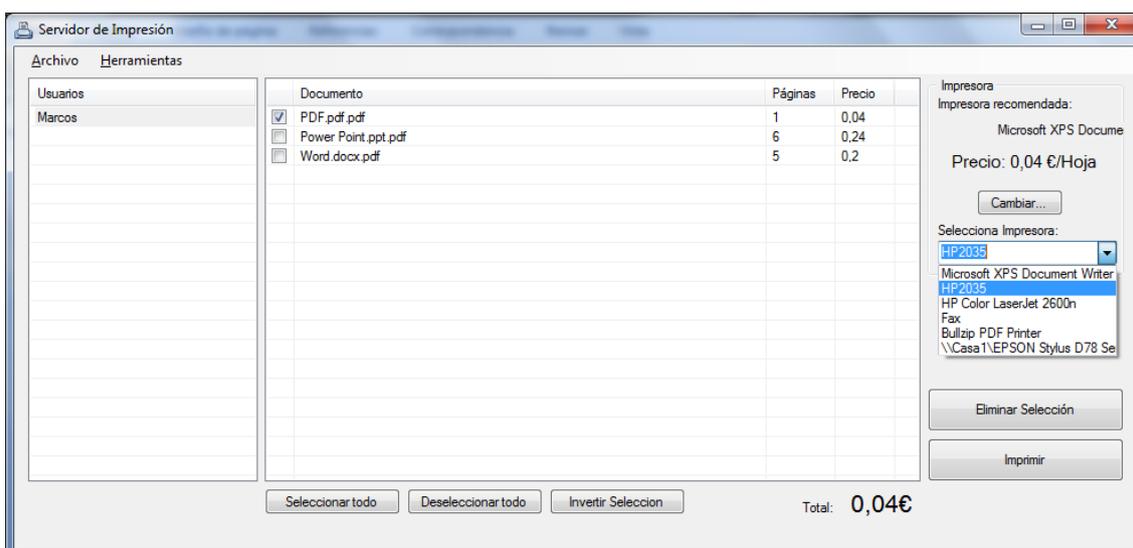
9.2.2 Aplicación de escritorio Servidor

La aplicación de escritorio instalada en el servidor permite visualizar los trabajos pendientes de impresión además de seleccionarla impresora y conocer el coste que tendría dicha impresión.



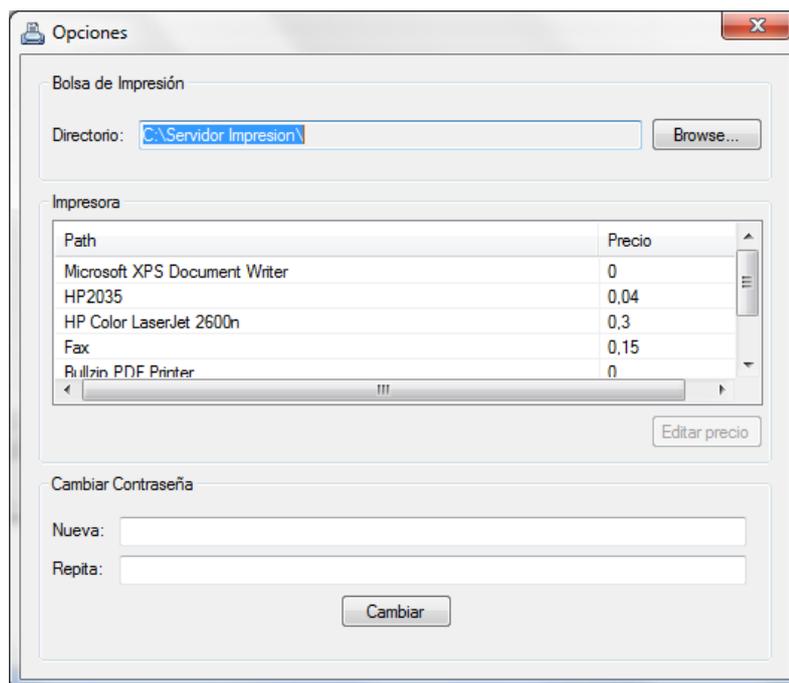
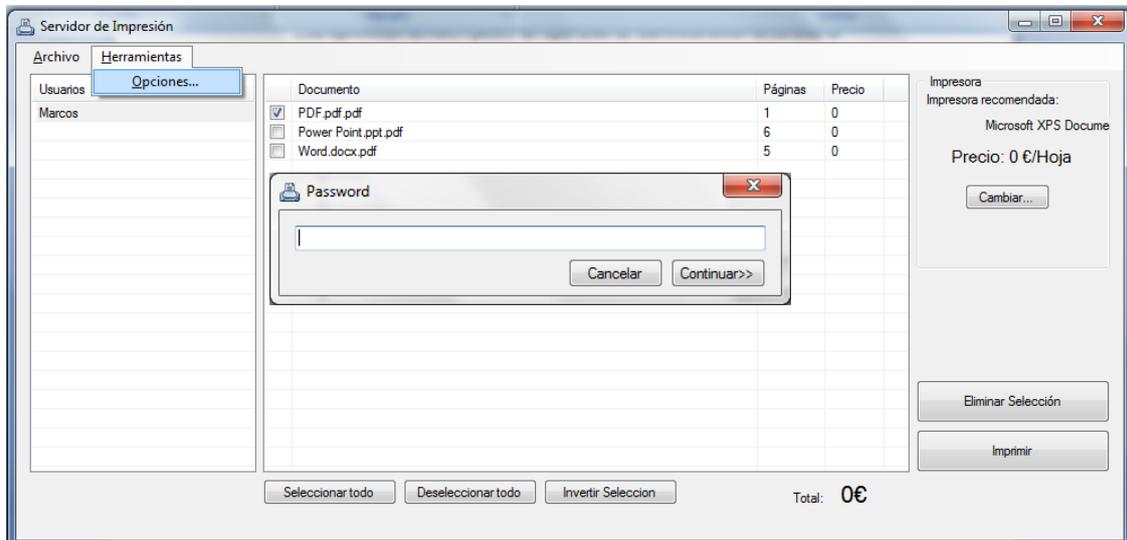
Como se puede apreciar en la captura anterior, los trabajos están organizados por usuario y muestran su precio en función de la impresora seleccionada.

Haciendo click sobre el botón "Cambiar..." situado en la zona de impresoras, se puede elegir una impresora diferente ya sea por la ubicación o por tener un precio más conveniente.



Mediante los botones inferiores se puede administrar la selección, y una vez seleccionados los trabajos deseados, estos se pueden imprimir haciendo click en el botón "Imprimir". Aparecerá un cuadro de confirmación en el que el usuario decidirá finalmente si desea imprimir los trabajos seleccionados o no.

Esta aplicación también posee un apartado de administración destinada al personal de la universidad, mediante la cual, se podrá actualizar ciertos parámetros como principalmente el precio de las impresoras instaladas en el sistema.



9.2.3 Aplicación Web

Debido a que el apartado Web de este proyecto se centra principalmente en la visualización de información, el manual no es muy extenso.

Todas las páginas contienen en la parte de navegación situada a la izquierda de las mismas, un link a las diferentes páginas que componen la aplicación. Cada una de ellas mostrará información relevante al cliente.

