



Universidad de Valladolid
Escuela de Ingeniería Informática

TRABAJO DE FIN DE GRADO

**DISEÑO Y ELABORACIÓN
DEL TABLERO DE DATOS
DEL DEPARTAMENTO DE
PINTURA DE
RENAULT ESPAÑA**

Grado en Ingeniería Informática

Autor
Santos Ángel Pardo Ramos

Tutor
Jesús María Vegas Hernández

1. Introducción	11
1.1. Visión General	11
1.2. Motivación	12
1.3. Problema Existente	12
1.4. Objetivos	13
1.5. Metodología	14
1.6. Plan de Fases	14
1.7. Estructura del Documento	15
2. Marco Conceptual	17
2.1. ¿Qué Es Renault?	17
2.2. Historia de Renault	17
2.3. Estructura de la Fábrica	18
2.4. Estructura del Departamento	18
2.5. Entorno Informático de Renault	21
2.5.1. Descripción	21
2.5.2. Seguridad	22
3. Planificación	25
3.1. Introducción	25
3.2. Descripción del Sistema a Desarrollar	25
3.3. Recursos	26
3.3.1. Recursos Humanos	26
3.3.2. Recursos Hardware y Software	27
3.4. Plan de Despliegue	28
3.5. Asignación de Recursos y Calendario	28
3.6. Plan de Proyecto	29
3.7. Costes	30
3.8. Análisis de Riesgos	31
3.8.1. Riesgo de Proyecto	31
3.8.2. Riesgo Técnicos	34
3.8.3. Riesgo de Negocio	34
4. Análisis	35
4.1. Método de Recogidas de Datos	35
4.2. Participantes en el Proyecto	35
4.3. Objetivos del Sistema	36
4.4. Requisitos	37

4.4.1.	Requisitos Funcionales	38
4.4.2.	Requisitos No Funcionales	41
4.5.	Actores	42
4.6.	Casos de Uso	43
4.7.	Diagramas de Actividad	52
4.7.1.	Iniciar Sesión	52
4.7.2.	Configurar	53
4.7.3.	Hacer Pedido Manual	54
4.7.4.	Hacer Pedido	55
4.7.5.	Enviar Pedido	56
4.7.6.	Leer Fichero	57
4.7.7.	Enviar Artículos	58
4.8.	Diagrama de Dominio	58
5.	Diseño	61
5.1.	Diseño de Persistencia	61
5.1.1.	Decisiones Sobre el Diseño de Persistencia	61
5.1.1.1.	Persistencia en la Aplicación Formulario	61
5.1.1.1.1.	Descripción	62
5.1.1.2.	Persistencia en la Aplicación Envío	63
5.1.1.2.1.	Descripción	63
5.2.	Diseño Arquitectónico	63
5.2.1.	Introducción	63
5.2.2.	Condiciones	63
5.2.3.	Organización del Sistema	64
5.2.4.	Flujo de Control	64
5.3.	Modelos del Sistema	65
5.3.1.	Patrones Utilizados	65
5.3.2.	Modelos Estáticos	67
5.3.2.1.	Aplicación Formulario	67
5.3.2.1.1.	Diagrama de clases global	68
5.3.2.1.2.	Diagrama de clases del modelo	70
5.3.2.1.3.	Diagrama de clases de las vistas	71
5.3.2.1.4.	Diagrama de clases de los controladores	72
5.3.2.1.5.	Controladores	73
5.3.2.1.5.1.	Clase Controlador	73
5.3.2.1.5.2.	Clase ControladorAdministracionSesion	73
5.3.2.1.5.3.	Clase ControladorConfiguracion	73
5.3.2.1.5.4.	Clase ControladorConfirmacion	74
5.3.2.1.5.5.	Clase ControladorControladorHacerPedido	75
5.3.2.1.5.6.	Clase ControladorInicioSesion	76
5.3.2.1.5.7.	Clase ControladorPantallaMenu	77
5.3.2.1.5.8.	Clase ControladorVerCatalogo	77
5.3.2.1.6.	DataMapper	78
5.3.2.1.6.1.	Clase DataMapperModelo	78
5.3.2.2.	Aplicación Envío	79
5.3.2.2.1.	Diagrama de clases global	79
5.3.2.2.2.	Diagrama de clases del modelo	81
5.3.2.2.3.	Diagrama de clases de las vistas	82
5.3.2.2.4.	Diagrama de clases de los controladores (Hilos)	83
5.3.2.2.5.	Diagrama de clases de los controladores	84
5.3.2.2.6.	Controladores	85
5.3.2.2.6.1.	Clase Controlador	85
5.3.2.2.6.2.	Clase ControladorLoading	85

5.3.2.2.6.3.	Clase ControladorIniciarSesion	86
5.3.2.2.6.4.	Clase ControladorDatosPedido	86
5.3.2.2.7.	Hilo	88
5.3.2.2.7.1.	Clase Hilo	88
5.3.2.2.7.2.	Clase HiloPowerShell	88
5.3.2.2.7.3.	Clase HiloInicioSesión	89
5.3.2.2.7.4.	Clase HiloValidarCantidadPedida	90
5.3.2.2.7.5.	Clase HiloFinDePedido	91
5.3.3.	Modelos Dinámicos	91
5.3.3.1.	Aplicación Formulario	92
5.3.3.1.1.	Diagramas de secuencia	92
5.3.3.1.1.1.	Iniciar sesión	93
5.3.3.1.1.2.	Acceder a los ajustes del sistema	94
5.3.3.1.1.3.	Ajustar el sistema	95
5.3.3.1.1.4.	Hacer el pedido respecto a otro anterior	96
5.3.3.1.1.5.	Hacer pedido	97
5.3.3.1.1.6.	Modificar pedido	98
5.3.3.1.1.7.	Ver referencias disponibles	99
5.3.3.2.	Aplicación Envío	100
5.3.3.2.1.	Diagramas de secuencia	100
5.3.3.2.1.1.	Iniciar sesión	101
5.3.3.2.1.2.	Leer fichero Excel	102
5.3.3.2.1.3.	Enviar pedido	103
5.4.	Diseño de Interfaz de Usuario	104
6.	Tecnología Empleada	107
6.1.	Java	107
6.2.	Microsoft Office Excel	107
6.3.	Apache POI	108
6.4.	Windows PowerShell	108
6.4.1.	Invoke-WebRequest	108
7.	Implementación	113
7.1.	Introducción	113
7.2.	Estructura de Directorios	113
7.3.	Seguridad en las Contraseñas	115
7.4.	Estructura del Fichero Excel	116
7.4.1.	Usuarios	116
7.4.2.	Fechas	117
7.4.3.	Presupuesto	117
7.4.4.	Referencias	118
7.4.5.	Detalles	118
7.4.6.	Ubicación	119
7.4.7.	Pedidos	119
8.	Instalación y Pruebas	121
8.1.	Sistema Operativo	121
8.2.	Pruebas de Sistema	121
8.3.	Pruebas de Caja Negra	121
9.	Conclusiones	125
9.1.	Objetivos Alcanzados	125
9.2.	Dificultades Encontradas	125
9.3.	Conocimiento Adquirido	126
9.4.	Futuros Objetivos y Ampliaciones	126

A. Aplicación Formulario: Manual de Instalación y Usuario	129
A.1. Manual de Instalación	129
A.1.1. Introducción	129
A.1.2. Requisitos de Instalación	129
A.1.3. Instalación	130
A.2. Manual de Usuario	131
A.2.1. Pantalla Principal	131
A.2.2. Iniciar Sesión	131
A.2.2.1. Hacer Pedido Respecto a Otro Mes	131
A.2.3. Hacer Pedido	133
A.2.3.1. Buscar	134
A.2.3.2. Ver Catálogo	134
A.2.3.3. Añadir	134
A.2.3.4. Modificar	134
A.2.3.5. Eliminar	135
A.2.4. Enviar, Artículos No Pedidos y Creación del Fichero	135
A.2.5. Acceder a los Ajustes	135
A.2.6. Ajustar el Sistema	136
B. Aplicación Envío: Manual de Instalación y Usuario	139
B.1. Manual de Instalación	139
B.1.1. Introducción	139
B.1.2. Requisitos de Instalación	139
B.1.3. Instalación	140
B.2. Manual de Usuario	140
B.2.1. Pantalla de Inicio	140
B.2.2. Iniciar Sesión	141
B.2.3. Enviar Pedido	141

ÍNDICE DE FIGURAS

2.1. Imagen aérea de la fabrica	19
2.2. Recursos humanos	19
2.3. Estructura del departamento	21
3.1. Calendario	29
3.2. Diagrama de Gant	29
3.3. Fase de Inicio	29
3.4. Fase de elaboración	30
3.5. Fase de construcción	30
3.6. Fase de transición	30
4.1. Diagrama de casos de uso	44
4.2. Diagrama de actividad principal	52
4.3. Diagrama de actividad principal	53
4.4. Diagrama de actividad principal	54
4.5. Diagrama de actividad para hacer un pedido	55
4.6. Diagrama de actividad para enviar un pedido	56
4.7. Diagrama de actividad para leer un fichero	57
4.8. Diagrama de actividad para enviar articulos	58
4.9. Diagrama de dominio	59
5.1. Arquitectura lógica	65
5.2. Representación gráfica del MVC.	65
5.3. MVC	66
5.4. Descripción gráfica de DataMapper.	67
5.5. Clase controlador	73
5.6. Clase ControladorAdministracionSesion	74
5.7. Clase ControladorConfiguración	75
5.8. Clase ControladorConfirmación	75
5.9. Clase ControladorHacerPedido	76
5.10. Clase ControladorInicioSesión	77
5.11. Clase ControladorPantallaMenu	77
5.12. Clase ControladorVerCatalogo	78
5.13. Clase DataMapperModelo	79
5.14. Clase controlador	80
5.15. Clase controlador	81
5.16. Clase vista	82
5.17. Clase controlador	83

5.18. Clase controlador	84
5.19. Clase controlador	85
5.20. Clase ControladorLoading	85
5.21. Clase ControladorInicioSesion	86
5.22. Clase ControladorDatosPedido	88
5.23. Clase Hilo	88
5.24. Clase HiloPowerShell	89
5.25. Clase HiloInicioSesion	90
5.26. Clase HiloValidarCantidadPedida	91
5.27. Clase HiloFinDePedido	91
5.28. Clase HiloFinDePedido	101
5.29. Clase HiloFinDePedido	102
5.30. Clase HiloFinDePedido	103
5.31. Boceto de la pantalla de inicio de sesión	104
5.32. Boceto de la pantalla de añadir los artículos	104
5.33. Boceto de la pantalla de confirmar el pedido	105
5.34. Boceto de la pantalla del catálogo	105
5.35. Boceto de la pantalla de configuración	106
5.36. Boceto de la pantalla de enviar al servidor	106
6.1. Herramienta Invoke WebRequest	109
6.2. Formulario recibido con Invoke WebRequest	110
7.1. Estructura de directorios de la aplicación de pedidos del JU	113
7.2. Estructura de directorios ampliada	114
7.3. Estructura de directorios de la aplicación de envío de pedidos al centro de pedidos	114
7.4. Estructura de directorios ampliada	115
7.5. Hoja usuarios	116
7.6. Hoja fechas	117
7.7. Hoja presupuestos	117
7.8. Hoja referencias	118
7.9. Hoja detalles	118
7.10. Hoja ubicación	119
7.11. Hoja pedidos	119
A.1. Pantalla principal	131
A.2. Pantalla en la que se inicia sesión	132
A.3. Menú	132
A.4. Pantalla para seleccionar el fichero	133
A.5. Pantalla para realizar el pedido manualmente	133
A.6. Pantalla en la que se muestra el catálogo	134
A.7. Pantalla con los artículos pendientes	135
A.8. Pantalla para el envío del pedido	136
A.9. Pantalla para acceder a los ajustes	136
A.10. Pantalla para realizar ajustes	137
B.1. Pantalla de inicio	141
B.2. Pantalla de iniciar sesión	142
B.3. Pantalla de enviar pedido	142

ÍNDICE DE TABLAS

1.1. Fases del proyecto	15
3.1. Recursos humanos	27
3.2. Recursos hardware y software	28
3.4. Incumplimiento de plazos(retrasos)	31
3.5. Incumplimiento de plazos (adelantos)	32
3.6. Modificación de los requisitos	32
3.7. Pérdida de contenido	32
3.8. Avería	33
3.9. Ausencia en el puesto de trabajo	33
3.10. Falta de experiencia.	33
3.11. Calidad inadecuada	34
3.12. Falta de seguimiento	34
4.1. Participante nº1	36
4.2. Participante nº2	36
4.3. Participante nº3	36
4.4. Participante nº4	36
4.5. Objetivo nº1	37
4.6. Objetivo nº2	37
4.7. Objetivo nº3	37
4.8. RF nº1	38
4.9. RF nº2	38
4.10. RF nº3	38
4.11. RF nº4	38
4.12. RF nº5	39
4.13. RF nº6	39
4.14. RF nº7	39
4.15. RF nº8	39
4.16. RF nº9	40
4.17. RF nº10	40
4.18. RF nº11	40
4.19. RF nº12	40
4.20. RF nº13	41
4.21. RNF nº1	41
4.22. RNF nº2	41
4.23. RNF nº3	41
4.24. RNF nº4	42

4.25. RNF n°5	42
4.26. Actor n°1	42
4.27. Actor n°2	42
4.28. Actor n°3	43
4.29. Caso de uso n°1	43
4.30. Caso de uso n°2	45
4.31. Caso de uso n°3	46
4.32. Caso de uso n°4	46
4.33. Caso de uso n°5	47
4.34. Caso de uso n°6	47
4.35. Caso de uso n°7	47
4.36. Caso de uso n°8	48
4.37. Caso de uso n°9	49
4.38. Caso de uso n°10	50
4.39. Caso de uso n°11	50
4.40. Caso de uso n°12	51
5.1. Datos asociados al usuario	62
5.2. Datos asociados a las fechas	62
5.3. Datos asociados al presupuesto	62
5.4. Datos asociados a los artículos	62
5.5. Datos asociados a los artículos obligatorios	63
5.6. Datos de entrada	63
5.7. Clase controlador	73
5.8. Clase ControladorAdministracionSesion	73
5.9. Clase ControladorConfiguración	74
5.10. Clase ControladorConfirmación	75
5.11. Clase ControladorHacerPedido	76
5.12. Clase ControladorInicioSesión	76
5.13. Clase ControladorPantallaMenu	77
5.14. Clase ControladorVerCatalogo	78
5.15. Clase DataMapperModelo	79
5.16. Clase controlador	85
5.17. Clase ControladorLoading	85
5.18. Clase ControladorInicioSesion	86
5.19. Clase ControladorDatosPedido	87
5.20. Clase Hilo	88
5.21. Clase HiloPowerShell	89
5.22. Clase HiloInicioSesion	90
5.23. Clase HiloValidarCantidadPedida	90
5.24. Clase HiloFinDePedido	91
8.1. Inicio de sesión: campo usuario	121
8.2. Prueba usuario	122
8.3. Inicio de sesión: campo contraseña	122
8.4. Prueba contraseña	122
8.5. Hacer pedido: campo referencia	122
8.6. Prueba referencia	122
8.7. Hacer pedido: campo cantidad	122
8.8. Prueba cantidad	123

1.1. Visión General

La informática es la ciencia que tiene como objetivo estudiar el tratamiento automático de la información a través de los computadores. Denominamos “informatizar” a aplicar los métodos de la informática en un negocio, un proyecto, etc [1]. El principal fin que buscamos en este Trabajo de Fin de Grado es obtener una solución, “informatizando” un problema, el cual es descrito mas adelante. Para poder entender el estado actual de la computación en la sociedad actual, necesitamos echar la vista atrás y ver cuál ha sido la evolución de ésta desde su origen.

Una de las primeras herramientas que se inventaron para acelerar la velocidad de cálculo fue el ábaco, el cual fue inventado entre los años 2.700-2.300 A.C. El Mecanismo de Anticitera(200 A.C) es la primera computadora analógica mecánica conocida, artefacto que permitía calcular las posiciones astronómicas. Cerrado este periodo antiguo, no sería hasta la edad media cuando ingenieros musulmanes recuperaron los mecanismos de computación analógica, mecanismos como el Equatorium o el Astrolabio. Ya en el siglo XVII, Pascal construyó la primera máquina que permitía realizar sumas. Durante el siglo XIX se investigó en la disciplina de la lógica binaria, implementándolo en máquinas analógicas como telares programados mediante tarjetas perforadas. Este desarrollo permitió, ya en el siglo XX, el desarrollo de computadoras. Originalmente, el término computador se veía referenciado en una persona cuyo trabajo consistía en hacer cálculos. Gobiernos y centros de investigación disponían de muchas personas realizando esta labor. Durante este desarrollo, el término computador cambió, su nuevo significado se tornó a referenciar a cualquier máquina capaz de realizar el trabajo de un computador humano. Un último cambio importante fue la aparición de las máquinas digitales que guardaban información originalmente en relés y motores de diferencia. Las computadores avanzaron es su desarrollo, durante la segunda mitad del siglo XX, las computadoras se clasificaron en diferentes generaciones. Finalmente, en el día de hoy, el desarrollo de computadores cada vez más potentes ha permitido el desarrollo de nuevas ramas de investigación como la inteligencia artificial, el cómputo científico, las redes de computadoras, etc [2].

Durante este breve repaso de la computación, apreciamos como ésta ha pasado por diferentes etapas. Desde el final de la Segunda Guerra Mundial hasta el día de hoy podemos determinar una última etapa, coincidiendo con la aparición de los computadores digitales. En esta última etapa el mundo ha sufrido una informatización, definición que se daba al principio de esta sección. Dentro del día a día de la humanidad, se ha pasado de realizar tareas de forma manual a realizarla con la ayuda de ordenadores. Hoy en día, casi todo en lo que se pueda pensar tiene un ordenador o un procesador haciéndolo funcionar. Se ha realizado un evolución en la manera de ser de las cosas, produciendo una mayor producción a cambio de un menor coste.

La realización de este proyecto trata sobre la realización de un proceso de informatización con el cual se espera obtener un ahorro en coste y tiempo respecto al proceso primitivo.

1.2. Motivación

A continuación presento los motivos que me llevaron a realizar el Trabajo de Fin de Grado sobre el tema que se presenta en las siguientes secciones.

En el último curso del grado en Ingeniería Informática cursé la asignatura de “Prácticas de empresa” en la factoría Renault de Palencia. El tutor que me fue asignado por parte de la escuela fue D^o Jesús María Vegas Hernández. Al poco de empezar su realización, se vio factible la posibilidad de plasmar el desarrollo de éstas en el Trabajo de Fin de Grado. Tras hablarlo con el tutor, decidí que sería buena idea. En ese momento, mi tutor de prácticas se convirtió también en mi tutor de TFG.

Las prácticas han sido realizadas en la factoría de Renault en Palencia, concretamente, dentro del departamento de Pintura. Desde un principio tenía claro en qué tipo de ambiente me gustaría realizar las prácticas. La idea que tenía en mente era una empresa grande, donde además de aprender el ámbito informático, se aprendiera también como es el funcionamiento de una gran empresa, dejando a un lado la fabricación del producto, ver como eran las fases de desarrollo, mejora, calidad y demás entornos que rodean la producción. Cuando se me explicó cuál era mi labor a realizar en esta organización vi que la idea de realizar sobre ello el TFG era una oportunidad muy grande y además con gran interés personal.

La labor de diseñar un software que cumpliera con los objetivos solicitados, la labor de investigar el entorno de trabajo y de las limitaciones que me rodeaban, la oportunidad de diseñar un proyecto según mis criterios de diseño, el trato con la gente para saber su punto de vista o sus ideas y la idea final de ofrecer un producto cuyo fin es mejorar el funcionamiento de un departamento eran puntos vitales que hicieron mucho más fácil y agradable la labor de realizar estas prácticas y reflejarlo paralelamente en este proyecto.

Como último punto, a lo largo de todos estos años de aprendizaje en la Escuela Técnica Superior de Ingeniería Informática se han enseñado multitud de campos de conocimiento relacionados con la informática, unos más interesantes y otros menos. Estas prácticas eran una oportunidad para poder seguir profundizando en estos campos que me llamaron la atención como la programación, el desarrollo de interfaces gráficas o el desarrollo web.

1.3. Problema Existente

En esta sección se presenta cual era el problema existente en el departamento. Posteriormente, se describe cual era la idea que buscaban los clientes como solución al problema. En este caso los clientes eran los responsables de ese aspecto en el departamento.

Para la realización de coches son empleados, dentro del departamento de Pintura, muchos materiales aplicados o no sobre el coche. Una pequeña muestra de ellos son: lijas, platos de poliéster, cinta de carrocería, pasta de pulir, protector, boquillas, pinceles, pistolas, etc. Para poder realizar correctamente los coches deben disponer de estos materiales, es por ello que una vez al mes se realiza un pedido al coordinador. El pedido es realizado por cada JU y consiste en el relleno de plantilla Excel y su posterior envío por correo.

El proceso manual es el siguiente, el JU rellena las columnas obligatorias a mano (UET, referencia, denominación, cantidad, fecha pedido, precio unitario y coste real) y lo envía por correo

electrónico al coordinador. El proceso de relleno contiene multitud de fallos y errores que suponen un gasto de tiempo por parte del coordinador en su proceso de corrección. Errores como referencias inexistentes, cantidades incompatibles o presupuestos fuera de los asignados son continuos fallos que corregirlos supone un tiempo considerable.

Una vez recibido y corregido el pedido, éste tiene que ser enviado al departamento encargado de proveer de materiales a la factoría. Este departamento, el cual llamaremos “centro de pedidos” dispone de una página web para hacer los pedidos.

El proceso que se sigue es el siguiente:

1. Iniciar sesión
2. Indicarle que vamos a realizar un pedido
3. Bucle
 - 3.1. Enviar referencia y cantidad
 - 3.2. Asignar al artículo enviado el centro de gastos
4. Fin de bucle
5. Confirmar pedido y enviar

La información recibida por el JU se encuentra en formato Excel pero la página que pide los datos de los artículos que se quieren pedir es diferente, los datos son introducidos de manera manual. Existe cierta inconsistencia en el formato de los datos, por lo que se pierde bastante tiempo leyendo el fichero Excel y escribiéndolo en los campos de la página web. El flujo de datos existente a lo largo de todo el proceso sufre alteraciones y cuellos de botella debido a la incompatibilidad entre departamentos.

Tanto corregir el pedido como enviarlo mediante la web es un proceso que lleva un tiempo considerable, a la vez que es repetitivo y que a su vez puede conllevar a nuevos errores. En general vemos un proceso lento, en el que el flujo de información varía y sigue diferentes e incompatibles formatos. ¿Qué se va a hacer? Se va a proceder a resolver este problema mediante una herramienta software. Esta herramienta será el resultado de realizar un proyecto software.

Una primera restricción añadida por los clientes fue que debían desarrollarse dos sistemas informáticos totalmente independientes. Uno que de solución a los pedidos del JU hacia el coordinador y otro que permita el envío por parte del coordinador (que en este caso tendrá un rol de “responsable”). Una ventaja clara de que ambas partes se encuentren separadas es la independencia en la disponibilidad, un fallo en una no tiene por que afectar a la otra.

Dicha herramienta estará formada por dos aplicaciones:

- Aplicación Formulario: la cual va a permitir realizar pedidos al JU. Éste seleccionará los artículos que desee pedir. Una vez finalizado el proceso de selección, se generará un fichero con los artículos que se desean.
- Aplicación Envío: dicha aplicación permitirá el envío automático de los pedidos al centro de pedidos. La aplicación leerá el fichero creado por la herramienta anterior y enviará automáticamente los datos.

1.4. Objetivos

Los objetivos principales que se van a tratar de alcanzar en este Trabajo de Fin de Grado son:

- Conseguir desarrollar un software que permita resolver los problemas existentes actualmente, logrando que este software responda a todas las necesidades del usuario, que sea usable e íntegro. A su vez, debe de ser eficiente, para no saturar los recursos del sistema, y por supuesto tiene que ser seguro.
- Conseguir desarrollar un software que permita ahorrar tiempo en el proceso de realización, corrección y envío del pedido.
- Desarrollar un software con una alta disponibilidad para el usuario.
- El software desarrollado debe facilitar la tarea del empleado, no debe dificultarla o entorpecerla.
- Profundizar en el conocimiento de las herramientas informáticas con las que se van a resolver los problemas que se han planteado.
- Realizar una planificación completa y documentada del proyecto, en las que se identifiquen las diferentes etapas e informaciones asociadas a cada una de ellas.
- Simular un proceso de trabajo profesional. Durante el proyecto, se tratará de captar cuales son las necesidades de la organización y ofrecer una solución al cliente. Este proceso debe de ser similar al de un trabajador profesional con las mismas garantías de calidad y seguridad.

1.5. Metodología

Para la realización de este proyecto se ha seguido el método de Proceso Unificado, dado que se conocía previamente y ha sido estudiado y aplicado en diversas asignaturas, por lo que teniendo un conocimiento previo sobre este método servirá para obtener un proyecto más completo.

El Proceso Racional Unificado o RUP (por sus siglas en inglés de Rational Unified Process) es un proceso de desarrollo de software. El proceso RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Está basado en componentes, lo que indica que el sistema a construir estará formado por componentes software interconectados a través de interfaces. Utiliza UML como lenguaje de modelado [3]. Los tres elementos claves que resumen el proceso unificado son:

- Iterativo e incremental: El desarrollo del proyecto es dividido en 4 fases. En cada fase se desarrollan actividades diferentes. Cada fase está constituida por una o varias iteraciones, en las cuales se desarrolla un proceso en cascada. Cuando una iteración acaba provoca un incremento en el producto que se ha desarrollado. Este incremento es usado como base en la siguiente iteración, así repetidamente.
- Dirigido por casos de uso: Los casos de uso permiten saber cual es la funcionalidad que debe desarrollar el sistema. Un buen análisis de casos de uso permitirá definir el dominio del proyecto y ofrecer un mejor producto al cliente.
- Centrado en la arquitectura: Centrado en la arquitectura significa que el trabajo de desarrollo se centra en obtener un patrón que dirigirá la construcción del sistema en las primeras fases.

1.6. Plan de Fases

Utilizando el modelo de Proceso Unificado, obtenemos las siguientes fases.

Fases	Producto resultante
Fase de comienzo	<ul style="list-style-type: none"> ■ Visión general de los requisitos y restricciones fundamentales. ■ Primera versión del modelo de casos de uso. ■ Primera aproximación al diseño arquitectónico.
Fase de elaboración	<ul style="list-style-type: none"> ■ Modelo completo de negocio. ■ RF y RNF revisados. ■ Descripción de la arquitectura software. ■ Revisión del plan de gestión de riesgos.
Fase de construcción	<ul style="list-style-type: none"> ■ Sistema software ejecutable en versión beta. ■ Modelo de sistema completo. ■ Iniciación de los manuales de usuario y de instalación.
Fase de transición	<ul style="list-style-type: none"> ■ Software completo. ■ Última versión de los manuales de usuario y de instalación.

Tabla 1.1: Fases del proyecto

1.7. Estructura del Documento

En las siguientes líneas se explica cuáles son los capítulos y la estructura que dan forma al presente documento. El documento está estructurado en 9 capítulos de los que se hace un breve resumen a continuación:

- **Capítulo 1:** Capítulo presente en el cual se hace una introducción a lo que se persigue en el presente documento.
- **Capítulo 2:** Puesta en situación del lugar y el entorno de la realización del proyecto. Al final se hace una descripción del problema a solucionar.
- **Capítulo 3:** Se describe en profundidad el problema a resolver y se presenta la planificación que se va a seguir durante el proceso de creación del sistema.
- **Capítulo 4:** Se profundiza en el análisis y en los requisitos que debe cumplir y satisfacer la aplicación a desarrollar.
- **Capítulo 5:** Se realiza un diseño de la estructura y se presentan los diferentes modelos del sistema.
- **Capítulo 6:** Se describe la tecnología empleada que con la que se van a implementar los modelos ya presentados.
- **Capítulo 7:** Se muestra el resultado de la implementación y cómo ha sido realizada.

- **Capítulo 8:** Se muestra sobre que sistema se ha instalado la aplicación y las pruebas que se realizan sobre éste.
- **Capítulo 9:** Finalmente se presentan las conclusiones y ampliaciones futuras sobre el sistema desarrollado.

CAPÍTULO 2

MARCO CONCEPTUAL

En este nuevo punto de la memoria, se procederá a dar a conocer la empresa Renault S.A, se describirá brevemente la historia de esta compañía automovilística y cuál es su situación actual en el mercado. Describiremos como es el funcionamiento de la factoría Renault en Palencia, la estructuración que posee y los diferentes departamentos. Como punto más importante de esta sección, se describirá cual es el problema existente en Renault y cual es la solución que se va a desarrollar para subsanar dicho problema.

2.1. ¿Qué Es Renault?

Renault es un fabricante de vehículos de origen francés que produce vehículos de gama comercial. Fue fundada en 1899 por Louis Renault. Actualmente posee una cantidad de 122 000 empleados (2010). Su presidente es Carlos Ghosn y su sede histórica está en Boulogne Billancourt en Paris (Francia). Cuenta con un centro de ingeniería y de I+D, el technocentro, situado en Guyancourt, en las afueras de París [7]. Desde hace varios años, Renault está integrado dentro del “Grupo Renault”, conformado por Renault, Dacia y Renault Samsung Motors. A su vez, Renault posee una participación en Nissan del 44% y ambos sostienen la llamada “alianza”, con la cual intercambian avances y tecnologías manteniéndose totalmente independientes. También posee un acuerdo de colaboración con Mercedes.

En el año 2016, Grupo Renault matriculó un total de 3.182.625 vehículos, creciendo un 13.3% respecto al año anterior. En España estas ventas registraron un cifra de 170 272 vehículos, dando como resultado un crecimiento del 12.9% [8].

La fábrica Renault en Villamuriel del Cerrato, Palencia, que es la que me acogió para realizar las prácticas de empresa, registro una cifra record de 332.000 vehículos en el año 2016, lo que supone un crecimiento del 58% respecto al año anterior. Inaugurada en 1978 cuenta con una superficie total de 323.081 m² de superficie construida, en la que se producen el Mégane y Kadjar [9].

2.2. Historia de Renault

Fue fundada por Louis Renault en el año 1898, en colaboración con sus hermanos Marcel y Fernand, ese mismo año vendieron su primer coche. En el año 1899, un total de 60 empleados fabricaron 71 unidades. No fue hasta 1903 cuando empezaron a fabricar sus propios motores y en 1905 recibieron un encargo de 250 taxis. Empezaron así con la producción en línea y la política taylorista (división del trabajo). Renault fabricó vehículos para la I Guerra Mundial como ambulancias, camiones, carros, etc [10].

Hasta antes de la II Guerra Mundial, Renault siguió con un ritmo imparable de crecimiento. Desarrollando nuevos vehículos, mejorando la tecnología y explorando nuevos negocios como la aviación, el ferrocarril y material agrícola. Durante la II Guerra Mundial, le son arrebatadas las factorías a Louis Renault durante la invasión Alemana. Obligado a producir para ellos, sería años más tarde juzgado y mandado a prisión, donde falleció.

Durante la segunda mitad del siglo XX, Renault sigue creciendo. Se produce su expansión a varios países, entre ellos España. Modelos famosos como el R3, R4, R6 y R12 ocuparon el mercado a la vez que Renault triunfaba en los rallyes. No es hasta la década de 1980 cuando Renault decae en ventas y empieza a producir pérdidas. Tras una privatización parcial de la empresa, nuevos directivos y el lanzamiento de nuevos modelos hicieron que ya en la década de 1990 Renault recuperara el terreno perdido en el mercado.

Ya en el XXI, una apuesta fuerte por la internacionalización y la colaboración con otras empresas finaliza con la llamada alianza entre Renault y Nissan y la creación del Grupo Renault, ofreciendo así una apuesta por los mercados mundiales [11].

2.3. Estructura de la Fábrica

En esta sección, se describe muy brevemente cual es la estructura de la fábrica.

La fabricación de un coche, visto desde un punto global, es un proceso en serie. Un coche comienza con el moldeo de la chapa a partir de grandes bobinas y su posterior soldadura. Una vez conformada la chapa, esta recibe el mástico y la pintura. A continuación pasa a montaje, donde se colocan todas las demás partes del coche. Parte de las piezas del coche son creadas allí y otras son montadas en otras fábricas de Renault, como los motores o las cajas de cambio, o en empresas externas. Esta descripción que acabamos de hacer es muy vaga y superficial, pero suficiente para ponernos en situación. En la imagen 2.1 se ve cómo es la distribución de la fábrica(1:Embutición y chapa; 2:Pintura; 3:Montaje; 4:Parque de proveedores).

2.4. Estructura del Departamento

En los siguientes párrafos, se tratará de describir cual es el ámbito de aplicación de las herramientas software que se van a desarrollar a lo largo de este proyecto.

Renault Palencia actualmente dispone de más de 4000 empleados distribuidos en cuatro turnos de trabajo, si nos centramos concretamente en el departamento de Pintura, la cifra ronda los 350 empleados. Intentar explicar la organización total de este departamento sería una tarea larga y tediosa, además de no estar gran parte relacionado con nuestro trabajo. Por ello se describe en la figura 2.2 un organigrama con las partes que nos interesan para comprender nuestro contexto.



Figura 2.1: Imagen aérea de la fabrica

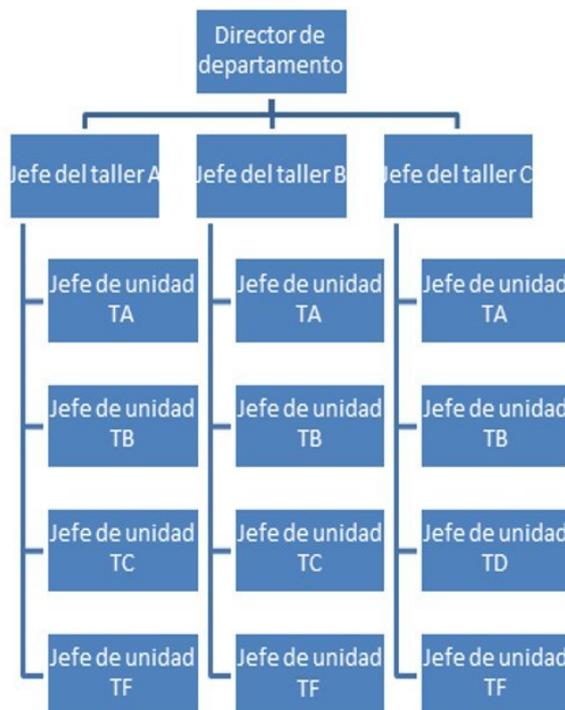


Figura 2.2: Recursos humanos

En la figura 2.2 podemos encontrar la composición más sencilla de cómo está estructurado internamente el departamento de pintura. Esta composición es muy vaga e imprecisa y no contempla todos los aspectos que nos interesan. Pero antes de dar una versión más amplia de la estructura, vamos a explicar esta figura para poder hacernos una primera idea.

El departamento de pintura está compuesto por un mando superior, este mando es el director de departamento. Este escalón engloba toda la dirección y responsabilidad de lo que ocurre y sucede dentro del departamento. La fabricación de vehículos está compuesta de multitud de pasos, estos se agrupan en departamentos (embutición, soldadura, pintura, montaje, etc.) y a su vez cada departamento está compuesto por diferentes talleres (dentro de pintura: másticos, terminación,...). Cada taller está dirigido por un jefe de taller, el cual tiene la responsabilidad de organizar la actividad, controlar la calidad, evaluar la producción, proponer mejoras en la fabricación, etc.

Un escalón por debajo se encuentran **los jefes de unidad, JU** de manera abreviada. Dentro de cada taller existen diferentes tareas, llevadas a cabo por un grupo de empleados bajo la supervisión de un JU. Los jefes de unidades son los encargados de pedir material mensualmente. Estos materiales son de muy diferente tipo, pueden ser ceras, lijas, bayetas, guantes, ropa de trabajo, etc. Existe un JU por cada turno de trabajo, habiendo tres talleres a cuatro turnos cada uno, obtenemos un total de 12 jefes de unidad. **Los JU's son usuarios finales del software que se va a desarrollar.**

La descomposición que acabamos de dar es una descripción muy directa, una descripción relacionada totalmente con la producción. Pero la producción no es una tarea sencilla y conlleva la coordinación con otras tareas muy importantes como son la gestión económica, el área encargada de los automatismos, la sección de ingeniería, la sección de mantenimiento o la sección de calidad. De una manera resumida, sería muy complicado intentar ilustrar una completa descripción de todo el departamento, por ello ampliaremos la figura 2.2, para intentar comprender la parte que nos atañe.

Como bien acabamos de mencionar, existen otras tareas indispensables para el correcto funcionamiento de la factoría, ahora mismo solo nos interesa conocer una parte de la gestión económica y de los recursos humanos. Vamos a ver en la figura 2.3 una idea más ampliada que la presentada en la figura anterior.

Podemos observar resaltado en color rojo, cuales son las secciones de las que acabamos de hablar. Entre ellas debe existir una comunicación constante y de vital importancia, ahora veremos por qué. Esa comunicación pasa explícitamente por dos personas, **Cristina Cuende y Ángel Mayordomo, usuarios finales del software que se va a desarrollar.**

El área de gestión económica dispone de toda la información sobre la economía de la empresa, derivado del número estipulado de vehículos que se van a producir, obtienen multitud de información sobre los aspectos financieros. Estos datos están totalmente relacionados con la producción y los JU's, desde el dinero disponible para materiales, dinero disponible para ropa, dinero disponible para formaciones, etc. Lo mismo con los jefes de taller, dinero disponible para mejoras, pruebas de calidad, contrataciones, obras, equipos informáticos, etc. Esta coordinación pasa por los antes mencionados. Para poder resumir más fácilmente su labor, a partir de ahora, vamos a referirnos a ellos como **coordinadores.**



Figura 2.3: Estructura del departamento

2.5. Entorno Informático de Renault

2.5.1. Descripción

Renault cuenta con un entorno informático acorde al tamaño de la empresa, en este caso podemos catalogar como un entorno grande y complejo. Hablar en profundidad de este entorno informático es algo muy amplio de describir además de no ser importante para nuestro trabajo. A pesar de ello, a continuación se presenta una breve descripción para tener cierto conocimiento.

Principalmente, podemos diferenciar entre dos clases de sistemas informáticos:

- Sistema informático para la producción.

Dentro de este grupo encontramos un amplio sistema compuesto por multitud de ordenadores en multitud de formas. Este sistema está relacionado íntegramente con la producción. Cuenta con una enorme variedad de sistemas, desde ordenadores de sobremesa, ordenadores industriales, sistemas empotrados, etc. Además cuenta con un alto nivel de seguridad y restricciones a su acceso. La producción diaria depende de estos ordenadores y un fallo o mal funcionamiento de un sistema puede producir retrasos en la producción con enormes costes laborales y económicos. Este sistema no es interesante para nuestro proyecto.

- Ordenadores no relacionados con la producción.

Este sistema engloba el resto de ordenadores que puedan encontrarse dentro de la factoría. Son ordenadores de sobremesa, en los que corre un sistema Windows. El sistema cuenta con una virtualización, es decir, realmente no se está ejecutando en el mismo ordenador sino que el trabajo se realiza en un ordenador central. Recordemos que la virtualización es una técnica informática que aprovecha los recursos físicos de los que una máquina anfitriona (host), la cual desconocemos su ubicación y sus detalles, dispone para poder ejecutar en ella otros u otros sistemas completamente independientes y funcionales (guests), con todas las consecuencias que ello conlleva; consumo de memoria, tiempo de procesador, almacenamiento en disco y en general de todos aquellos dispositivos de la máquina anfitriona que vayan a ser compartidos por las máquinas denominadas virtuales (VM).

Para poder acceder a uno de estos se debe disponer de conexión a la red y de una cuenta de usuario. Estos ordenadores son usados para temas relacionados indirectamente con la producción, como es la gestión económica, recursos humanos, desarrollo de automatismos, desarrollo de prototipos, comunicaciones, ingeniería, etc.

El sistema cuenta con numerosas limitaciones para el usuario, no puede instalar programas, solo puede ejecutar aquellos ya instalados, ni acceder a Internet. Estas restricciones son debidas a motivos de seguridad. Se cuenta con un sistema de ficheros compartido, cada departamento tiene una carpeta dentro de ella que derivan en muchas más. Cada usuario tiene acceso a la parte que le interesa para su trabajo y carece de permisos para acceder a las carpetas que son ajenas a su competencia. Un detalle importante para nuestro proyecto es que tienen instalado el lenguaje Java 7 en todas las máquinas.

Renault cuenta con una serie de servidores, cuya ubicación se desconoce, en los cuales ofrece una serie de herramientas vía web. Accediendo al navegador, es posible acceder a ellas. Estos servidores se encuentran en la intranet de la empresa, por lo que no es posible acceder desde fuera, salvo por VPN¹, sistema con el que cuenta Renault.

Durante las últimas semanas de mi estancia en Renault se llevaron a cabo nuevas adquisiciones en sistemas informáticos, estos sistemas eran tablets. Renault adquirió una serie de tablets para que los empleados pudieran usarlas en las labores que realizaran.

Debido a que el sistema informático es muy grande y la cantidad de usuarios también, Renault cuenta con un servicio telefónico de asistencia. En el caso de que algo no funcione correctamente o de problemas, existe un departamento de ayuda telemática. Vía telefónica resuelven tus dudas y ellos mediante acceso a través del ordenador, te solucionan los problemas o te envían un técnico en caso de que no se haya podido solucionar.

2.5.2. Seguridad

Como ya se ha mencionado anteriormente, la seguridad es un tema fundamental para la empresa. Una alto nivel de seguridad reduce las posibilidades de un ataque externo pero también de un ataque interno, intencionado o no intencionado. Renault, en el entorno informático, ofrece una serie de recursos a sus empleados para que puedan desarrollar su labor de una manera cómoda, pero no puede permitir que un empleado exceda de sus competencias o pueda suponer una brecha en la seguridad del sistema por lo que un usuario está muy acotado en sus labores. Existe también seguridad en la red existente, limitando ciertos puertos o direcciones. Un filtro bloquea, según un criterio, aquellas páginas no apropiadas. No solo se establece seguridad a nivel software sino también a nivel personal, todo ordenador que entre o salga de la factoría es registrado su número de serie.

La conexión a la red interna de Renault está muy controlada. Cada ordenador está asociado a una única clavija de acceso a la red, por lo que es imposible, sin permiso, realizar cambios. Además, cada asociación clavija-ordenador está registrada en un sistema, donde debe de ser dado de alta para que la conexión funcione. El nivel de seguridad asociado a las contraseñas es muy alto. Éstas deben ser renovadas periódicamente de manera obligatoria, tiene un tamaño mínimo, no deben ser iguales que las anteriores ni contener palabras clave como “renault”, “Palencia”, “coche”, etc. Por lo tanto, se aprecia que existe una política de seguridad bastante restrictiva.

Pero toda esta seguridad no es suficiente para evitar ataques externos, como hecho destacado

¹VPN: Una red privada virtual, Virtual Private Network, es una tecnología de red de computadoras que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet. Permite que la computadora en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada con toda la funcionalidad, seguridad y políticas de gestión de una red privada. Esto se realiza estableciendo una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado o la combinación de ambos métodos.

podemos señalar que durante mi periodo de prácticas en la factoría se produjo el famoso ataque conocido como “WannaCry”². Éste sucedió el 12 de Mayo del 2017 y afectó a varios ordenadores de la producción de varios departamentos de la factoría. Como consecuencia hubo alteraciones en el orden normal de la factoría [12].

²WanaCry es un ataque informático, catalogado como ransomware, que usan el criptogusano conocido como WannaCry dirigidos al sistema operativo Windows de Microsoft. Durante el ataque, los datos de la víctima son encriptados, y se solicita un rescate económico pagado con la criptomoneda Bitcoin, para permitir el acceso a los datos.

3.1. Introducción

A continuación se presenta la planificación del proyecto, la cual tiene una serie de funciones: declarar el orden de las tareas y estimar el tiempo necesario para llevarlas a cabo. Una buena planificación está basada en una buena descripción y comprensión de los objetivos que debe de alcanzar el proyecto. Para ello se deben estimar los tiempos, identificar los hitos, encadenar las actividades necesarias para alcanzar los hitos ya definidos y finalmente realizar una planificación temporal. Para representar la planificación se usarán diagramas de Gantt, que sirven para representar de una manera simple e intuitiva. La metodología empleada, como ya se ha descrito anteriormente, ha sido RUP.

3.2. Descripción del Sistema a Desarrollar

En las siguientes secciones se va a realizar una única planificación para los dos sistemas software que se van a desarrollar. Para poder hacer una buena planificación es necesario saber que se desea desarrollar. Para ello, a continuación, se va a proceder a describir cual es la funcionalidad que debe reunir dichas aplicaciones. Como ambas aplicaciones deben estar separadas, primero se describe una y después la otra.

- Envío de pedidos del JU al coordinador (**Aplicación Formulario**):

Se necesita un sistema que permita, mediante interacción con su interfaz gráfica, seleccionar unos artículos y enviárselos al coordinador. Los pedidos los hacen exclusivamente los JU, siendo necesario un sistema que controle quien puede acceder a la aplicación. Una vez validado el inicio de sesión del usuario permitimos que haga un pedido. Para ello el sistema debe de disponer de información, tales como referencias, cantidades y precios de los artículos. Los usuarios tienen asignado una cantidad de dinero para gastar y el sistema debe gestionar que los pedidos se hacen acordes a esa cantidad, impidiendo gastos superiores al dinero disponible. También se hacen pedidos de artículos obligatorios, siendo esto controlado y validado por el sistema. No se permiten hacer más de un pedido al mes, salvo casos exclusivos bajo permiso y configuración del administrador. Para facilitar la tarea, el usuario puede ver un listado de todos los artículos que se encuentran disponibles. Solo se permiten hacer pedidos un periodo de días al mes, por lo que el resto del mes no se debe poder realizar pedidos. Durante el proceso de añadir artículos, el usuario puede equivocarse, por lo que es necesario permitir que se modifiquen los artículos que ya se han añadido al listado.

Una vez realizado el pedido, si todo es correcto, se procede a generar un fichero con la

información correspondiente, lo podemos llamar “resguardo del pedido”. El fichero albergará información de los artículos que el cliente desea pedir. Una vez generado generado dicho fichero es guardada en una carpeta interna para que el responsable pueda acceder a ellos. Dicho fichero se debe guardar bajo contraseña, para evitar así su manipulación. Una copia, sin contraseña, es ofrecida al usuario que puede guardarla donde desee. Como alternativa al proceso de buscar artículos, se debe permitir que el usuario pueda repetir un pedido anterior. Para ello el usuario debe ofrecer un fichero de resguardo de un pedido anterior.

El coordinador puede realizar labores de configuración sobre la aplicación, como permitir hacer más de un pedido al mes o configuración de contraseñas.

- Envío de pedidos del responsable al centro de pedidos (**Aplicación envío**):

En esta segunda aplicación se procede al envío del fichero, generado en la carpeta del responsable, al servidor del centro de pedidos. Este sistema debe interactuar con la página web del centro como si lo hiciera una persona. En este aspecto, el sistema no controla por si mismo la validación de los datos sino que lo hace según la respuesta que de la página web en su validación de datos, el sistema solo se limita a transmitir esta información.

Para poder enviar un pedido, se debe iniciar sesión con el usuario y contraseña de un usuario con permisos para hacerlo en la página web. El siguiente paso es obtener los datos del pedido, para ello el usuario, en este caso el responsable busca el fichero generado anteriormente. Una vez reunidos los datos, el sistema establece conexión con la página web e inserta los datos automáticamente, de manera similar a una persona. Una vez finalizado el envío, muestra el resultado en pantalla ofrecido por la página web con la información.

Durante el tiempo que tarde el envío del pedido, el sistema mostrará por pantalla el proceso de envío de este. Con los artículos que se van pidiendo y los resultados que devuelve la página web. Si en algún momento se produce un error, el proceso de envío de debe cancelar.

3.3. Recursos

A continuación se describen cuales son los recursos, tanto humanos, físicos y software, de los que se disponen para realizar el proyecto.

3.3.1. Recursos Humanos

El desarrollo de este proyecto ha sido realizado por la misma persona, el autor del mismo, pero fue necesario la realización de varios perfiles profesionales para el desarrollo de éste. Estos perfiles también han sido realizados por el autor del proyecto. A continuación se presenta en forma de tabla:

Rol	Responsabilidades
Responsable del proyecto	<ul style="list-style-type: none"> ■ Desarrollar y dar formato al documento final. ■ Responsabilidad de coordinar las fases de acuerdo al calendario establecido.
Analista	<ul style="list-style-type: none"> ■ Documentar el trabajo realizado en su fase. ■ Extracción de RF y RNF. ■ Determinar los límites y el dominio abarcados por el proyecto. ■ Realización de los casos de uso.
Diseñador	<ul style="list-style-type: none"> ■ Documentar el trabajo realizado en su fase. ■ Elaboración de la arquitectura lógica y física. ■ Realización del diagrama de diseño. ■ Realización de los diagramas de secuencia. ■ Realización de los esquemas de persistencia.
Desarrollador	<ul style="list-style-type: none"> ■ Documentar el trabajo realizado en su fase. ■ Realización del código fuente y su depuración.
Evaluador	<ul style="list-style-type: none"> ■ Documentar el trabajo realizado en su fase. ■ Determinar las pruebas a realizar. ■ Realizar de las pruebas diseñadas.

Tabla 3.1: Recursos humanos

3.3.2. Recursos Hardware y Software

Para la realización del proyecto fue necesario una serie de herramientas. Éstas son hardware, y como no, software.

Recurso	Tipo	Detalles
Ordenador	Hardware	Ordenador portátil en el que se desarrolla el proyecto.
Ordenador	Hardware	Ordenadores de sobremesa en los que se accede al programa.
Microsoft Excel	Software	Programa para poder ver los pedidos realizados y contener la información.
MS Project	Software	Programa para poder realizar la planificación y gestión de recursos.
Navegador	Software	Navegador para poder realizar el diseño Latex del documento.
Astah Profesional	Software	Herramienta software para poder la construcción de los casos de uso, secuencia, actividad, etc.
NetBeans IDE 8.2	Software	Herramienta software para poder desarrollar el código fuente.

Tabla 3.2: Recursos hardware y software

3.4. Plan de Despliegue

Tras la elaboración final del producto se necesitan una serie de elementos, tanto hardware, software, permisos y humanos, para desplegar y hacer que el software esté disponible para su uso.

- Hardware
 - Como mínimo un ordenador desde el que ejecutar el programa.
- Software
 - Java 7.
 - Microsoft Excel.
 - Carpetas compartidas.
 - Cuentas de usuario.
- Permisos
 - Permisos de seguridad para el acceso a esas carpetas.
- Humanos
 - Un administrador que gestiona la aplicación, manteniéndola y se encarga de gestionar los pedidos.

3.5. Asignación de Recursos y Calendario

Fase	Nº de iteraciones	Duración
Inicio	1	2 semana
Elaboración	2	3 semanas
Construcción	2	8 semanas
Transición	1	2 semanas

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Comienzo del proyecto	0 días	lun 30/01/17	lun 30/01/17	
▲ Fase de inicio	14 días	lun 30/01/17	jue 16/02/17	1
▸ 1ª Iteración	14 días	lun 30/01/17	jue 16/02/17	1
▲ Fase de elaboración	21 días	vie 17/02/17	vie 17/03/17	2
▸ 1ª Iteración	10 días	vie 17/02/17	jue 02/03/17	3
▸ 2ª Iteración	11 días	vie 03/03/17	vie 17/03/17	8
▲ Fase de construcción	60 días	lun 20/03/17	vie 09/06/17	4
▸ 1ª Iteración	30 días	lun 20/03/17	vie 28/04/17	15
▸ 2ª Iteración	30 días	lun 01/05/17	vie 09/06/17	23
▲ Fase de transición	10 días	lun 12/06/17	vie 23/06/17	22
▸ 1ª Iteración	10 días	lun 12/06/17	vie 23/06/17	29

Figura 3.1: Calendario

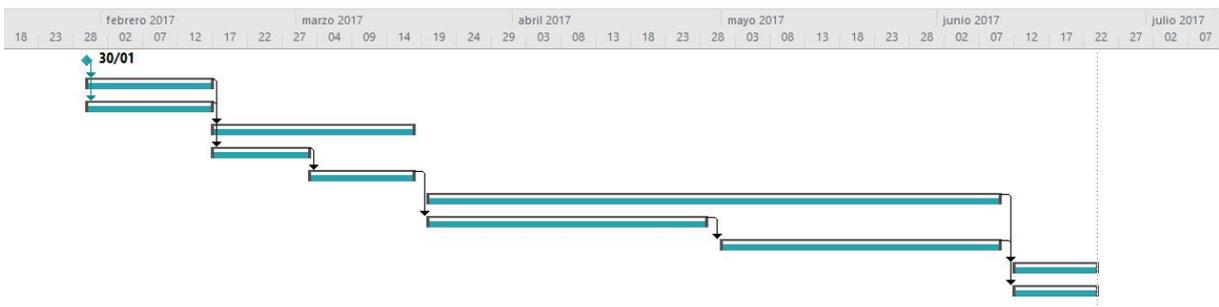


Figura 3.2: Diagrama de Gant

3.6. Plan de Proyecto

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
Comienzo del proyecto	0 días	lun 30/01/17	lun 30/01/17		
▲ Fase de inicio	14 días	lun 30/01/17	jue 16/02/17	1	
▲ 1ª Iteración	14 días	lun 30/01/17	jue 16/02/17	1	
▲ Plan de desarrollo del proyecto	14 días	lun 30/01/17	jue 16/02/17	1	
Determinar el alcance del proyecto	6 días	lun 30/01/17	lun 06/02/17		Santos Ángel Pardo
Elaboración del plan de proyecto	8 días	mar 07/02/17	jue 16/02/17		Santos Ángel Pardo

Figura 3.3: Fase de Inicio

4 Fase de elaboración	21 días	vie 17/02/17	vie 17/03/17	2	
4 1ª Iteración	10 días	vie 17/02/17	jue 02/03/17	3	
Documento de análisis	2 días	vie 17/02/17	lun 20/02/17		Santos Ángel Pardo
Creación de casos de uso	3 días	mar 21/02/17	jue 23/02/17	9	Santos Ángel Pardo
Diagramas de secuencia	3 días	vie 24/02/17	mar 28/02/17	10	Santos Ángel Pardo
Descripción hardware	0,5 días	mié 01/03/17	mié 01/03/17	11	Santos Ángel Pardo
Descripción software	0,5 días	mié 01/03/17	mié 01/03/17	12	Santos Ángel Pardo
Planificación de la 2ª iteración	1 día	jue 02/03/17	jue 02/03/17	13	Santos Ángel Pardo
4 2ª Iteración	11 días	vie 03/03/17	vie 17/03/17	8	
Revisión documento de análisis	3 días	vie 03/03/17	mar 07/03/17		
Revisión de casos de uso	3 días	mié 08/03/17	vie 10/03/17	16	Santos Ángel Pardo
Revisión de diagramas de secuencia	3 días	lun 13/03/17	mié 15/03/17	17	Santos Ángel Pardo
Revisión de descripción hardware	0,5 días	jue 16/03/17	jue 16/03/17	18	Santos Ángel Pardo
Revisión de descripción software	0,5 días	jue 16/03/17	jue 16/03/17	19	Santos Ángel Pardo
Panificación fase de construcción 1ª iteración	1 día	vie 17/03/17	vie 17/03/17	20	Santos Ángel Pardo

Figura 3.4: Fase de elaboración

4 Fase de construcción	60 días	lun 20/03/17	vie 09/06/17	4	
4 1ª Iteración	30 días	lun 20/03/17	vie 28/04/17	15	
Diagrama de clases	4 días	lun 20/03/17	jue 23/03/17		Santos Ángel Pardo
Diagrama de secuencia	4 días	vie 24/03/17	mié 29/03/17	24	Santos Ángel Pardo
Diagrama de arquitectura	2 días	jue 30/03/17	vie 31/03/17	25	Santos Ángel Pardo
Programación	18 días	lun 03/04/17	mié 26/04/17	26	Santos Ángel Pardo
Planificación fase de construcción 2ª iteración	2 días	jue 27/04/17	vie 28/04/17	27	Santos Ángel Pardo
4 2ª Iteración	30 días	lun 01/05/17	vie 09/06/17	23	
Programación de código	20 días	lun 01/05/17	vie 26/05/17		Santos Ángel Pardo
Manual de usuario	2 días	lun 29/05/17	mar 30/05/17	30	Santos Ángel Pardo
Revisión diagramas de clase	2 días	mié 31/05/17	jue 01/06/17	31	Santos Ángel Pardo
Revisión diagramas de secuencia	2 días	vie 02/06/17	lun 05/06/17	32	Santos Ángel Pardo
Revisión diagramas de arquitectura	2 días	mar 06/06/17	mié 07/06/17	33	Santos Ángel Pardo
Planificación fase de transición 1ª iteración	2 días	jue 08/06/17	vie 09/06/17	34	Santos Ángel Pardo

Figura 3.5: Fase de construcción

4 Fase de transición	10 días	lun 12/06/17	vie 23/06/17	22	
4 1ª Iteración	10 días	lun 12/06/17	vie 23/06/17	29	
Depuración y pruebas	4 días	lun 12/06/17	jue 15/06/17		Santos Ángel Pardo
Corrección de errores	3 días	vie 16/06/17	mar 20/06/17	38	Santos Ángel Pardo
Revisión y documentación	3 días	mié 21/06/17	vie 23/06/17	39	Santos Ángel Pardo

Figura 3.6: Fase de transición

3.7. Costes

Para la realización del cálculo de los costes se han tenido en cuenta diferentes aspectos que conllevan diferentes costes:

- **Coste de material y equipos:** Dada la imposibilidad de utilizar un ordenador de Renault (por motivos de seguridad era imposible instalar software para realizar el desarrollo) hubo que aportar un ordenador privado. El software utilizado estaba preinstalado de fábrica o fue adquirido con licencia de pago (incluido en la matrícula de la universidad) por lo que estos costes no serán incluidos en el presupuesto. Como el sistema será utilizado finalmente en ordenadores de Renault ya adquiridos previamente tampoco serán incluidos en el presupuesto.

- MSI GP60 2QE LEOPARRD: 799 €.

COSTE TOTAL DEL MATERIAL = 799 €

- **Coste de personal:** El proyecto ha sido desarrollado dentro de la realización de las prácticas de empresa dentro de Renault. Estas prácticas tienen una duración de 5 meses con una ayuda económica de 2700 €. Haciendo un cálculo mensual obtenemos 540 euros mensuales.

La realización de este proyecto fuera de estas prácticas, solicitando los servicios de una empresa profesional, hubiera supuesto un coste más elevado.

- $540 \text{ €} * 5 * 1 \text{ persona} = 2700 \text{ €}$.

COSTE TOTAL DE PERSONAL = 2700 €

No se ha estimado ningún gasto adicional más, por lo que haciendo cuentas el desarrollo total de este proyecto supondrá un **coste de 3499 €**.

3.8. Análisis de Riesgos

Durante la realización de este proyecto existen posibles riesgos que amenazan el éxito final de la aplicación o que provocan que no se cumplan las pautas establecidas antes del comienzo de su desarrollo. El objetivo de todo proyecto debe ser obtener un producto con la mayor calidad y minimizar el impacto de los riesgos que se pueden presentar.

La gestión de riesgos se encarga de realizar un análisis de riesgos para evaluar los recursos y establecer políticas de gestión contra amenazas presentes o futuras, ayudando a garantizar que ocurre lo que se había planificado inicialmente. A continuación, se presentan una serie de riesgos, los cuales son descritos y detallados, también se presentan una serie de acciones para su corrección. Los riesgos pueden ser de tres tipos:

- **Riesgos del proyecto:** amenazan al plan del proyecto. Si aparecen puede que la planificación temporal y los costes aumenten.
- **Riesgos técnicos:** amenazan la calidad y la planificación. Identifican problemas potenciales de diseño, implementación, interfaz, etc.
- **Riesgos de negocio:** amenazan la viabilidad del sistema a construir.

3.8.1. Riesgo de Proyecto

Riesgo-001 Incumplimiento de plazos (retrasos)	
Descripción	Los plazos establecidos para las diferentes fases del proyecto no son alcanzables y se sufren retrasos en el desarrollo.
Efecto	Retraso en las fases del proyecto.
Frecuencia	Alta
Gravedad	Alta
Detección	Alta
Acción correctora	La planificación inicial debe realizarse de una manera más realista.
Plan de contingencia	Aumentar las horas de trabajo diarias o realizar horas extra.

Tabla 3.4: Incumplimiento de plazos(retrasos)

Riesgo-002		Incumplimiento de plazos (adelantos)	
Descripción	Los plazos establecidos para las diferentes fases del proyecto no son alcanzables y se sufren adelantos en el desarrollo.		
Efecto	Adelanto en las fases del proyecto.		
Frecuencia	Baja		
Gravedad	Alta		
Detección	Alta		
Acción correctora	La planificación inicial debe realizarse de una manera más realista.		
Plan de contingencia	Trabajar con menos estrés y más relajación o realizar un código más limpio y eficiente, realizando una depuración mayor.		

Tabla 3.5: Incumplimiento de plazos (adelantos)

Riesgo-003		Cambios en los requisitos	
Descripción	Los requisitos establecidos en las fase iniciales del proyecto son modificados, eliminados o añadidos otros nuevos.		
Efecto	Cambio en las fases de desarrollo.		
Frecuencia	Alta		
Gravedad	Alta		
Detección	Alta		
Acción correctora	Añadir un margen de tiempo extra al final de cada fase para la realización de estas modificaciones.		
Plan de contingencia	Aumentar las horas de trabajo diarias o realizar horas extra.		

Tabla 3.6: Modificación de los requisitos

Riesgo-004		Perdida de contenido	
Descripción	Perdida de partes del proyecto como código fuente, imágenes, documentos, etc.		
Efecto	Retraso en las fases.		
Frecuencia	Baja		
Gravedad	Media		
Detección	Media		
Acción correctora	Realizar copias de seguridad con un frecuencia suficiente como para que las perdidas no sean significativas y no supongan un grave retraso.		
Plan de contingencia	Aumentar las horas de trabajo diarias o realizar horas extra para recuperar la información.		

Tabla 3.7: Perdida de contenido

Riesgo-005		Avería del ordenador en el que se desarrolla	
Descripción	El ordenador en el que se desarrolla el proyecto sufre una avería.		
Efecto	Retraso en las fases.		
Frecuencia	Baja		
Gravedad	Media		
Detección	Media		
Acción correctora	Realizar copias de seguridad en otros dispositivos externos y disponer de un segundo ordenador.		
Plan de contingencia	Aumentar las horas de trabajo diarias o realizar horas extra para recuperar proyecto.		

Tabla 3.8: Avería

Riesgo-006		Enfermedad o ausencia en el puesto de trabajo	
Descripción	Debido a una baja por enfermedad o por motivos personales no se puede cumplir el calendario establecido.		
Efecto	Retraso en las fases.		
Frecuencia	Baja		
Gravedad	Media		
Detección	Media		
Acción correctora	A la hora de planificar el calendario de fases, contar con este imprevisto a la hora de su realización.		
Plan de contingencia	Aumentar las horas de trabajo diarias o realizar horas extra para recuperar proyecto.		

Tabla 3.9: Ausencia en el puesto de trabajo

Riesgo-007		Falta de experiencia	
Descripción	Debido a una falta de conocimientos, ya sea sobre el proyecto, sobre la programación o sobre el entorno, se produce alteraciones en la planificación del proyecto.		
Efecto	Retraso en las fases e incremento del coste.		
Frecuencia	Baja		
Gravedad	Media		
Detección	Media		
Acción correctora	Formarse y documentarse previamente en el entorno, en el proyecto y en la tecnología que se va a emplear.		
Plan de contingencia	Revisión de los conocimientos antes de realizar el proyecto o en las fases iniciales.		

Tabla 3.10: Falta de experiencia.

3.8.2. Riesgo Técnicos

Riesgo-008		Calidad inadecuada	
Descripción	El software presentado cumple con los requisitos pedidos por el usuario pero sufre numerosos fallos o un gastos de tiempo y recursos excesivo.		
Efecto	Perdida de confianza por parte del cliente para futuros proyectos.		
Frecuencia	Baja		
Gravedad	Alta		
Detección	Alta		
Acción correctora	Realizar pruebas en las que se ponga a prueba el software y corregir los fallos.		
Plan de contingencia	Planificar las pruebas para su realización durante la realización del producto.		

Tabla 3.11: Calidad inadecuada

3.8.3. Riesgo de Negocio

Riesgo-009		Falta de seguimiento	
Descripción	El proyecto no ha recibido un seguimiento suficiente por parte del responsable del proyecto y presenta defectos en una o varias partes.		
Efecto	El sistema no habrá sido realizado correctamente y puede presentar muy diversos errores.		
Frecuencia	Baja		
Gravedad	Muy alta		
Detección	Alta		
Acción correctora	Mayor implicación por parte por parte del jefe del proyecto.		
Plan de contingencia	Corregir los defectos, lo que puede provocar un incremento en coste y tiempo del proyecto.		

Tabla 3.12: Falta de seguimiento

En las siguientes secciones se procede a desmenuzar y analizar la funcionalidad que deben recoger los sistemas. Con la información aportada en la sección 3.2 se procederá a analizar cuales son los requisitos que se deben alcanzar, las personas que interactuaran con la aplicación así como un análisis en profundidad de los casos de uso que compongan la funcionalidad esperada por parte de los clientes.

4.1. Método de Recogidas de Datos

En el apartado 3.2 se presentó una descripción detallada del sistema. Fue presentada en ese punto para poder tener un conocimiento previo antes de presentar la planificación. Dicha información fue obtenida de la manera que se presenta a continuación.

Para la recogida de datos se pueden emplear múltiples métodos. La recogida de datos consiste en obtener información para poder tener conocimiento de los objetivos y diseñar los casos de uso. Esta observación también incluye el análisis del entorno para saber que herramientas se adaptan mejor a la solución que se puede ofrecer. La recogida de información puede ser de muy variadas formas, desde entrevistas personales y encuestas, a observación o exploración.

Las entrevistas con los coordinadores fueron la principal manera de obtener información. Los coordinadores eran los clientes, ellos habían solicitado el desarrollo de una serie de proyectos por lo que ellos eran los que iban a decidir como debe de ser el sistema. Las entrevistas tuvieron lugar en el propio departamento, en una sala de reuniones, al principio del proyecto. No solo se emplearon reuniones planeadas, el hecho de estar en la misma sala con ellos daba lugar a conversaciones que propiciaban pequeños matices al proyecto. Debido a que ellos ya usaban antes un sistema para la realización de pedidos, se pudo obtener ciertos detalles de esas fuentes de datos secundarias.

Otro aspecto a considerar es que este proyecto fue entregado cuando mi estancia en Renault no había finalizado. Debido al trato frecuente con los propios JU's, ellos me comunicaban ciertas mejoras en el software, principalmente en la interfaz de usuario. Estas mejoras trataban de pequeñas implementaciones que permitían ahorrar unos segundos en la realización del pedido.

4.2. Participantes en el Proyecto

A continuación, se presentan los participantes relacionados con la realización de este proyecto.

Participante n°1	
Nombre y apellidos	Santos Ángel Pardo Ramos
Organización	Estudiante en la E.T.S.I.I
Rol	Jefe de proyecto, analista, diseñador y desarrollador
¿Es desarrollador?	Sí
¿Es cliente?	No
¿Es usuario?	No

Tabla 4.1: Participante n°1

Participante n°2	
Nombre y apellidos	Jesús María Vegas Hernández
Organización	Profesor de la E.T.S.I.I
Rol	Tutor de TFG
¿Es desarrollador?	No
¿Es cliente?	No
¿Es usuario?	No

Tabla 4.2: Participante n°2

Participante n°3	
Nombre y apellidos	Cristina Cuende Puertas
Organización	Trabajadora de Renault Palencia
Rol	Cliente y usuario final
¿Es desarrollador?	No
¿Es cliente?	Sí
¿Es usuario?	Sí

Tabla 4.3: Participante n°3

Participante n°4	
Nombre y apellidos	Ángel Mayordomo
Organización	Trabajador de Renault Palencia
Rol	Cliente y usuario final
¿Es desarrollador?	No
¿Es cliente?	Sí
¿Es usuario?	Sí

Tabla 4.4: Participante n°4

4.3. Objetivos del Sistema

Las siguientes tablas que se muestran a continuación son las conclusiones del apartado 6.1, en el cual se describe cual es el proceso, actualmente realizado manualmente, a informatizar. En las próximas tablas se muestran, con una visión global, cuales son los objetivos finales del sistema.

Objetivo n°1	
Título	Hacer pedido
Autor	Santos Ángel Pardo Ramos
Fuente	Cristina Cuende Puertas y Ángel Mayordomo
Descripción	El sistema deberá permitir a una serie de usuarios específicos la realización de pedidos y su posterior envío.

Tabla 4.5: Objetivo n°1

Objetivo n°2	
Título	Generar ficheros con los artículos solicitados
Autor	Santos Angel Pardo Ramos
Fuente	Cristina Cuende Puertas y Ángel Mayordomo
Descripción	El sistema deberá rellenar dos plantillas, guardadas como ficheros Excel, con los artículos y demás información complementaria sobre el pedido que acaba de realizar el usuario.

Tabla 4.6: Objetivo n°2

Objetivo n°3	
Título	Enviar pedido al servidor.
Autor	Santos Ángel Pardo Ramos
Fuente	Cristina Cuende Puertas y Ángel Mayordomo
Descripción	El sistema deberá permitir a todo usuario registrado en el servidor el envío de los artículos solicitados por el JU.

Tabla 4.7: Objetivo n°3

4.4. Requisitos

En el siguiente apartado, se presentan cuáles son las funcionalidades que el sistema debe de tener. Estas funcionalidades han sido extraídas a partir de las necesidades planteadas por los clientes, en este caso, los coordinadores. Las tablas que se presentan a continuación son el resultado de las entrevistas con los coordinadores. El final de este proceso, concluye con la presentación de los casos de uso.

La presentación de los requisitos se hará en forma de tablas. Las tablas siguen un patrón que incluye los siguientes campos:

- Nombre corto
- Descripción
- Estado (propuesto, aprobado, incluido o validado)
- Coste estimado de implementación
- Prioridad (critico, importante o secundario)
- Nivel de riesgo asociado a la implementación de la característica (crítico, significativo, ordinario)

4.4.1. Requisitos Funcionales

Requisito funcional n°1	
Nombre	Iniciar sesión
Descripción	El sistema deberá permitir iniciar sesión a aquellos usuarios que tengan permiso para hacer pedidos. Si no se ha iniciado sesión no se puede realizar ningún pedido.
Estado	Incluido
Coste estimado	1 día
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.8: RF n°1

Requisito funcional n°2	
Nombre	Acceso a los ajustes del sistema
Descripción	El sistema deberá permitir el acceso a los ajustes del sistema a aquellos usuarios que tengan permiso para ajustarlo.
Estado	Incluido
Coste estimado	1 día
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.9: RF n°2

Requisito funcional n°3	
Nombre	Ajustar el sistema
Descripción	El sistema deberá permitir realizar ajustes sobre su funcionamiento, permitiendo así ajustarse a diferentes cambios en la toma de datos o cambios en el sistema en que se aloja.
Estado	Incluido
Coste estimado	5 días
Prioridad	Crítico
Nivel de riesgo	Crítico

Tabla 4.10: RF n°3

Requisito funcional n°4	
Nombre	Hacer pedido respecto a otro anterior
Descripción	El sistema deberá permitir hacer un pedido, tomando como fuente de información pedidos anteriores.
Estado	Incluido
Coste estimado	3 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.11: RF n°4

Requisito funcional n°5	
Nombre	Hacer pedido
Descripción	El sistema deberá permitir buscar referencias y cantidades. Si el usuario está conforme permite añadirlas al pedido.
Estado	Incluido
Coste estimado	5 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.12: RF n°5

Requisito funcional n°6	
Nombre	Ver referencias disponibles
Descripción	El sistema deberá permitir ver la información sobre todos los artículos que están disponibles, facilitando así la realización del pedido.
Estado	Incluido
Coste estimado	3 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.13: RF n°6

Requisito funcional n°7	
Nombre	Modificar pedido
Descripción	El sistema deberá permitir hacer modificaciones o eliminaciones de los artículos añadidos al pedido antes de ser enviado.
Estado	Incluido
Coste estimado	3 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.14: RF n°7

Requisito funcional n°8	
Nombre	Control económico
Descripción	El sistema deberá llevar un control sobre el coste del pedido y el dinero que posee. Deberá mostrar cual es el estado de sus gastos y validar el pedido en función de este.
Estado	Incluido
Coste estimado	2 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.15: RF n°8

Requisito funcional nº9	
Nombre	Generar salida
Descripción	El sistema deberá cumplimentar fichero con los datos de los artículos que ha solicitado, que será entregada al coordinador en una carpeta específica del sistema. También otro para el JU.
Estado	Incluido
Coste estimado	2 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.16: RF nº9

Requisito funcional nº10	
Nombre	Seleccionar salida
Descripción	El sistema deberá permitir seleccionar una carpeta al usuario donde guardar su resguardo del pedido.
Estado	Incluido
Coste estimado	2 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.17: RF nº10

Requisito funcional nº11	
Nombre	Obtener datos del usuario
Descripción	El sistema deberá solicitar al usuario los datos necesarios para poder autenticarse con el servidor. Para ello solicita el usuario y contraseña.
Estado	Incluido
Coste estimado	1 día
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.18: RF nº11

Requisito funcional nº12	
Nombre	Leer fichero Excel
Descripción	El sistema deberá ser capaz de leer un fichero Excel que contiene los artículos y sus cantidades. Una vez leído serán mostrados por pantalla para que el usuario sepa cuales va a pedir.
Estado	Incluido
Coste estimado	1 día
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.19: RF nº12

Requisito funcional nº13	
Nombre	Enviar artículos
Descripción	El sistema debe ser capaz de enviar al servidor artículo por artículo, de igual manera que si se hiciera manualmente. Al finalizar, el sistema mostrará por pantalla la lista de los artículos que se han pedido, en caso de error, se mostrará el error.
Estado	Incluido
Coste estimado	20 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.20: RF nº13

4.4.2. Requisitos No Funcionales

Requisito no funcional nº1	
Nombre	Seguridad
Descripción	El sistema deberá implementar un nivel de seguridad que no comprometa la integridad de la información almacenada o la información generada.
Estado	Incluido
Coste estimado	5 días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.21: RNF nº1

Requisito no funcional nº2	
Nombre	Facilidad de uso
Descripción	El sistema deberá ser fácil de usar y de una manera intuitiva.
Estado	Incluido
Coste estimado	- días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.22: RNF nº2

Requisito no funcional nº3	
Nombre	Funcionamiento
Descripción	El sistema deberá funcionar en diferentes tipos de ordenadores así como en diferentes tipos de pantalla.
Estado	Incluido
Coste estimado	- días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.23: RNF nº3

Requisito no funcional nº4	
Nombre	Escalabilidad
Descripción	El sistema debe ser escalable, es decir, el sistema debe estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.
Estado	Incluido
Coste estimado	- días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.24: RNF nº4

Requisito no funcional nº5	
Nombre	Extensibilidad
Descripción	El sistema debe ser capaz de añadir nuevos servicios de manera facil, sin que esto produzca un cambio en toda la estructura interna del programa.
Estado	Incluido
Coste estimado	- días
Prioridad	Importante
Nivel de riesgo	Ordinario

Tabla 4.25: RNF nº5

4.5. Actores

El sistema contempla una serie de actores, que son aquellas personas que interactúan con la aplicación. En nuestro sistema encontramos dos, un actor que hace referencia a los JU's en la realización del pedido y otro actor que es el coordinador, el cual recibe el pedido ya hecho y administra los ajustes de la aplicación.

Actor nº1	
Nombre	JU
Descripción	Actor que interactúa con el sistema mediante la realización de pedidos.

Tabla 4.26: Actor nº1

Actor nº2	
Nombre	Coordinador
Descripción	Actor que interactúa con el sistema mediante los ajustes de este.

Tabla 4.27: Actor nº2

Actor nº3	
Nombre	Responsable
Descripción	Actor que interactúa con el sistema y realiza los diferentes envíos de pedidos.

Tabla 4.28: Actor nº3

4.6. Casos de Uso

Una vez hemos obtenido información sobre nuestro proyecto, sabemos cuáles son los requisitos funcionales y no funcionales, sabemos cuáles son los actores que van a interactuar con nuestro sistema; procedemos a realizar el diagrama de casos de uso.

El diagrama de casos de uso, figura 4.1, trata de explicar, desde la perspectiva del actor, que puede hacer con el sistema y de qué funcionalidades dispone.

Una vez presentado el diagrama de casos de uso se procede a su explicación, la explicación se lleva a cabo por cada caso de uso específico. Se procede a desmenuzar cada caso de uso en las iteraciones que el usuario debe de realizar así como de las excepciones que puedan surgir. También se indica si existe alguna condición previa antes de poder ejecutarlo.

Caso de uso nº1		
Nombre	Iniciar sesión	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá permitir iniciar sesión a los usuarios que estén registrados en el sistema.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1º	El JU (Actor nº1) inicia la ejecución del programa en el icono correspondiente.
	2º	El sistema muestra por pantalla una pantalla principal para iniciar sesión.
	3º	El JU (Actor nº1) pulsa el botón iniciar sesión.
	4º	El sistema muestra por pantalla un formulario con un campo usuario y otro contraseña para iniciar sesión. Junto a ello un botón para confirmar.
	5º	El JU (Actor nº1) rellena los campos usuario y contraseña y pulsa sobre el botón de confirmación.
	6º	El sistema comprueba los datos y se muestra la pantalla siguiente.
Excepciones	Paso	Acción
	4º	El sistema no ha podido localizar las fuentes de información. En ese caso, añade un mensaje de error por la pantalla. El sistema no puede funcionar.
	6º	El usuario o la contraseña son erróneos, el sistema muestra por pantalla el error. Permite volver a introducirlos.

Tabla 4.29: Caso de uno nº1

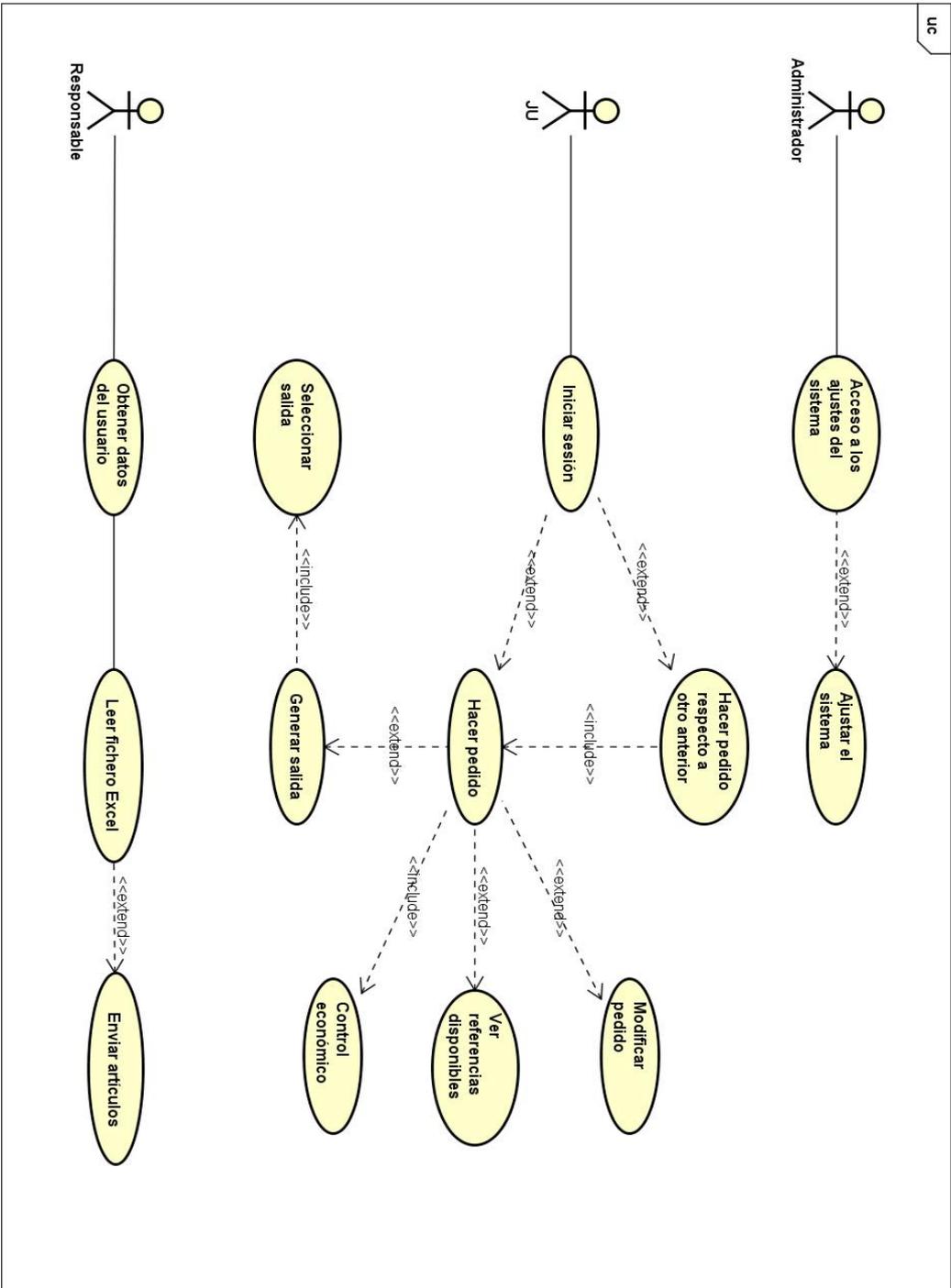


Figura 4.1: Diagrama de casos de uso

Caso de uso n°2		
Nombre	Hacer pedido	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá permitir la realización manual de pedidos.	
Precondición	Haber iniciado sesión (Caso de uso n° 1)	
Secuencia normal	Paso	Acción
	1°	El sistema muestra por pantalla (nada más acabar el caso de uso n°1) dos botones, uno de ellos para poder hacer un pedido.
	2°	El JU (Actor n°1) hace clic sobre el botón “hacer pedido”.
	3°	El sistema muestra por pantalla una serie de campos, botones y tablas en las que se podrán ir realizando los casos de uso siguientes.
	4°	El usuario pulsa el botón enviar
	5°	El sistema valida los datos introducidos. Si todo es correcto da por válido el pedido. El sistema muestra por pantalla una segunda confirmación para la realización del pedido.
	6°	El usuario confirma el pedido
Excepciones	Paso	Acción
	3°	El sistema comprueba que el usuario ya ha realizado un pedido ese mes. El sistema muestra parcialmente la pantalla siguiente impidiendo la realización de algunos casos de uso. Además, muestra un mensaje comunicándoselo al usuario.
	5°	El sistema no da por válido el pedido. El muestra de nuevo los datos introducidos para que el usuario realice los cambios apropiados.

Tabla 4.30: Caso de uso n°2

Caso de uso nº3		
Nombre	Hacer pedido anterior	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá permitir la carga de pedidos de meses anteriores.	
Precondición	Haber iniciado sesión (Caso de uso nº 1)	
Secuencia normal	Paso	Acción
	1º	El sistema muestra por pantalla (nada más acabar el caso de uso nº1) dos botones, uno de ellos para poder hacer un pedido igual que otro anterior.
	2º	El JU (Actor nº1) hace clic sobre el botón “hacer pedido anterior”.
	3º	El sistema muestra por pantalla un selector de ficheros.
	4º	El usuario selecciona el resguardo respectivo del mes que quiere hacer el pedido.
	5º	El sistema lee el fichero y añade lee artículos contenidos en ese fichero. Cada artículo ha sido validado para comprobar si sigue estando disponible y en las mismas cantidades. Una vez finalizado la carga, el sistema realiza el caso de uso nº2 pero mostrando por pantalla estos artículos.
Excepciones	Paso	Acción
	3º	El sistema comprueba que el usuario ya ha realizado un pedido ese mes. El sistema muestra parcialmente la pantalla del caso de uno nº2 impidiendo la realización de algunos casos de uso Además, muestra un mensaje comunicándoselo al usuario.
	5º	El sistema no da por válido el fichero. Muestra un mensaje de error al usuario.

Tabla 4.31: Caso de uso nº3

Caso de uso nº4		
Nombre	Ver catálogo	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema permite ver una lista completa de todos los artículos disponibles, acompañados de su referencia, descripción, cantidad mínima y precio.	
Precondición	Hacer un pedido (Caso de uso nº2)	
Secuencia normal	Paso	Acción
	1º	El JU (Actor nº1) hace clic sobre el botón “ver catálogo”.
	2º	El sistema muestra por pantalla una lista con todos los artículos disponibles. En la tabla se mostrarán los campos referencias, descripción, cantidad y precio.
	3º	El usuario navega y busca el artículo que desee.
Excepciones	Paso	Acción

Tabla 4.32: Caso de uso nº4

Caso de uso nº5		
Nombre	Buscar artículo	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema permite realizar una búsqueda de un artículo, introduciendo también una cantidad. Así podemos ver a información y coste que tiene.	
Precondición	Hacer un pedido (Caso de uso nº2)	
Secuencia normal	Paso	Acción
	1º	El usuario introduce en el campo artículo y en el campo cantidad los datos correspondientes. Después pulsa sobre el botón buscar.
	2º	El sistema muestra por pantalla la información correspondiente, así como su coste.
Excepciones	Paso	Acción
	2º	El artículo no existe. El sistema muestra un mensaje de error diciendo que el artículo no existe.
	2º	La cantidad introducida no es válida. El sistema muestra un mensaje de error explicando por qué no es válida esa cantidad.

Tabla 4.33: Caso de uso nº5

Caso de uso nº6		
Nombre	Añadir artículo	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema permite añadir un artículo al pedido.	
Precondición	Buscar un artículo (Caso de uso nº 5)	
Secuencia normal	Paso	Acción
	1º	El usuario introduce en el campo artículo y en el campo cantidad los datos correspondientes. Después pulsa sobre el botón buscar.
	2º	El sistema muestra por pantalla la información, correspondiente, así como su coste.
Excepciones	Paso	Acción
	2º	El artículo no existe. El sistema muestra un mensaje de error diciendo que el artículo no existe.
	2º	La cantidad introducida no es válida. El sistema muestra un mensaje de error explicando por qué no es válida esa cantidad.

Tabla 4.34: Caso de uso nº6

Caso de uso nº7		
Nombre	Eliminar artículo	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema permite eliminar un artículo del pedido.	
Precondición	Añadir un artículo (Caso de uso nº6)	
Secuencia normal	Paso	Acción
	1º	El usuario introduce en el campo artículo correspondiente a la sección eliminar la referencia del artículo que va a borrar
	2º	El sistema comprueba que el artículo se encuentra en el pedido y lo borra.
Excepciones	Paso	Acción
	2º	El artículo no se encuentra en el pedido. El sistema muestra un mensaje de error indicando que no se ha pedido.

Tabla 4.35: Caso de uso nº7

Caso de uso nº8		
Nombre	Modificar artículo	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema permite modificar un artículo del pedido.	
Precondición	Añadir un artículo (Caso de uso nº6)	
Secuencia normal	Paso	Acción
	1º	El usuario introduce en el campo artículo y en el campo correspondiente la sección modificar, la referencia del artículo y la nueva cantidad deseada.
	2º	El sistema comprueba que el artículo ha sido pedido y que la cantidades compatible. Si todo es correcto, el sistema realiza el cambio.
Excepciones	Paso	Acción
	2º	El artículo no existe. El sistema muestra un mensaje de error diciendo que el artículo no existe.
	2º	La cantidad introducida no es válida. El sistema muestra un mensaje de error explicando por qué no es válida esa cantidad.

Tabla 4.36: Caso de uso nº8

Caso de uso n°9		
Nombre	Configuración del sistema	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá permitir la configuración de este para que pueda ajustar a las necesidades del usuario.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1º	El coordinador (Actor n°2) inicia la ejecución del programa en el icono correspondiente.
	2º	El sistema muestra por pantalla una pantalla principal para iniciar sesión.
	3º	El coordinador (Actor n°2) pulsa el botón iniciar sesión.
	4º	El sistema muestra por pantalla un formulario con un campo usuario y otro contraseña para iniciar sesión. Junto a ello un botón para confirmar. También muestra un enlace con el texto “¿Eres administrador?”.
	5º	El coordinador (Actor n°2) hace clic sobre el enlace.
	6º	El sistema muestra una pantalla en la que solicita una contraseña para acceder a la configuración del sistema El sistema muestra un campo y un botón.
	7º	El coordinador (Actor n°2) introduce la contraseña correspondiente y hace clic sobre el botón “Entrar”.
	8º	El sistema comprueba la contraseña y muestra una nueva ventana. Esta nueva ventana contiene los campos correspondientes a las configuraciones del sistema. Se muestra un botón.
	9º	El coordinador (Actor n°2) realiza los cambios oportunos y pulsa sobre el botón guardar.
	10º	El sistema guarda la configuración elegida.
	11º	El coordinador (Actor n°2) hace clic sobre el botón finalizar.
	12º	El programa finaliza.
Excepciones	Paso	Acción
	4º	El sistema no ha podido localizar las fuentes de información En ese caso, añade un mensaje de error por la pantalla.
	8º	La contraseña introducida es errónea, el sistema muestra por pantalla el error. Permite volver a introducirlo.

Tabla 4.37: Caso de uso n°9

Caso de uso n°10		
Nombre	Obtener datos del usuario	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá obtener los datos del responsable (usuario y contraseña) para permitir hacer el pedido en el servidor.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1º	El responsable (Actor n°3) inicia la ejecución del programa en el icono correspondiente.
	2º	El sistema muestra por pantalla una pantalla principal para iniciar sesión.
	3º	El responsable (Actor n°3) pulsa el botón iniciar sesión.
	4º	El sistema muestra por pantalla un formulario con un campo usuario y otro contraseña para iniciar sesión. Junto a ello un botón para confirmar.
	5º	El responsable (Actor n°3) rellena los campos usuario y contraseña y pulsa sobre el botón de confirmación.
	6º	El sistema cambia a la pantalla siguiente.
Excepciones	Paso	Acción
	4º	El sistema no ha podido localizar las fuentes de información. En ese caso, añade un mensaje de error por la pantalla. El sistema, no puede funcionar.
	5º	El supervisor no ha insertado datos en los campos, el sistema muestra un error para que se corrija y no permite el cambiar de ventana.

Tabla 4.38: Caso de uno n°10

Caso de uso n°11		
Nombre	Leer fichero	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá permitir leer un fichero, con un formato específico. Este fichero contiene los artículos que se desean pedir, sus cantidades y sus centros de gastos.	
Precondición	Haber dado los datos iniciales (Caso de uso n° 10)	
Secuencia normal	Paso	Acción
	1º	El sistema muestra por pantalla (nada más acabar el caso de uso n°10) una pantalla en la que se muestra un botón para leer el fichero, otro botón para enviar, una pantalla en la que se mostrarán los artículos que se van a pedir y una última en la que se mostrará el proceso del envío.
	2º	El responsable (Actor n°3) hace clic sobre el botón "seleccionar".
	3º	El sistema muestra por pantalla un selector de ficheros.
	4º	El responsable selecciona el fichero que desee.
	5º	El sistema lee el fichero y añade los artículos contenidos en él mostrándolos por pantalla.
Excepciones	Paso	Acción
	5º	El sistema no da por válido el fichero. Muestra un mensaje de error al usuario.

Tabla 4.39: Caso de uso n°11

Caso de uso nº12		
Nombre	Enviar artículos	
Autor	Santos Ángel Pardo Ramos	
Descripción	El sistema deberá permitir el envío de los pedidos que el responsable desee.	
Precondición	Haber cargado los datos (Caso de uso nº 10) y leído el fichero Excel (Caso de uso nº 11).	
Secuencia normal	Paso	Acción
	1º	El sistema muestra por pantalla, nada más acabar el caso de uso nº10, la lista de artículos.
	2º	El responsable (Actor nº3) si está conforme con los artículos pulsa el botón enviar.
	3º	El sistema comienza el envío, mostrando por pantalla el proceso del envío.
	4º	Una vez terminado, el sistema muestra por pantalla la lista total de artículos y el número de referencia del pedido.
Excepciones	Paso	Acción
	4º	Ocurre un error en alguno de los artículos. El sistema muestra por pantalla el error cancelando el envío.

Tabla 4.40: Caso de uso nº12

4.7. Diagramas de Actividad

4.7.1. Iniciar Sesión

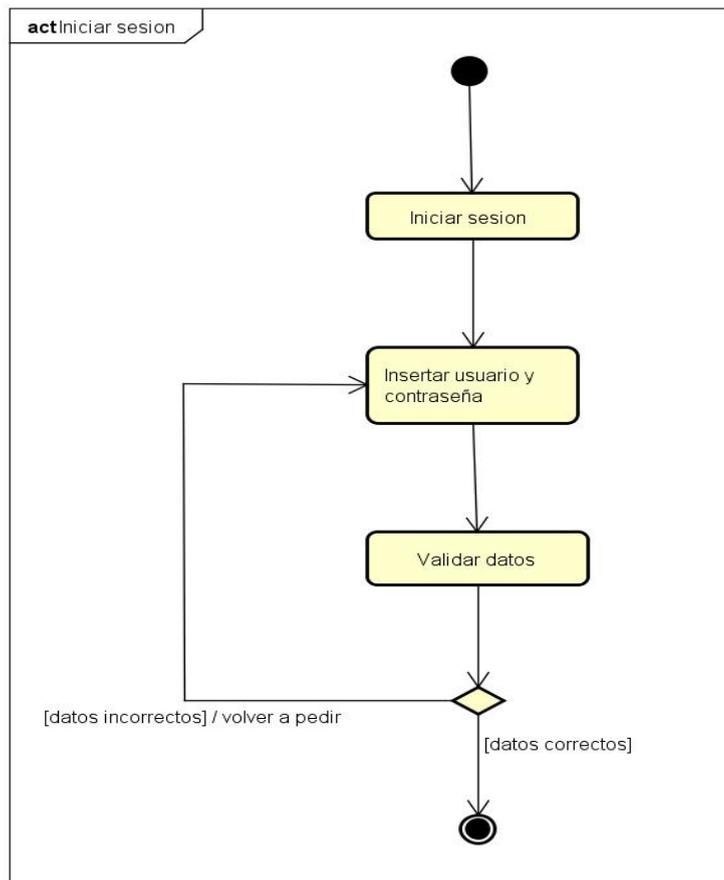


Figura 4.2: Diagrama de actividad principal

4.7.2. Configurar

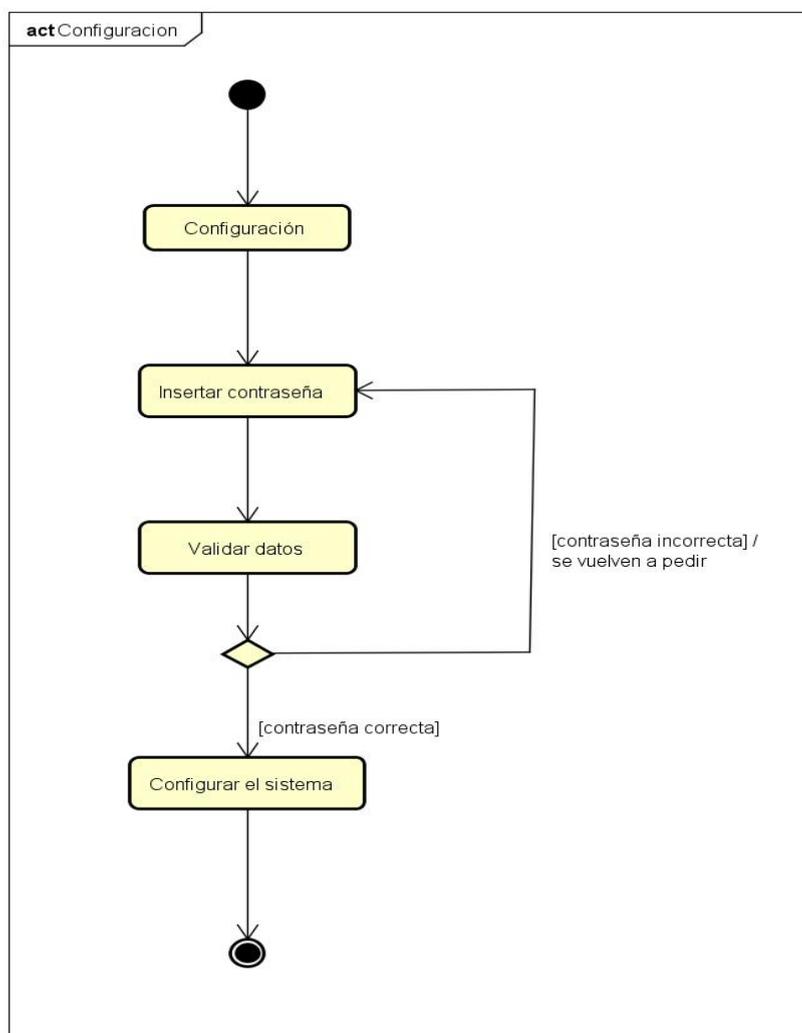


Figura 4.3: Diagrama de actividad principal

4.7.3. Hacer Pedido Manual

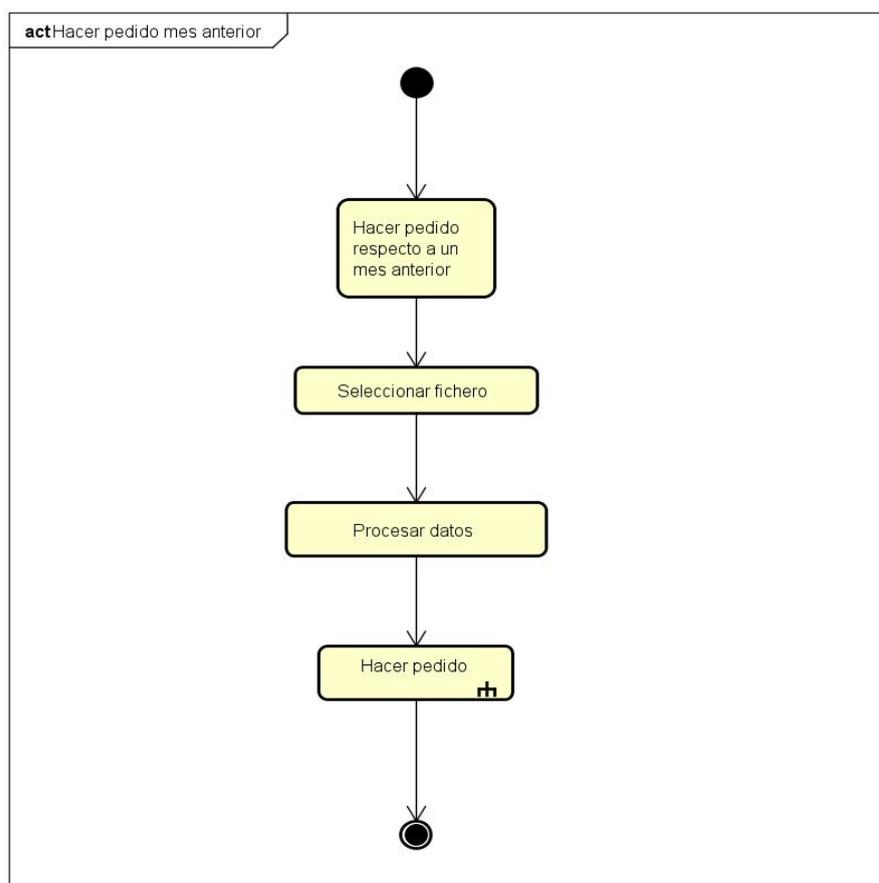


Figura 4.4: Diagrama de actividad principal

4.7.4. Hacer Pedido

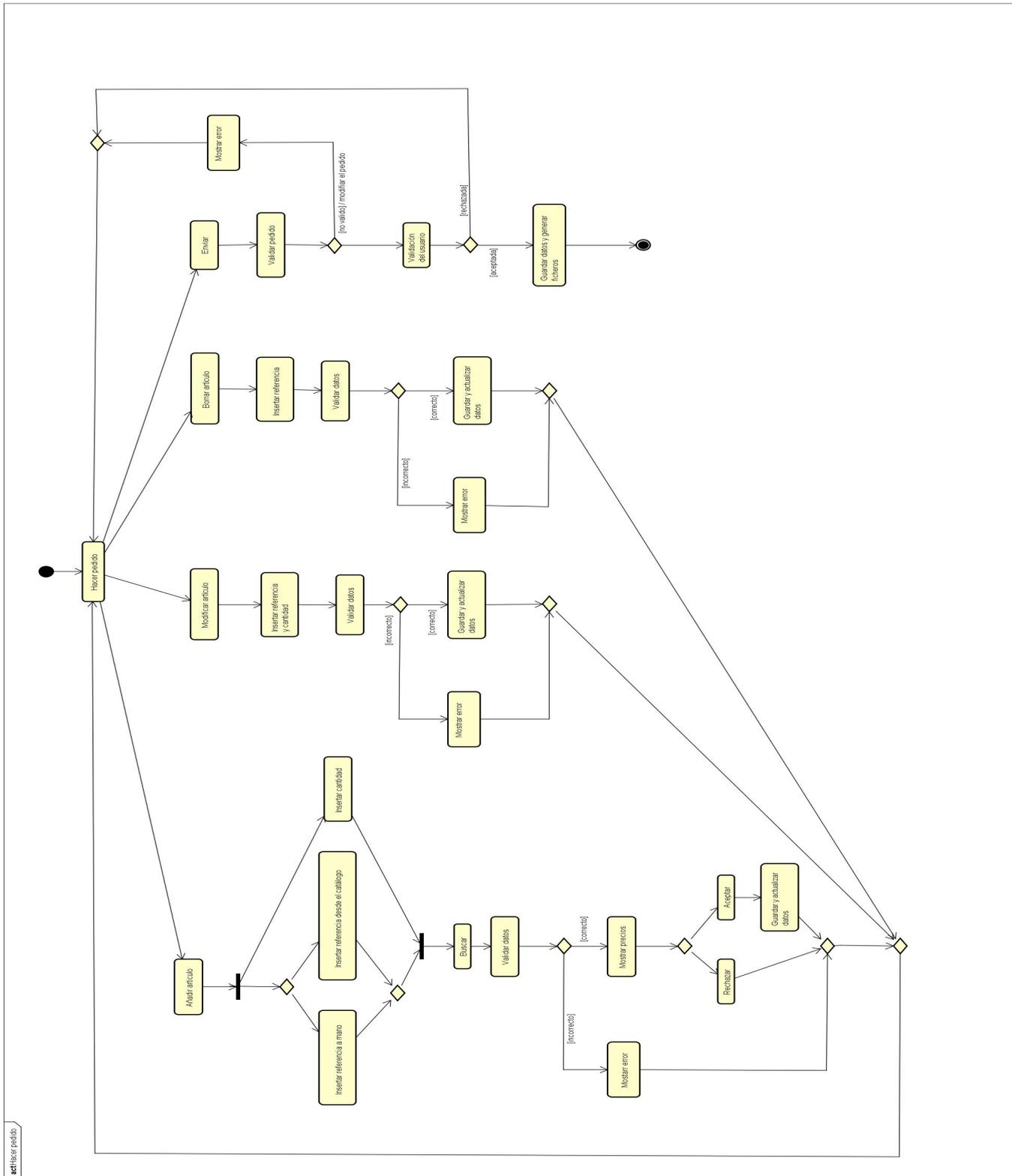


Figura 4.5: Diagrama de actividad para hacer un pedido

4.7.5. Enviar Pedido

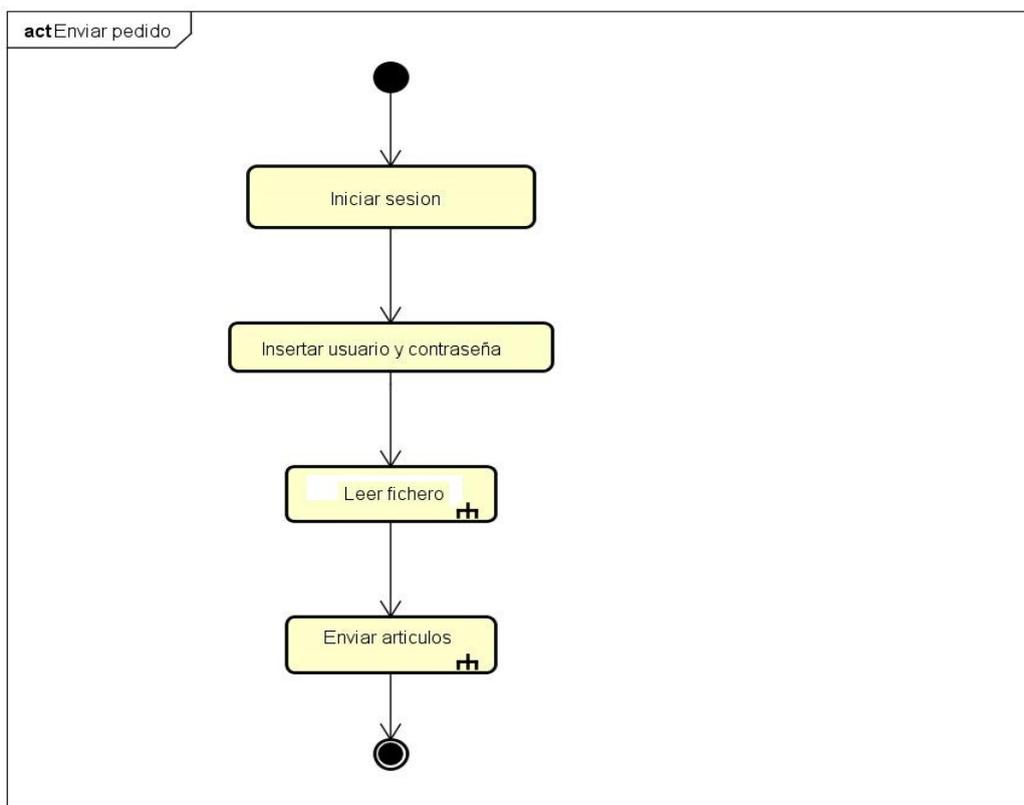


Figura 4.6: Diagrama de actividad para enviar un pedido

4.7.6. Leer Fichero

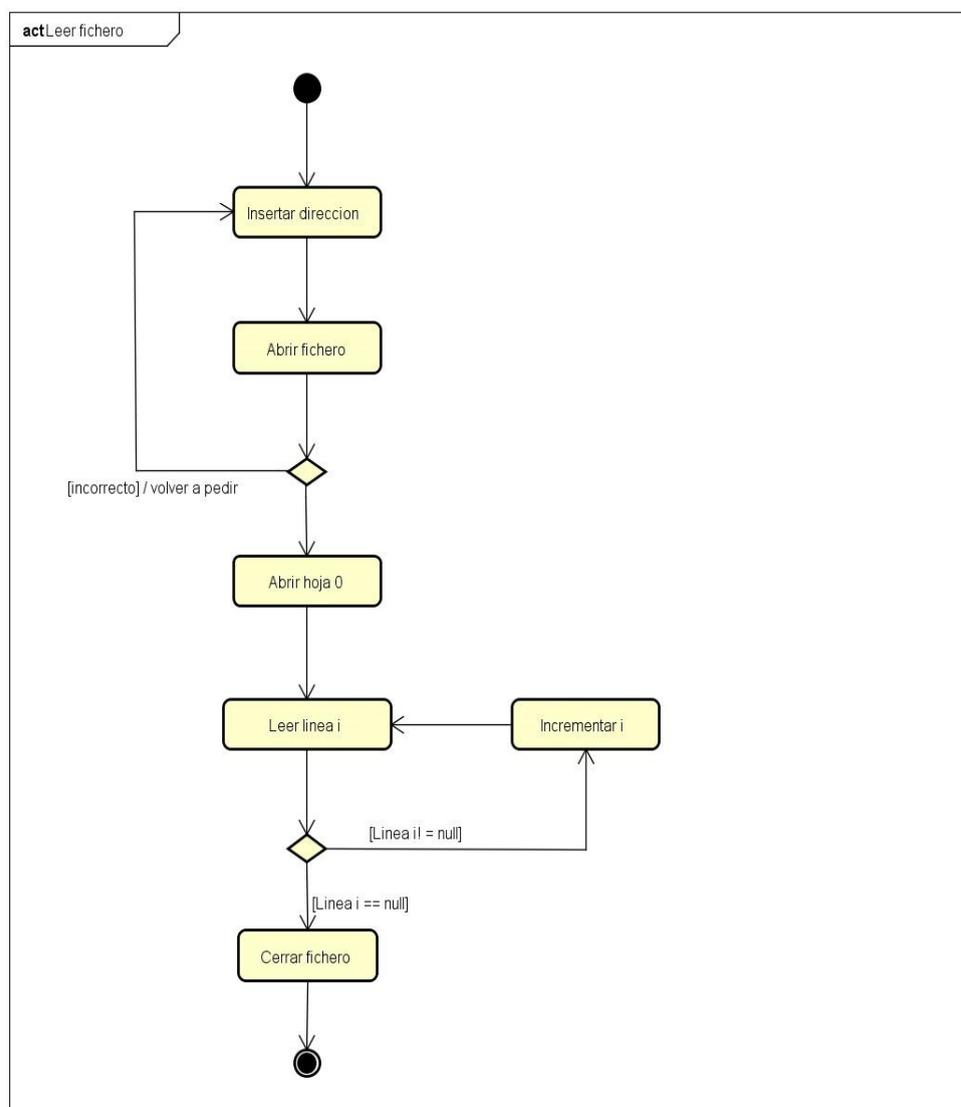


Figura 4.7: Diagrama de actividad para leer un fichero

4.7.7. Enviar Artículos

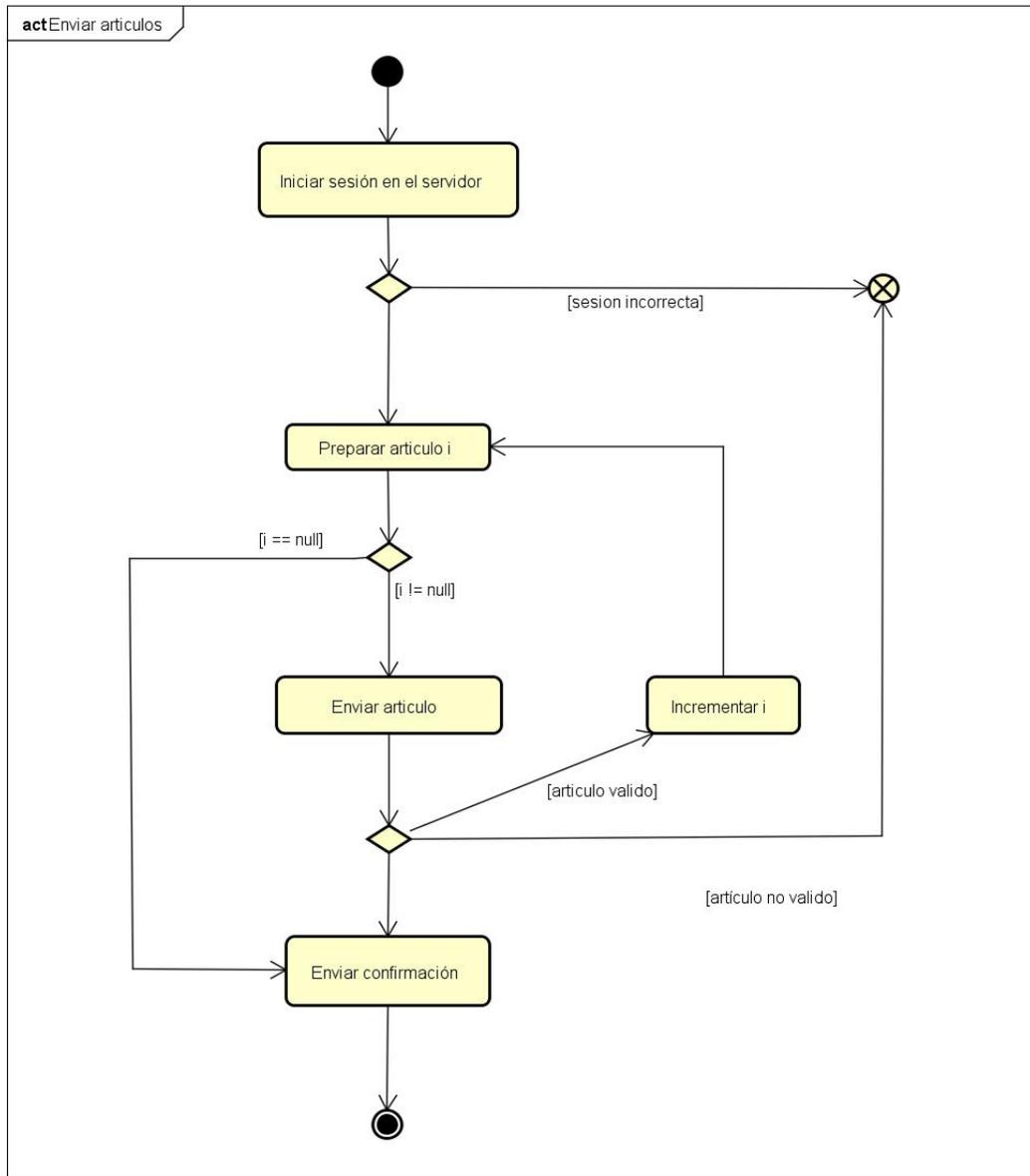


Figura 4.8: Diagrama de actividad para enviar artículos

4.8. Diagrama de Dominio

A continuación, se mostrará el diagrama de dominio resultante. En él podemos ver cuál es el alcance de nuestro sistema y las entidades que participan en él

Las entidades que en él participan son las siguientes:

- Usuario: representa a la persona que interactúa con la aplicación. La clase Usuario, será la encargada de la de los usuarios. Es una clase principal en el sistema ya que se estará instanciada durante toda la funcionalidad del mismo.

- Pedido: representa un pedido que ha realizado el usuario. En él se almacena todos los artículos pedidos por el usuario, así como el dinero disponible.
- Artículo: representa la entidad usada por la entidad pedido.

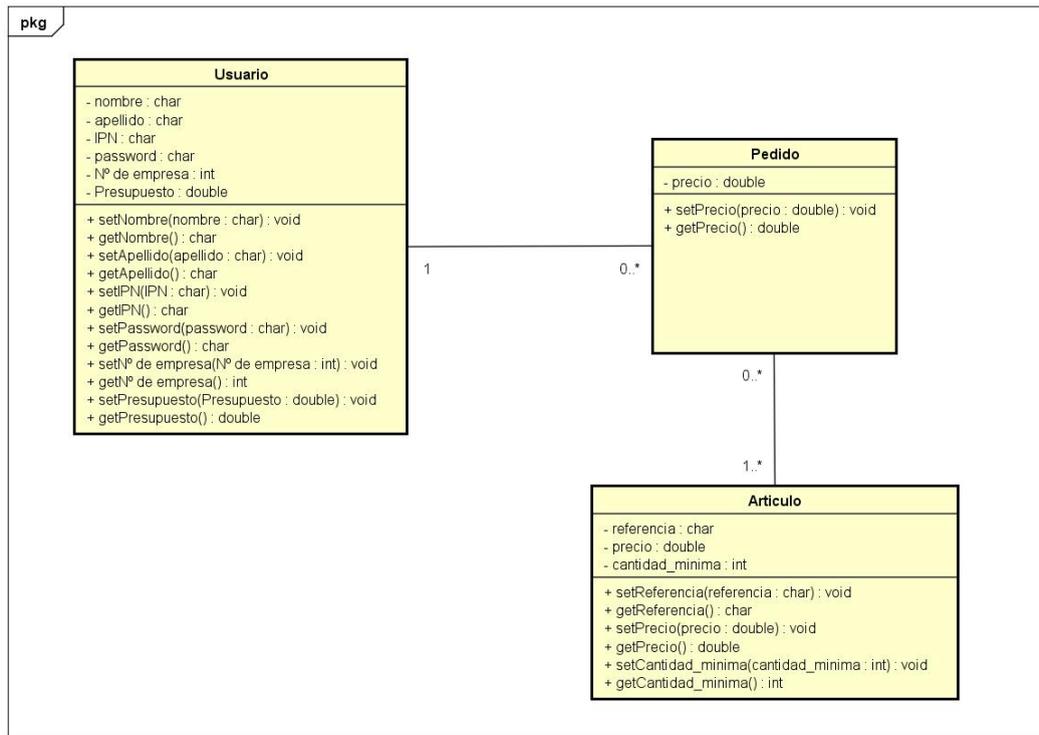


Figura 4.9: Diagrama de dominio

En este capítulo nos centraremos en realizar las trazas que van a ir dando forma a nuestro proyecto. Se presentarán temas importantes como la recuperación de datos y la arquitectura que albergará a nuestro sistema. Para finalizar se presentan los diferentes tipos de modelos que representan simplícidamente la forma que tendrá el proyecto.

5.1. Diseño de Persistencia

5.1.1. Decisiones Sobre el Diseño de Persistencia

Respecto a la persistencia de la información, solo la aplicación formulario requiere de esta funcionalidad. Esta aplicación necesita obtener información de los artículos y de los usuarios cada vez que se inicie. Al contrario de la aplicación envío, que no requiere cargar ningún tipo de información por defecto para poder funcionar pero sí necesita que el usuario se lo aporte, aunque no con la misma finalidad que con la primera.

5.1.1.1. Persistencia en la Aplicación Formulario

Debido a la cantidad de información que la aplicación ha de manejar se hace obligatorio tratar con una base de datos para el correcto manejo de la información. Debido al entorno de trabajo, la posibilidad de instalar una base de datos era totalmente imposible, ya que no había nada parecido instalado en el sistema. Ante esta situación no se pudo usar una base de datos por lo que hubo que planear una alternativa como solución.

Como alternativa, se ideó una hoja Excel para la consulta y la persistencia de los datos. Una de las razones para decantarnos por Office Excel y no por otros modelos, fue el alto uso y conocimiento que hacían de esta herramienta en la empresa. Conocimiento que permitiría al usuario administrador, la introducción de nuevos datos sobre el documento Excel. Otra razón importante es la seguridad, Office Excel incluye internamente un sistema de cifrado que podríamos usar para evitar alteraciones en la información por parte de personas no autorizadas para ello. Con esta solución no podemos hacer frente a un desarrollo de una base de datos, propiamente dicho. En muchas ocasiones no disponemos de una redundancia de datos 100% fiable, cosa que debemos hacer manualmente en la aplicación mediante código. A pesar de esto, hacemos una explicación de cómo hemos estructurado la información, asemejándose lo más posible a como lo haríamos en una base datos.

5.1.1.1.1. Descripción

Tabla n°1		
Usuario		
Autor	Santos Ángel Pardo Ramos	
Descripción	Esta tabla guarda información de los usuarios registrados en el sistema.	
Campos	Tipo	Descripción
Usuario	char	Correspondiente al IPN del usuario
Contraseña	char	Contraseña del usuario para acceder
UET	Int	Grupo de trabajo asociado al usuario

Tabla 5.1: Datos asociados al usuario

Tabla n°2		
Fechas		
Autor	Santos Ángel Pardo Ramos	
Descripción	Esta tabla guarda información acerca de las fechas de inicio y fin en las que se puede usar la aplicación	
Campos	Tipo	Descripción
Inicio	Int	Día de inicio del mes
Fin	Int	Día del fin de mes

Tabla 5.2: Datos asociados a las fechas

Tabla n°3		
Presupuesto		
Autor	Santos Ángel Pardo Ramos	
Descripción	Esta tabla guarda información del presupuesto que tiene asociado cada uno de los usuarios.	
Campos	Tipo	Descripción
Usuarios	char	Corresponde al IPN del usuario.
Presupuesto	Double	Cantidad de dinero que el usuario dispone para hacer el pedido cada mes.

Tabla 5.3: Datos asociados al presupuesto

Tabla n°4		
Artículo		
Autor	Santos Ángel Pardo Ramos	
Descripción	Esta tabla guarda información de todos los artículos de los que se tenga conocimiento	
Campos	Tipo	Descripción
Descripción	char	Descripción del artículo
Referencia	char	Referencia asociada a ese artículo
Precio	Double	Precio por unidad del artículo
Pedido mínimo	Int	Cantidad mínima que puede ser pedida

Tabla 5.4: Datos asociados a los artículos

Tabla n°5		
Referencias obligatorias		
Autor	Santos Ángel Pardo Ramos	
Descripción	Esta tabla guarda información de los usuarios registrados en el sistema	
Campos	Tipo	Descripción
Usuarios	char	Corresponde al IPN del usuario.
Referencia	char	Referencia obligatoria que el usuario debe de pedir.

Tabla 5.5: Datos asociados a los artículos obligatorios

5.1.1.2. Persistencia en la Aplicación Envío

En esta segunda aplicación, al contrario que en la primera, no se necesita que la aplicación se alimente por si sola de datos para poder funcionar. Carecemos por tanto de la necesidad de disponer de una “base de datos” que provea de información al sistema. Es el usuario el que elige los datos que él quiere para poder funcionar. Estos datos se corresponden con la salida de la primera aplicación. Están contenidos en un fichero Excel, cuya estructura es fija y se presenta a continuación.

5.1.1.2.1. Descripción

Entrada		
Usuario		
Autor	Santos Ángel Pardo Ramos	
Descripción	Esta entrada guarda los artículos que se van a enviar.	
Campos	Tipo	Descripción
Centro de gastos	char	Centro de gastos correspondiente al usuario.
Referencia	char	Referencia que identifica al artículo.
Descripción	char	Breve comentario sobre el artículo.
Cantidad	Int	Número de unidades que se van a pedir.

Tabla 5.6: Datos de entrada

5.2. Diseño Arquitectónico

5.2.1. Introducción

A continuación, se presentará con detalle el proceso de diseño de la arquitectura del sistema. Para ello, primero elaboraremos un diseño arquitectónico del sistema para comprender la estructura que éste tendrá. Después aplicaremos diferentes patrones de diseño que permitan así cumplir con una serie de condiciones expuestas en el punto siguiente.

Importante decir que ambas aplicaciones van a disponer de una arquitectura y patrones similares. Las diferentes estructuras elegidas ofrecen un comportamiento y unas singularidades acordes a las necesidades de ambas aplicaciones y por tanto del proyecto en si. Esto también ha permitido reducir la complejidad del sistema y hacer un conjunto de herramientas más unificado y con características internas similares.

5.2.2. Condiciones

Las siguientes condiciones que se exponen a continuación han de estar presentes a lo largo de todo el desarrollo arquitectónico. Al finalizar el desarrollo, estas han de cumplirse además de existir un equilibrio entre ellas.

- **Mantenimiento y rendimiento:** el sistema debe de disponer de un buen rendimiento así como de un mantenimiento cómodo. Ambas características deben de estar en equilibrio, disponiendo así de un buen rendimiento a pesar de que pequeños cambios en el software, mantenimiento, provoquen los menores trastornos posibles.
- **Protección:** el sistema debe proteger partes sensibles del sistema, en nuestro caso eso se centra en los datos.
- **Seguridad:** el sistema debe implementar políticas para asegurar la integridad, disponibilidad y confidencialidad de los datos.
- **Disponibilidad:** el sistema necesita información para funcionar, se debe reforzar esa disposición de los datos hacia el sistema.

5.2.3. Organización del Sistema

Un primer aspecto a analizar en este sistema, es como se ejecuta el sistema. Debido a que todo se ejecuta de manera local, solo se nos hace necesaria una sola computadora para ejecutar el sistema. Rehusamos así de una arquitectura distribuida, y no aplicamos modelos como el cliente-servidor o aplicación WEB. Debido a que la totalidad del software se ejecuta de manera local, disponemos entonces de un único subsistema. Nos referiremos a él como subsistema “local”.

La aplicación hace uso de la “base de datos” que hemos diseñado, se nos hace indispensable intentar aplicar un modelo de repositorio de datos, para que así, el subsistema local pueda hacer un uso eficiente de esta información.

Para finalizar, aplicaremos también un modelo de capas al sistema. Estas capas son la capa de aplicación, datos y presentación. Consiguiendo así una jerarquía en la que los datos se encuentran en la capa más interna, y para acceder a ellos debe existir una comunicación entre los diferentes niveles. De esta manera, el subsistema local queda particionado en tres partes, cada parte quedará representado por una capa que empleará una lógica.

- **Lógica de presentación:** su función recae en hacer de intermediario entre el sistema y el usuario. Muestra al usuario la información de una visualmente sencilla para el usuario y a la vez recoge las interacciones que este tenga con el sistema.
- **Lógica de negocio:** gestiona toda la lógica de la aplicación. En ella radica el proceso de todos los datos que recibe de la capa de presentación para que así se cumplan los objetivos de funcionamiento del sistema.
- **Lógica de persistencia:** se asocia con la última capa del modelo e interactúa con la base de datos intercambiando información.

Bien se podría haber diseñado soluciones alternativas que hubieran prestado un mejor servicio. Estas alternativas fueron rechazadas debido a las restricciones impuestas por parte de los clientes. Como en capítulos anteriores se ha mencionado, los sistemas de Renault cuentan con políticas muy estrictas respecto a lo que se puede o no se puede hacer. Debido a esto, se plantearon diseños alternativos para poder ofrecer la funcionalidad requerida.

5.2.4. Flujo de Control

Para controlar la transición entre las diferentes ventanas que contiene la aplicación, se recurre a un sistema de control centralizado, concretamente en una máquina de estados. Esta máquina de estados se encarga de la creación de las diferentes vistas del sistema, creando las nuevas y destruyendo las antiguas.

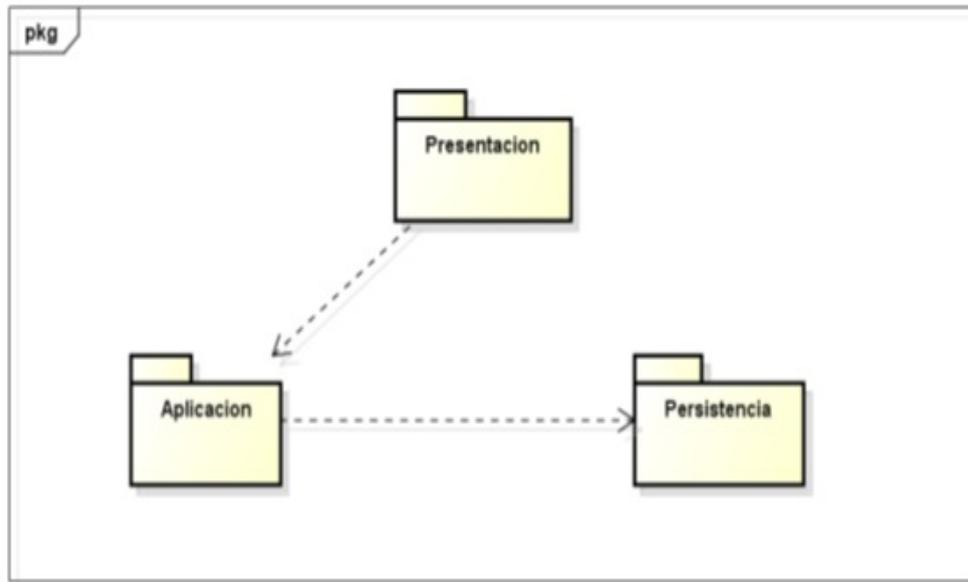


Figura 5.1: Arquitectura lógica

5.3. Modelos del Sistema

5.3.1. Patrones Utilizados

Los patrones que se presentan a continuación han sido utilizados durante la etapa de diseño, junto a ellos se incluye una explicación breve sobre su funcionamiento.

- MVC:** El Model-View-Controller (Modelo-Vista-Controlador) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores. Según Martin Fowler, escritor del libro *Pattern of Enterprise Application Architecture*, MVC queda catalogado dentro de unos de las siete categorías en las que clasifica los patrones, concretamente en “Patrones de Presentación Web”. Análogamente, Buschman, autor del libro *Pattern Oriented Software Architecture*, sitúa MVC dentro de una de las cuatro categorías que describe, “Sistemas Interactivos” [5].

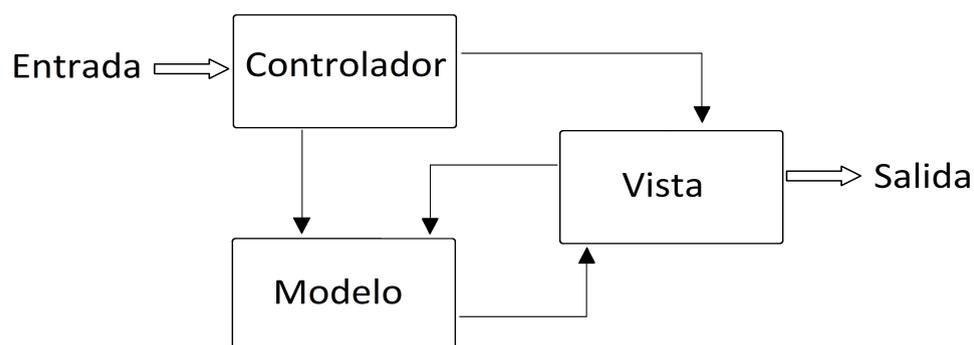


Figura 5.2: Representación gráfica del MVC.

MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Este modelo podemos verlo de manera genérica en figura 5.2. Para esto, utiliza las siguientes abstracciones:

- **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista (View):** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio por el modelo o la vista. En nuestro caso, utilizaremos la vista.

Existen variantes del patrón MVC en función de cuál es la relación entre cada una de las tres partes. Nosotros hemos decidido desarrollar la que se muestra en la siguiente figura 5.3.

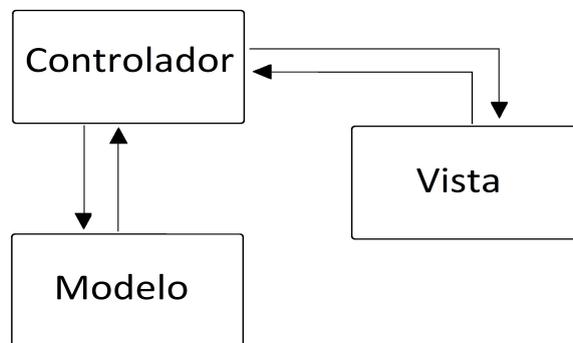


Figura 5.3: MVC

Este modelo de MVC carece de interacción entre la vista y el modelo, siendo el centro de todo el controlador. Los pasos que rigen su funcionamiento son:

1. El usuario interactúa con la interfaz de usuario.
2. El controlador recibe (por parte de los objetos de la interfaz- vista) la notificación de la acción solicitada por el usuario.
3. El controlador ejerce la lógica de funcionamiento y accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del controlador para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

La utilización del patrón MVC aporta los siguientes beneficios a nuestro sistema:

- Mayor cohesión
- Las vistas ofrecen una mayor flexibilidad y agilidad
- Mayor claridad en el diseño
- Mayor facilidad de mantenimiento

- Una gran escalabilidad a nuestro sistema
- **DataMapper:** El patrón dataMapper es un patrón que permite separar de forma clara, el modelo de dominio de la persistencia de datos. Cada clase perteneciente al modelo de dominio debe asociarse con una clase dataMapper para gestionar los datos entre la capa de lógica de la aplicación y la persistencia de los datos. Las clases dataMapper son las encargadas de insertar, modificar y recuperar los objetos desde la base de datos hacia la memoria de la aplicación.

La aplicación de este patrón busca obtener un bajo acoplamiento entre los objetos y el conocimiento persistentes. Incluimos también la necesidad de tener una alta cohesión y un fácil mantenimiento. La utilización de este patrón supone una correspondencia indirecta, ya que usamos otros objetos para establecer la correspondencia con los objetos persistentes [5].



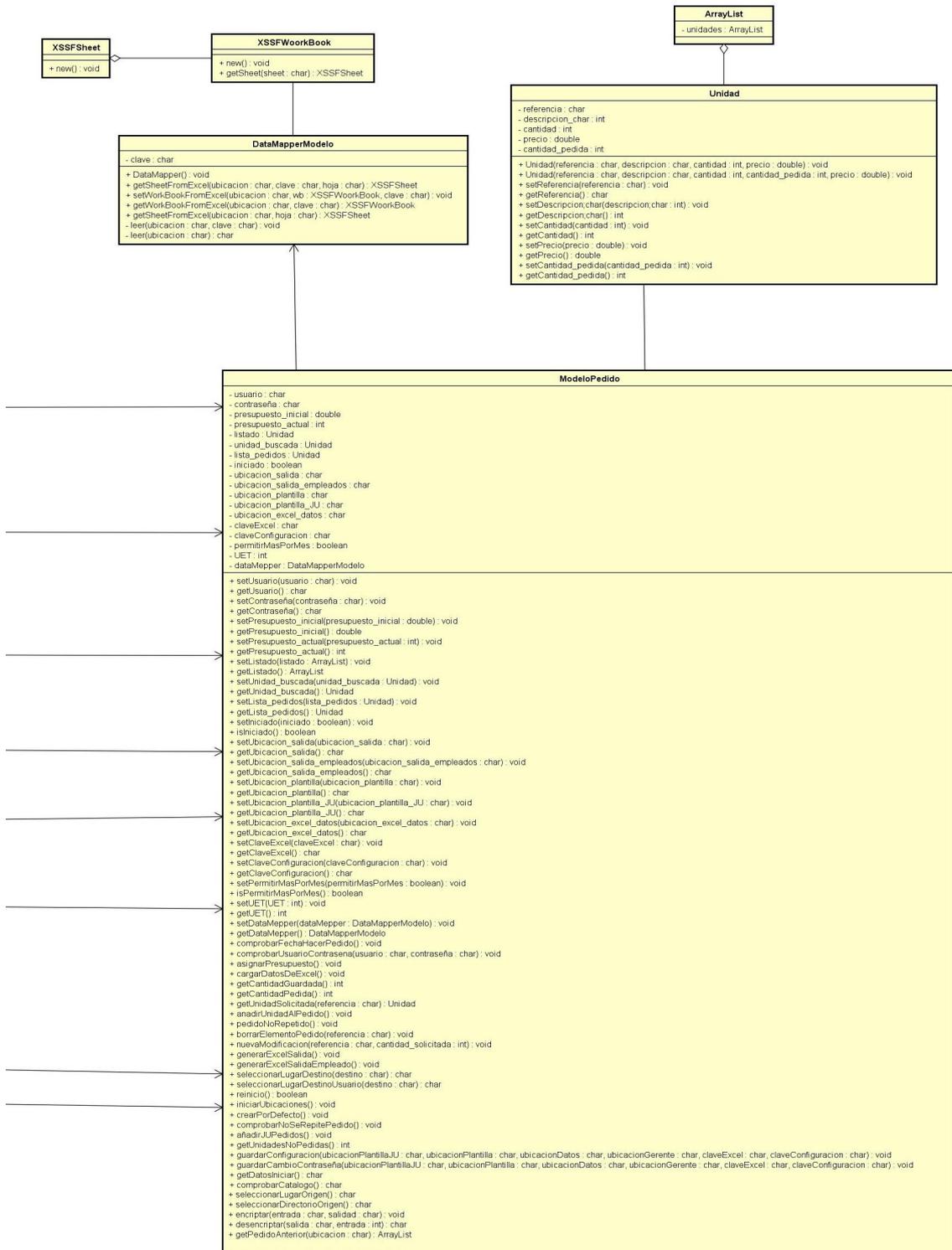
Figura 5.4: Descripción gráfica de DataMapper.

5.3.2. Modelos Estáticos

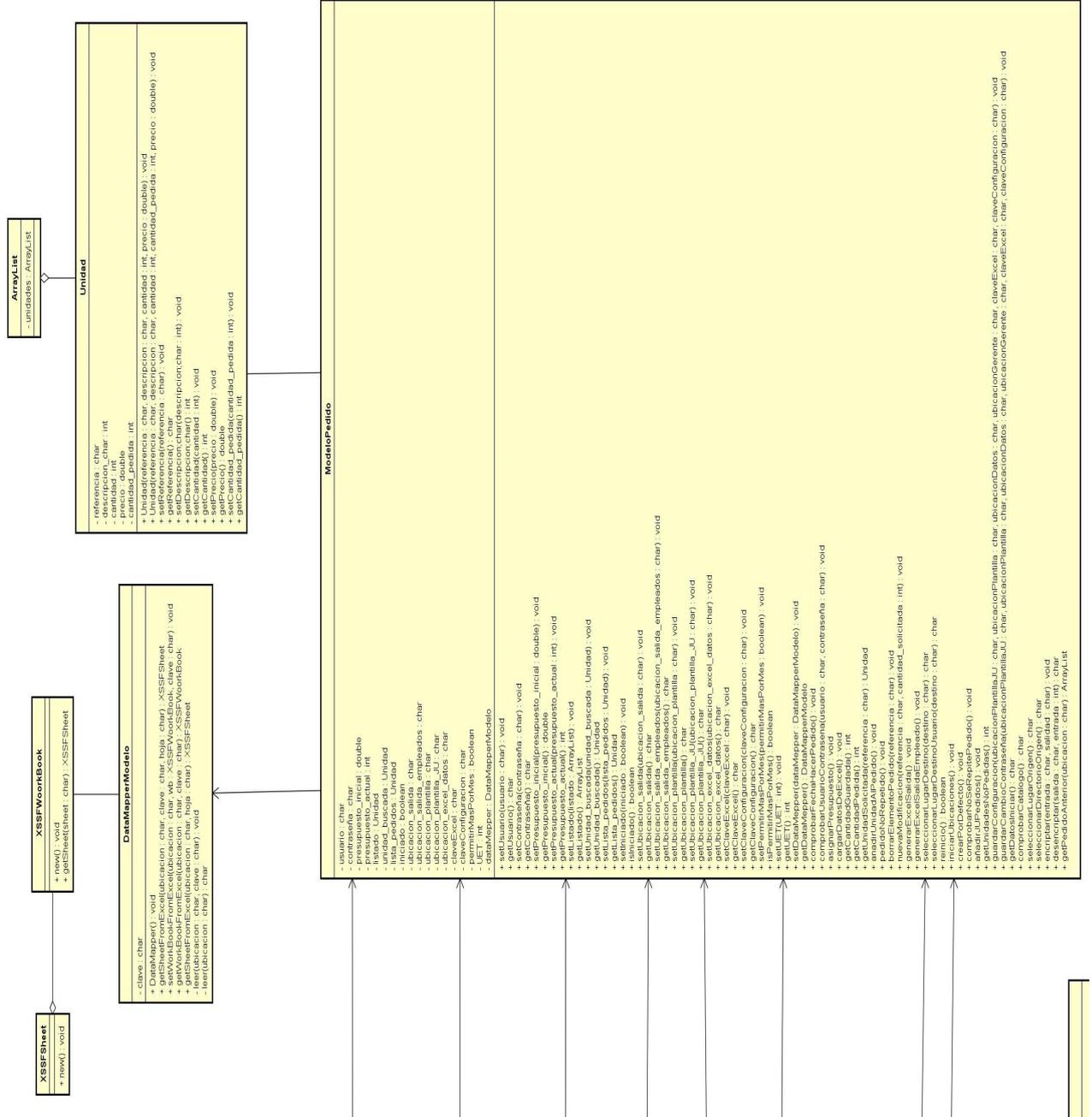
Hemos empleado la misma arquitectura y los mismos patrones para ambas aplicaciones. Pero debido a que ambas aplicaciones cumplen funcionalidades diferentes crean modelos con el mismo perfil pero con diferencias internas entre ellos. Esto hace que los modelos se presenten por separado.

5.3.2.1. Aplicación Formulario

En el apartado que se presenta a continuación se documentan los diagramas de clases de diseño. Como el diagrama de clases es bastante grande, se ha dividido en varias secciones, siendo separadas éstas en vistas, modelos y controladores.



5.3.2.1.2. Diagrama de clases del modelo



5.3.2.1.5. Controladores

5.3.2.1.5.1. Clase Controlador

Clase Controlador	
Definición	La clase controlador es una clase abstracta, la cual es heredada por las demás clases controladoras.
Métodos	Descripción
Controlador(ModeloPedido mimodelo)	Constructor de la clase

Tabla 5.7: Clase controlador

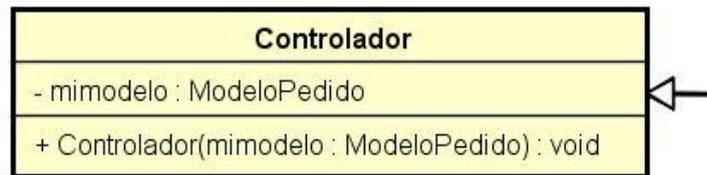


Figura 5.5: Clase controlador

5.3.2.1.5.2. Clase ControladorAdministracionSesion

Clase ControladorAdministracionSesion	
Definición	La clase ControladorAdministracionSesion se encarga de validar la entrada para acceder a la configuración de la aplicación.
Métodos	Descripción
ControladorAdministracionSesion (VistaAdministracionSesion mivista, ModeloPedido mimodelo)	Constructor de la clase.
enviar (char clave): void	Método que recibe como parámetro la clave introducida y controla el acceso.

Tabla 5.8: Clase ControladorAdministracionSesion

5.3.2.1.5.3. Clase ControladorConfiguracion



Figura 5.6: Clase ControladorAdministracionSesion

Clase ControladorConfiguracion	
Definición	La clase ControladorConfiguracion se encarga de validar la entrada para acceder a la configuración de la aplicación.
Métodos	Descripción
ControladorConfiguracion (VistaConfiguracion mivista, ModeloPedido mimodelo)	Constructor de la clase
cambiarContrasenaExcel (char contraseña_vieja, char contraseña_nueva, char contraseña_nueva2):void	Método que recibe como parámetro la clave introducida y controla el acceso.
cambiarPermisoMes(boolean estado) :void	Método que se encarga de cambiar el estado que permite realizar más de un pedido al mes o no.
cambiarSalidaGerente(char ubicacion) :void	Método que se encarga de cambiar la ubicación en la que se genera el Excel para el administrador.
comprobarCatalogo():void	Método que lee el fichero Excel para obtener los artículos y muestra por pantalla que líneas de la hoja Excel han sido añadidas y cuáles no.
guardar(char ubicacionPlantillaJU, char ubicacionPlantilla, char ubicacionDatos, char ubicacionGerente, char claveExcel, char claveConfiguracion) :void	Método que recoge los cambios realizados en las diferentes configuraciones y realiza las gestiones necesarias para que sean efectivos.
guardarCambioContraseña():void	Método que recoge los cambios realizados en las contraseñas de los ficheros y realiza las gestiones necesarias para que sean efectivos.
iniciar():void	Método llamado al iniciarse la vista y que pone los valores a mostrar en los diferentes campos.
resetear():void	Método que cambia toda la configuración a un estado predeterminado.
salir():void	salir():void Método que finaliza la ejecución del programa.
seleccionarUbicacionFicheroDatos(char ubicacion) :void	Método que recoge el cambio de ubicación del fichero de datos.
seleccionarUbicacionPlantilla(char ubicacion) :void	Método que recoge el cambio de ubicación de la plantilla.
seleccionarUbicacionPlantillaJU(char ubicacion) :void	Método que recoge el cambio de ubicación de la plantilla del JU.

Tabla 5.9: Clase ControladorConfiguración

5.3.2.1.5.4. Clase ControladorConfirmacion

ControladorConfiguracion
- mivista : VistaConfiguracion
+ ControladorConfiguracion(mivista : VistaConfiguracion, mimodelo : int) : ModeloPedido + seleccionarUbicacionPlantillaJU(ubicacion : char) : void + seleccionarUbicacionPlantilla(ubicacion : char) : void + seleccionarUbicacionFicheroDatos(ubicacion : char) : void + guardar(ubicacionPlantillaJU : char, ubicacionPlantilla : char, ubicacionDatos : char, ubicacionGerente : char, claveExcel : char, claveConfiguracion : char) : void + guardarCambioContraseña() : void + iniciar() : void + cambiarContraseña(contraseña_vieja : char, contraseña_nueva : char, contraseña_nueva2 : char) : void + resetear() : void + cambiarContraseñaExcel(contraseña_vieja : char, contraseña_nueva : char, contraseña_nueva2 : char) : void + salir() : void + cambiarSalidaGerente(ubicacion : char) : void + comprobarCatalogo() : void + cambiarPermisoMes(estado : boolean) : void

Figura 5.7: Clase ControladorConfiguración

Clase ControladorConfirmación	
Definición	La clase ControladorAdministracionSesion se encarga de validar la entrada para acceder a la configuración de la aplicación
Métodos	Descripción
ControladorConfirmacion (VistaConfirmacion mivista, ModeloPedido mimodelo)	Constructor de la clase
enviar (char clave): void	Método que recibe como parámetro la clave introducida y controla el acceso.
btnSi():void	Método llamado cuando se quiere confirmar el pedido.
btnNo():void	Método llamado cuando se desea cambiar algo del pedido.

Tabla 5.10: Clase ControladorConfirmación

ControladorConfirmacion
- mivista : VistaConfirmacion
+ ControladorConfirmacion(mivista : VistaConfirmacion, mimodelo : ModeloPedido) : void + btnSi() : void + cambiarAPunteroEspera() : void + btnNo() : void + Finalizar() : void + getNoPedidos() : ArrayList

Figura 5.8: Clase ControladorConfirmación

5.3.2.1.5.5. Clase ControladorControladorHacerPedido

Clase ControladorHacerPedido	
Definición	La clase ControladorHacerPedido se encarga de gestionar los pasos para ir añadiendo artículos al pedido o modificando los que ya han sido añadidos.
Métodos	Descripción
ControladorHacerPedido (VistaHacerPedido, ModeloPedido mimodelo)	Constructor de la clase
iniciaVista():void	Método llamado al iniciarse la vista y que muestra por pantalla los datos necesarios.
buscar(char referencia, int cantidad_solicitada):void	Método utilizado para buscar un artículo.
borrarPedido(char referencia):void	Método utilizado para borrar un artículo que ha sido añadido.
Modificar(char referencia, int cantidad):void	Método utilizado para modificar un artículo que ha sido añadido.

Tabla 5.11: Clase ControladorHacerPedido

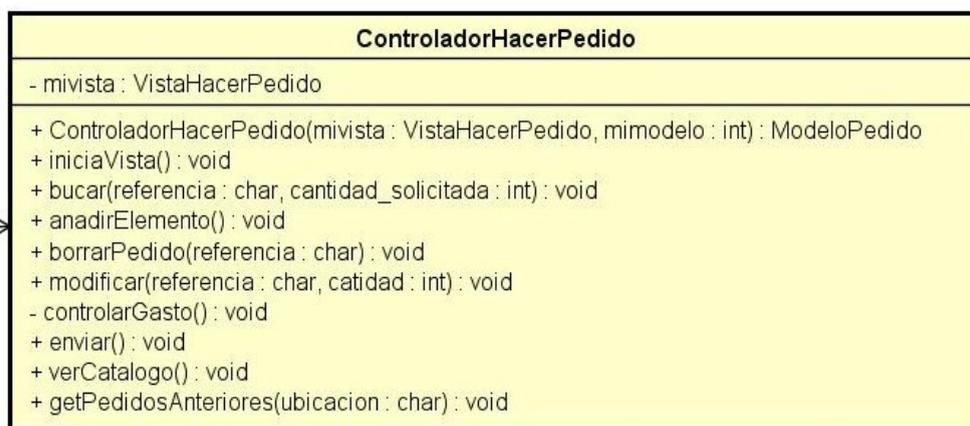


Figura 5.9: Clase ControladorHacerPedido

5.3.2.1.5.6. Clase ControladorInicioSesion

Clase ControladorInicioSesion	
Definición	La clase ControladorInicioSesion se encarga de controlar el inicio de sesión.
Métodos	Descripción
ControladorInicioSesion (VistaInicioSesion mivista, ModeloPedido mimodelo)	Constructor de la clase
iniciarSesion(char usuario, char contraseña):void	Método que recibe como parámetro el usuario y la contraseña introducidos y valida el inicio de sesión.
iniciarSesionAdministracion():void	Método llamado cuando se quiere acceder a la configuración de la administración.
iniciarUbicaciones():void	Método llamado al iniciarse el controlador y que chequea los datos del fichero Excel para poder ser utilizado.

Tabla 5.12: Clase ControladorInicioSesión



Figura 5.10: Clase ControladorInicioSesión

5.3.2.1.5.7. Clase ControladorPantallaMenu

Clase ControladorPantallaMenu	
Definición	La clase ControladorPantallaMenu se encarga de controlar la pantalla en la que el usuario decide cómo hacer el pedido.
Métodos	Descripción
ControladorPantallaMenu (VistaPantallaMenu mivista, ModeloPedido mimodelo)	Constructor de la clase
hacerPedido():void	Método llamado cuando el usuario desea hacer un pedido nuevo añadiendo los artículos uno a uno.
verPedidosAnteriores():void	Método llamado cuando el usuario desea hacer un pedido similar a otro anterior.

Tabla 5.13: Clase ControladorPantallaMenu



Figura 5.11: Clase ControladorPantallaMenu

5.3.2.1.5.8. Clase ControladorVerCatalogo

Clase ControladorVerCatalogo	
Definición	La clase ControladorVerCatalogo se encarga de controlar la pantalla en la que el usuario puede ver la lista de artículos disponibles.
Métodos	Descripción
ControladorVerCatalogo (VistaVerCatalogo mivista, ModeloPedido mimodelo)	Constructor de la clase
iniciaVistaCatalogo():void	Método llamado al iniciarse la vista para que se recojan todos los artículos disponibles y se muestren por pantalla.
volver():void	Método llamado cuando el usuario desea volver a la vista de hacer el pedido.
volver(char referencia):void	Método llamado cuando el usuario hace doble clic sobre un artículo, para ello se recoge su referencia y se devuelve a la vista de hacer pedido.

Tabla 5.14: Clase ControladorVerCatalogo



Figura 5.12: Clase ControladorVerCatalogo

5.3.2.1.6. DataMapper

5.3.2.1.6.1. Clase DataMapperModelo

Clase DataMapperModelo	
Definición	La clase ControladorInicioSesion se encarga de controlar el inicio de sesión.
Métodos	Descripción
DataMapper():void	Constructor de la clase
getSheetFromExcel(char ubicación, char clave, char hoja):XSSFSheet	Método que devuelve una hoja de un fichero Excel cifrado con una clave.
getSheetFromExcel(char ubicación, char hoja):XSSFSheet	Método que devuelve una hoja de un fichero Excel.
getWorkBookFromExcel(char ubicación, char clave):XSSFWorkBook	Método que devuelve un libro Excel entero cifrado con una clave.
setWorkBookFromExcel(char ubicación, XSSF-WorkBook wb, char clave):void	Método que actualiza un libro Excel y lo cifra con una clave.
leer(char ubicación, char clave):void	Método que lee el contenido de un fichero que este cifrado con clave.
leer(char ubicación):char	Método que lee el contenido de un fichero.

Tabla 5.15: Clase **DataMapperModelo**

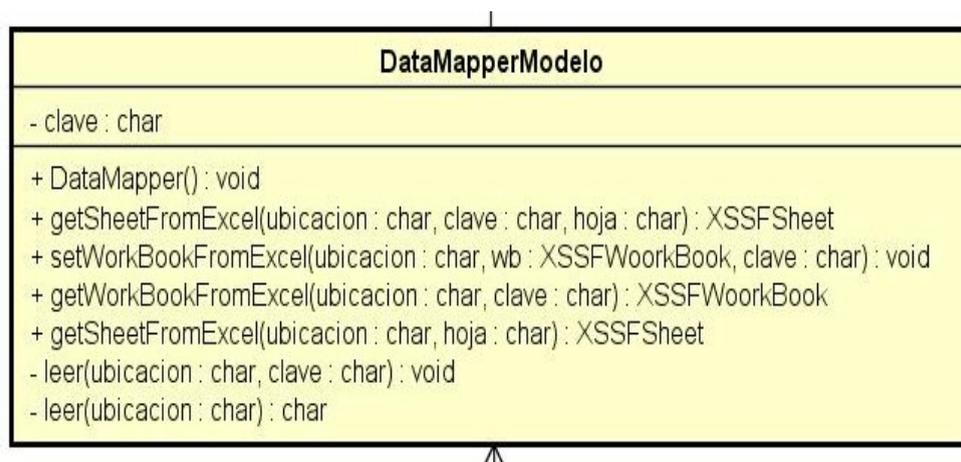


Figura 5.13: Clase **DataMapperModelo**

5.3.2.2. Aplicación Envío

5.3.2.2.1. Diagrama de clases global

5.3.2.2.2. Diagrama de clases del modelo

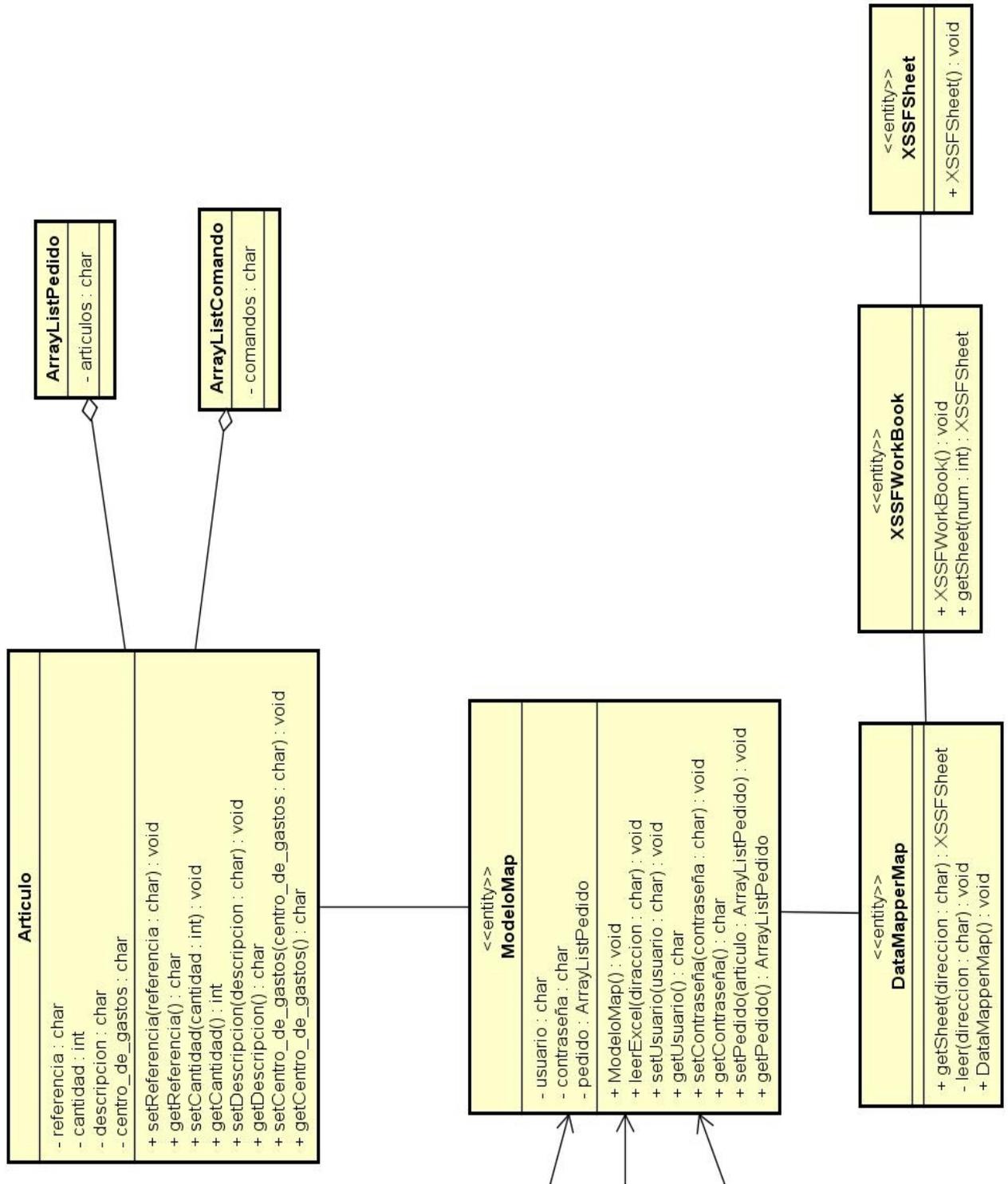


Figura 5.15: Clase controlador

5.3.2.2.3. Diagrama de clases de las vistas

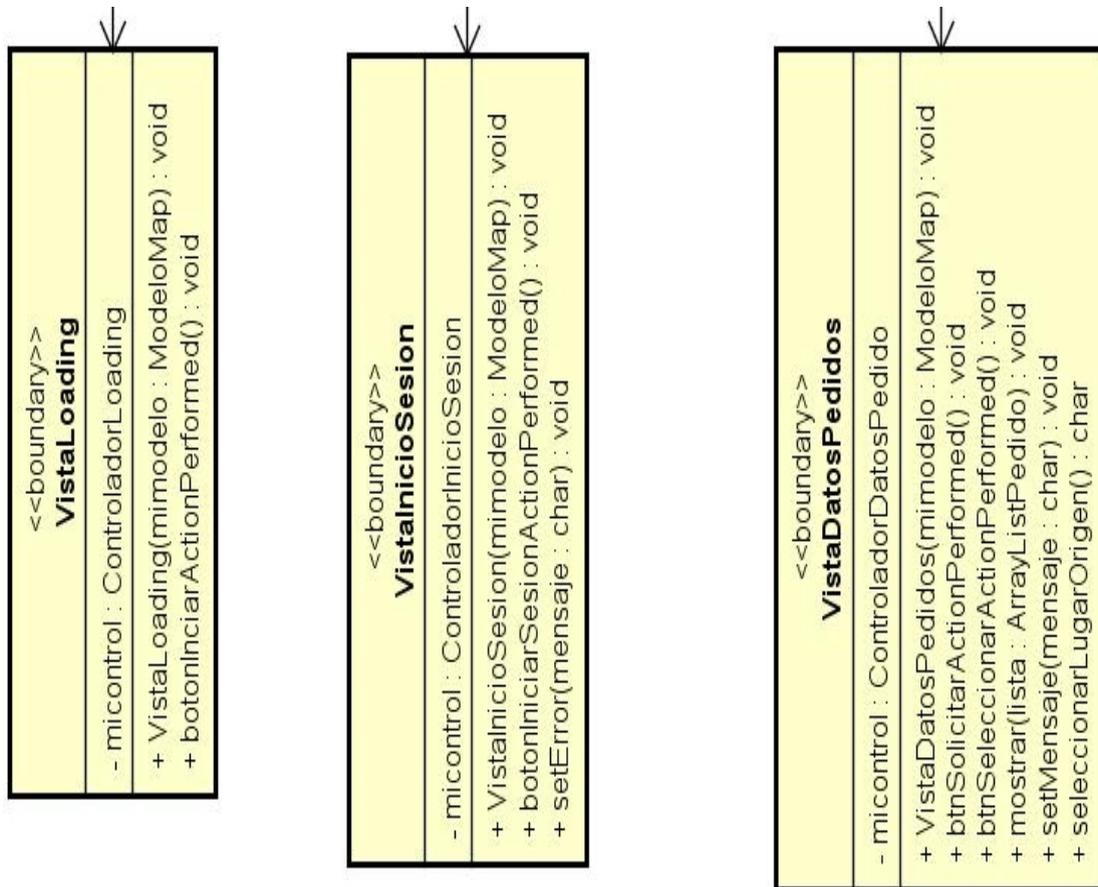


Figura 5.16: Clase vista

5.3.2.2.4. Diagrama de clases de los controladores (Hilos)

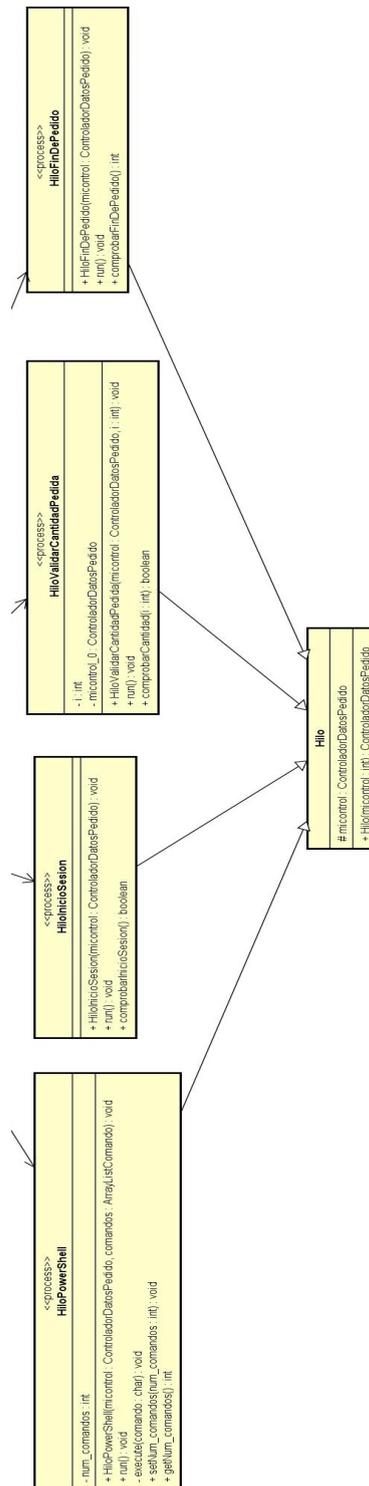


Figura 5.17: Clase controlador

5.3.2.2.5. Diagrama de clases de los controladores

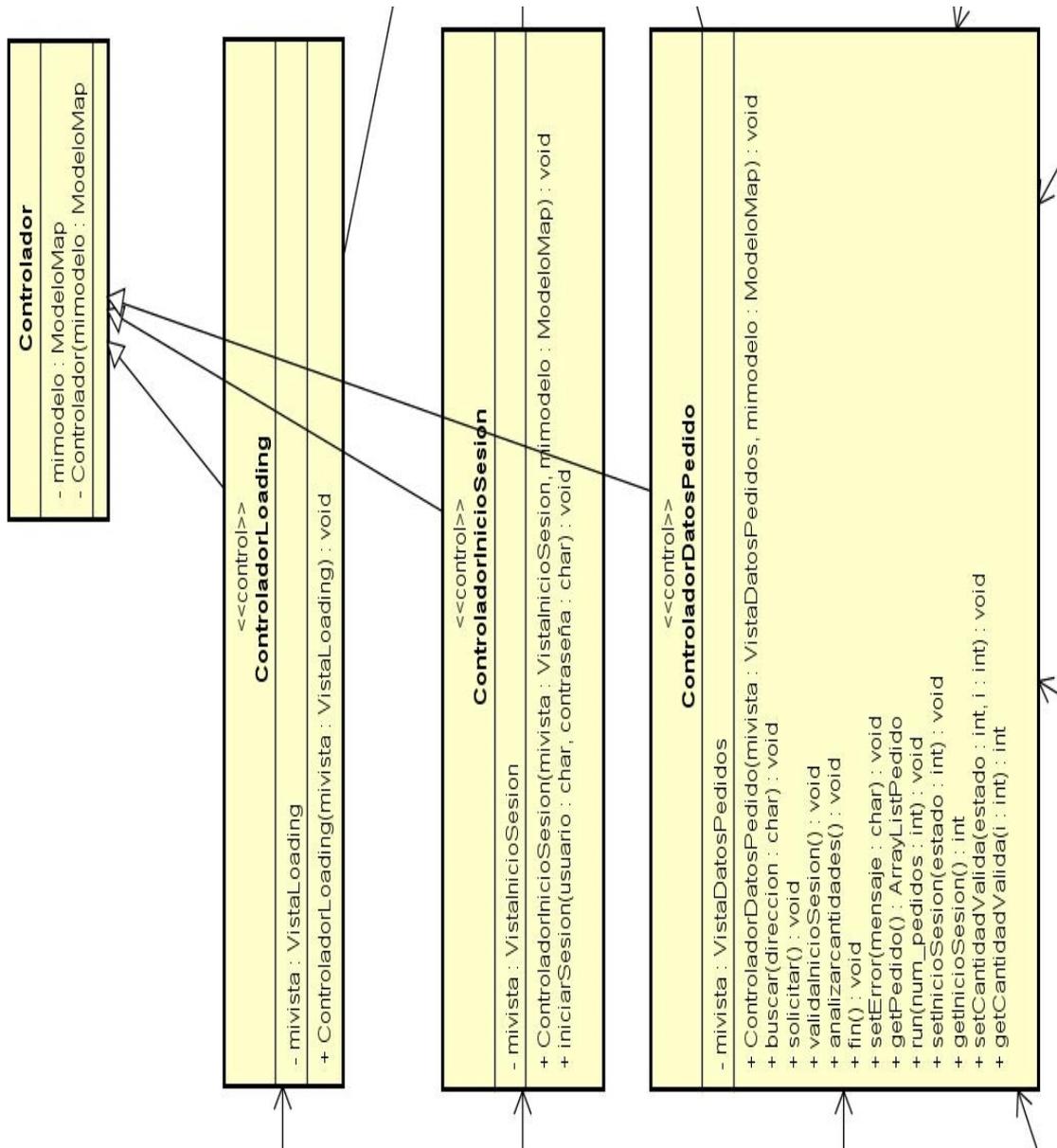


Figura 5.18: Clase controlador

5.3.2.2.6. Controladores

5.3.2.2.6.1. Clase Controlador

Clase Controlador	
Definición	La clase controlador es una clase abstracta, la cual es heredada por las demás clases controladoras
Métodos	Descripción
Controlador(ModeloMap mimodelo)	Constructor de la clase

Tabla 5.16: Clase controlador



Figura 5.19: Clase controlador

5.3.2.2.6.2. Clase ControladorLoading

Clase ControladorLoading	
Definición	La clase ControladorLoading es una clase que se encarga de controlar la primera pantalla, la que hace de presentación a la aplicación
Métodos	Descripción
ControladorLoading(VistLoading mivista)	Constructor de la clase

Tabla 5.17: Clase ControladorLoading

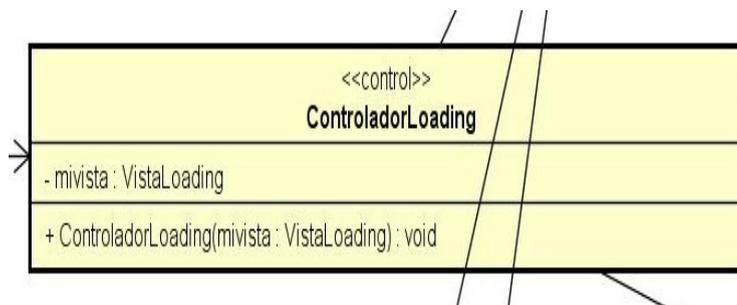


Figura 5.20: Clase ControladorLoading

5.3.2.2.6.3. Clase ControladorIniciarSesion

Clase ControladorIniciarSesion	
Definición	La clase ControladorIniciarSesion es una clase que se encarga de controlar la gestión del inicio de sesión
Métodos	Descripción
ControladorInicioSesion(VistaInicioSesion mivista, ModeloMap mimodelo)	Constructor de la clase
IniciarSesion(char usuario, char contraseña)	Método llamado por la vista correspondiente, el cuál avisa al controlador de que se ha pulsado el botón para iniciar sesión, recibe el usuario y la contraseña como parámetro y los añade al modelo.

Tabla 5.18: Clase ControladorInicioSesion

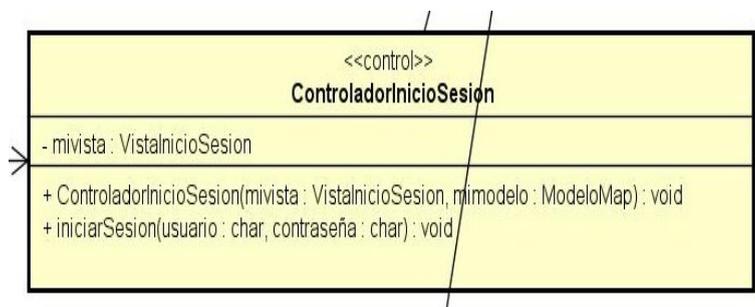


Figura 5.21: Clase ControladorInicioSesion

5.3.2.2.6.4. Clase ControladorDatosPedido

Clase ControladorDatosPedido	
Definición	La clase ControladorDatosPedido es una clase que se encarga de controlar la gestión que supone recoger los datos y establecer la comunicación con el servidor. Se ayuda de otras clases, Hilos, para realizar su función.
Métodos	Descripción
ControladorDatosPedido(VistaDatosPedido mivista, ModeloMap mimodelo)	Constructor de la clase
buscar(char direccion)	Método llamado por la vista correspondiente, la cuál le avisa de que se ha seleccionado un fichero, dirección pasada como parámetro, y se procede a leerlo para obtener los artículos deseados.
solicitar()	Método llamado por la vista correspondiente, la cuál le avisa de que se ha pulsado el botón de enviar. Se procede a activar los mecanismos para realizar el envío al servidor de los artículos deseados.
validaInicioSesion()	Método llamado por el HiloPowerShell para avisar al controlador de que ya se ha enviado la información de inicio de sesión al servidor y se lanza el HiloInicioSesión para comprobarlo.
analizarCantidades()	Método llamado por el HiloPowerShell para avisar al controlador de que ya se ha enviado la información de los artículos y sus cantidades al servidor y se puede proceder a lanzar los hilos de HiloAnalizarCantidades para comprobarlo.
fin()	Método llamado por el HiloPowerShell para avisar al controlador de que ya se ha enviado la información de fin del pedido y su aceptación por lo que se puede lanzar ya el hilo HiloFinDePedido que valide esto.
setError(char mensaje)	Método llamado por cualquiera de los Hilos para escribir un mensaje por pantalla.
getPedido()	Función del controlador que devuelve el listado de artículos que se van a pedir así como su cantidad y centro de gastos.
run(int num_pedidos)	Método llamado que genera la lista de comandos a ejecutar en PowerShell.
setInicioSesion(int estado)	Método llamado por el HiloInicioSesion para colocar el estado final del inicio de sesión, si ha sido válido o no. El estado de inicio de sesión inicial es un 0, si este es aceptado se añade pone un 1 y sino un 2.
getInicioSesion()	Función llamada por el HiloPowerShell para comprobar el estado del inicio de sesión. Se continua o no en función del resultado.
setCantidadValida(int estado, int i)	Método llamado por el HiloCantidadValida para dar un validez a un artículo o no. Cada artículo tiene asociado un estado 0 inicial, si es válido se le asigna un 1 y sino un 2.
getCantidadValida(int i)	Función llamada por el HiloPowerShell para saber el estado de un artículo. Se continua o no en función del resultado.

Tabla 5.19: Clase ControladorDatosPedido

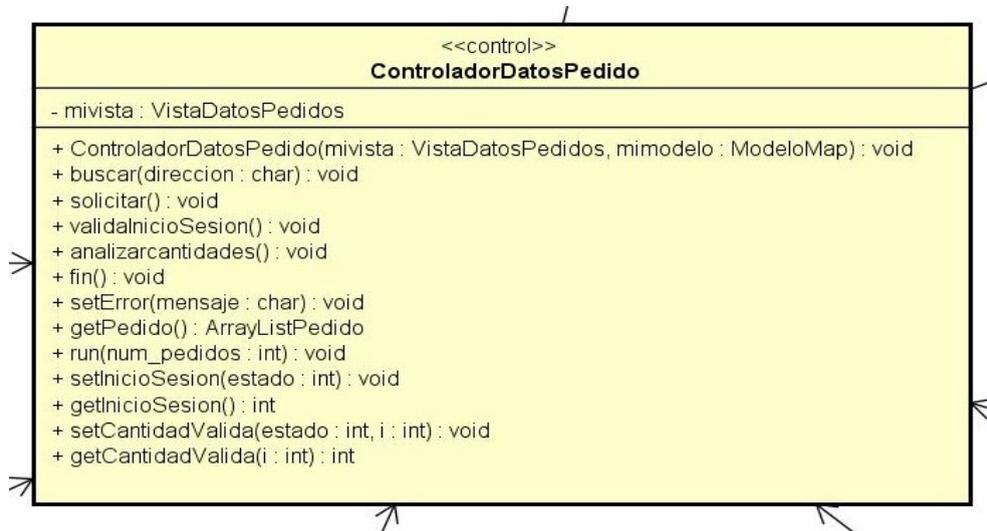


Figura 5.22: Clase ControladorDatosPedido

5.3.2.2.7. Hilo

5.3.2.2.7.1. Clase Hilo

Clase Hilo	
Definición	La clase Hilo es una clase abstracta heredada por las demas clases Hilo.
Métodos	Descripción
Hilo(ControladorDatosPedido micontrol)	Constructor de la clase

Tabla 5.20: Clase Hilo

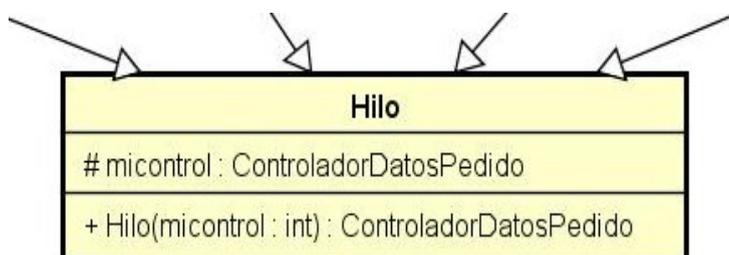


Figura 5.23: Clase Hilo

5.3.2.2.7.2. Clase HiloPowerShell

Clase HiloPowerShell	
Definición	La clase HiloPowerShell es una clase que implementa un hilo que se encarga de crear un objeto PowerShell e ir ejecutando los diferentes comandos. Interactúa con el ControladorDatosPedido para sincronizar el proceso.
Métodos	Descripción
HiloPowerShell(ControladorDatosPedido mi-control, ArrayList<char>comandos)	Constructor de la clase
run()	Método llamado automáticamente al iniciarse. Pone en ejecución el hilo.
execute(char comando)	Método llamado cada vez que se desea ejecutar un comando en el PowerShell. Recibe como parámetro el comando a realizar.
setNum_comandos(int num_comandos)	Método que asigna el número de comandos.
getNum_comandos()	Método que devuelve el número de comandos asignados.

Tabla 5.21: Clase HiloPowerShell

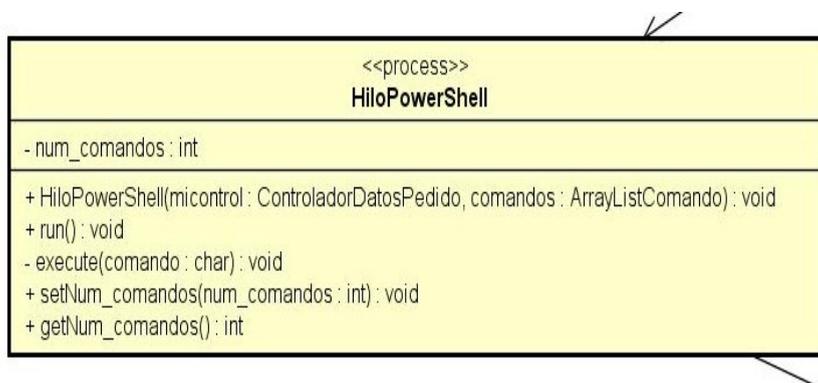


Figura 5.24: Clase HiloPowerShell

5.3.2.2.7.3. Clase HiloInicioSesión

Clase HiloInicioSesion	
Definición	La clase HiloInicioSesion es una clase que implementa un hilo que se encarga de validar o no el inicio de sesión y comunicarlo al ControladorDatosPedidos para sincronizar el proceso.
Métodos	Descripción
HiloInicioSesion(ControladorDatosPedido micontrol)	Constructor de la clase
run()	Método llamado automáticamente al iniciarse. Pone en ejecución el hilo.
comprobarInicioSesion()	Método llamado desde el propio hilo el cual sirve para empezar a comprobar si se ha iniciado sesión y comunicarlo al controlador correspondiente.

Tabla 5.22: Clase HiloInicioSesion

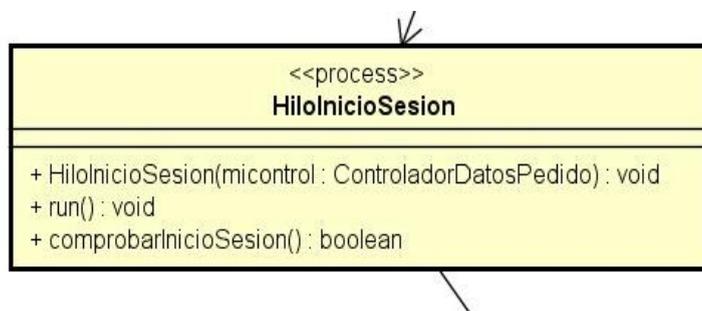


Figura 5.25: Clase HiloInicioSesion

5.3.2.2.7.4. Clase HiloValidarCantidadPedida

Clase HiloValidarCantidadPedida	
Definición	La clase HiloValidarCantidadPedida es una clase que implementa un hilo que se encarga de validar si las cantidades solicitadas son válidas y aceptadas por el servidor, y estas respuestas son comunicadas al ControladorDatosPedidos para sincronizar el proceso.
Métodos	Descripción
HiloValidarCantidadPedida(ControladorDatosPedido micontrol, int i)	Constructor de la clase
run()	Método llamado automáticamente al iniciarse. Pone en ejecución el hilo.
comprobarCantidad()	Método llamado desde el propio hilo el cual sirve para empezar a comprar si cada uno de los artículos pedidos ha sido aceptado por el servidor correctamente y se lo comunica al controlador.

Tabla 5.23: Clase HiloValidarCantidadPedida

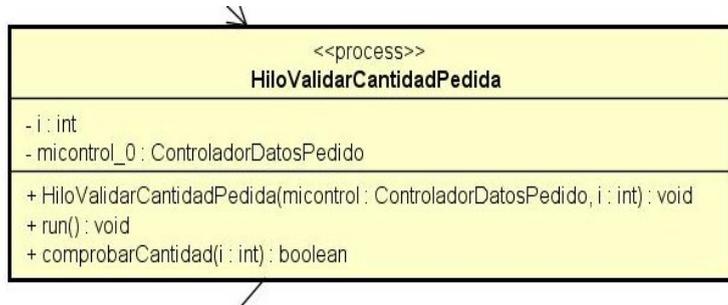


Figura 5.26: Clase HiloValidadorCantidadPedida

5.3.2.2.7.5. Clase HiloFinDePedido

Clase HiloFinDePedido	
Definición	La clase HiloFinDePedido es una clase que implementa un hilo que se encarga de validar la confirmación final del envío de los artículos al servidor, y comunicar la respuesta al ControladorDatosPedidos para sincronizar el proceso.
Métodos	Descripción
HiloFinDePedido(ControladorDatosPedido micontrol)	Constructor de la clase
run()	Método llamado automáticamente al iniciarse. Pone en ejecución el hilo.
comprobarFinDePedido()	Método llamado desde el propio hilo el cual sirve para empezar a comprar si la confirmación enviada al servidor es correcta y se lo comunica al controlador.

Tabla 5.24: Clase HiloFinDePedido

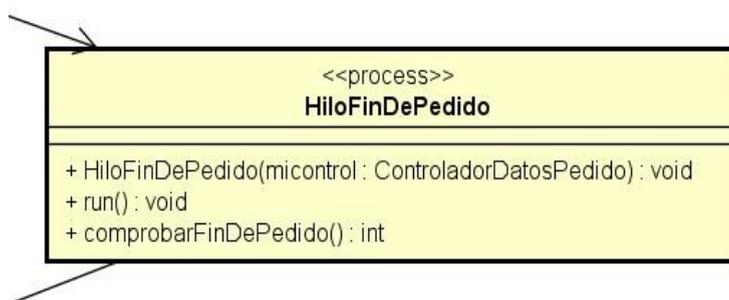


Figura 5.27: Clase HiloFinDePedido

5.3.3. Modelos Dinámicos

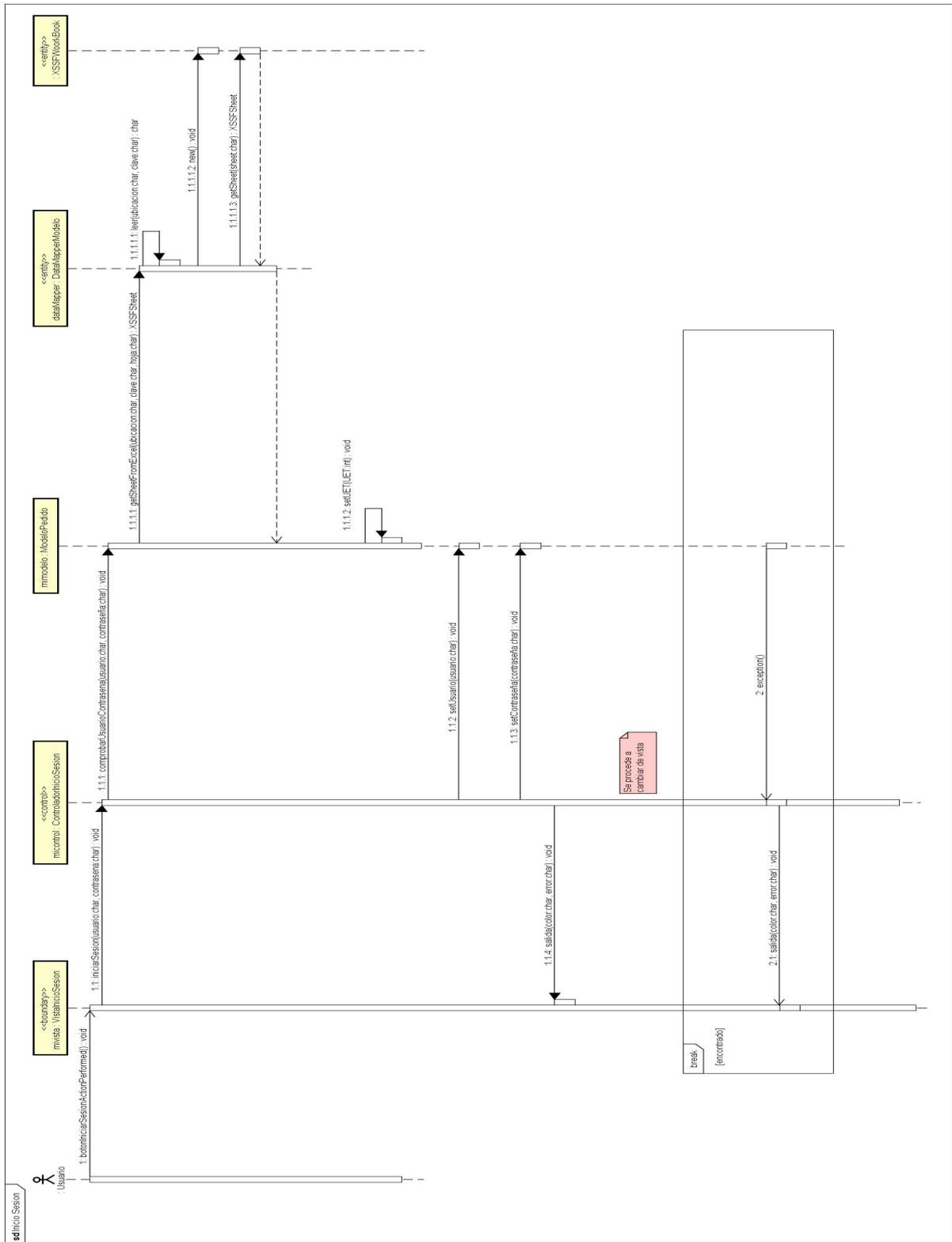
De igual manera que los modelos estáticos, éstos presentan perfiles similares pero su contenido es diferente debido a que la funcionalidad tampoco es la misma. Por lo tanto, también se presentan por separado.

5.3.3.1. Aplicación Formulario

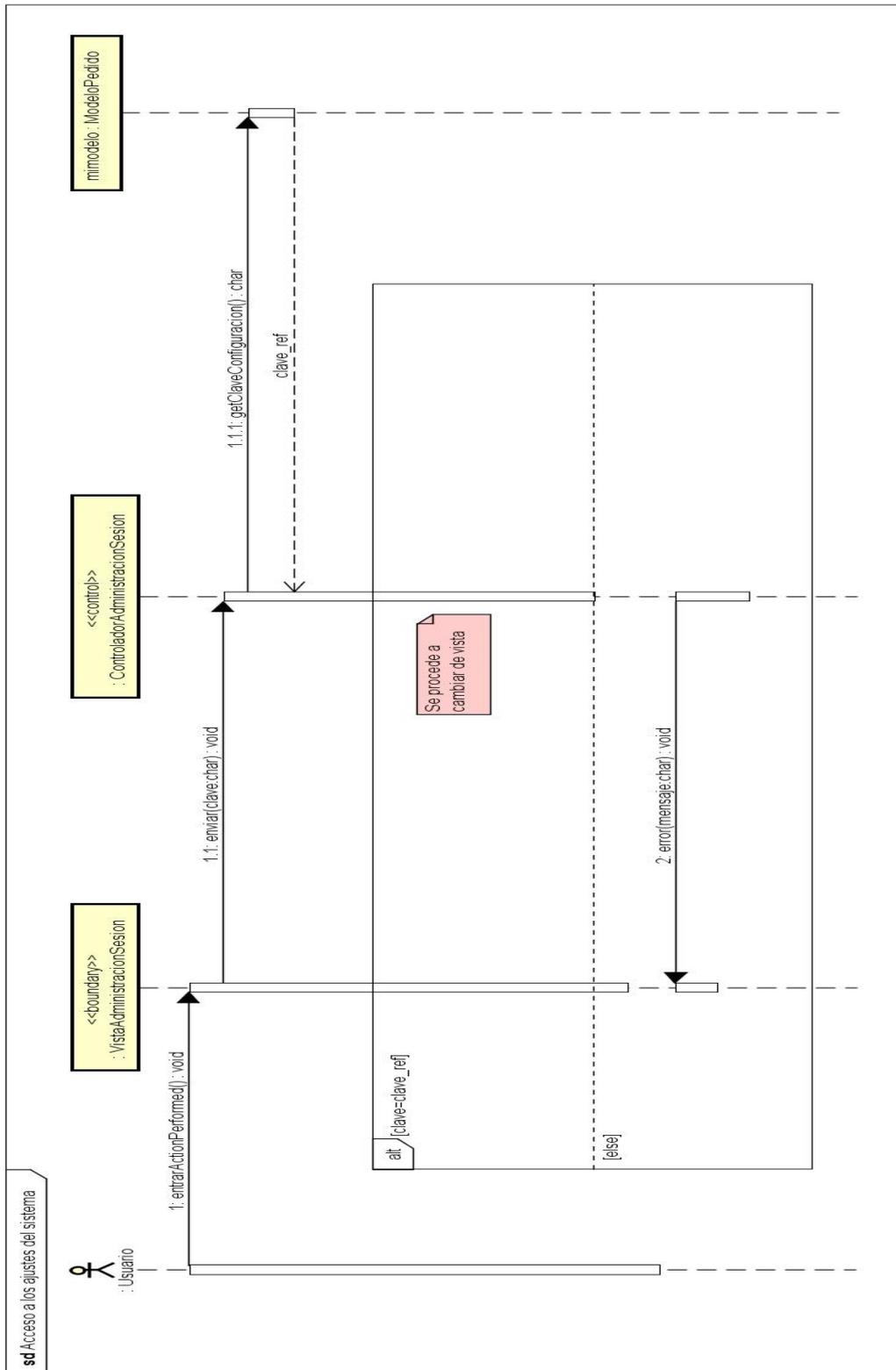
5.3.3.1.1. Diagramas de secuencia

En este apartado se presentan, desde el punto de vista dinámico, la acciones que realizan los usuarios y como responde internamente la aplicación para dar respuesta. Estas acciones de los usuarios son los principales casos de uso del sistema.

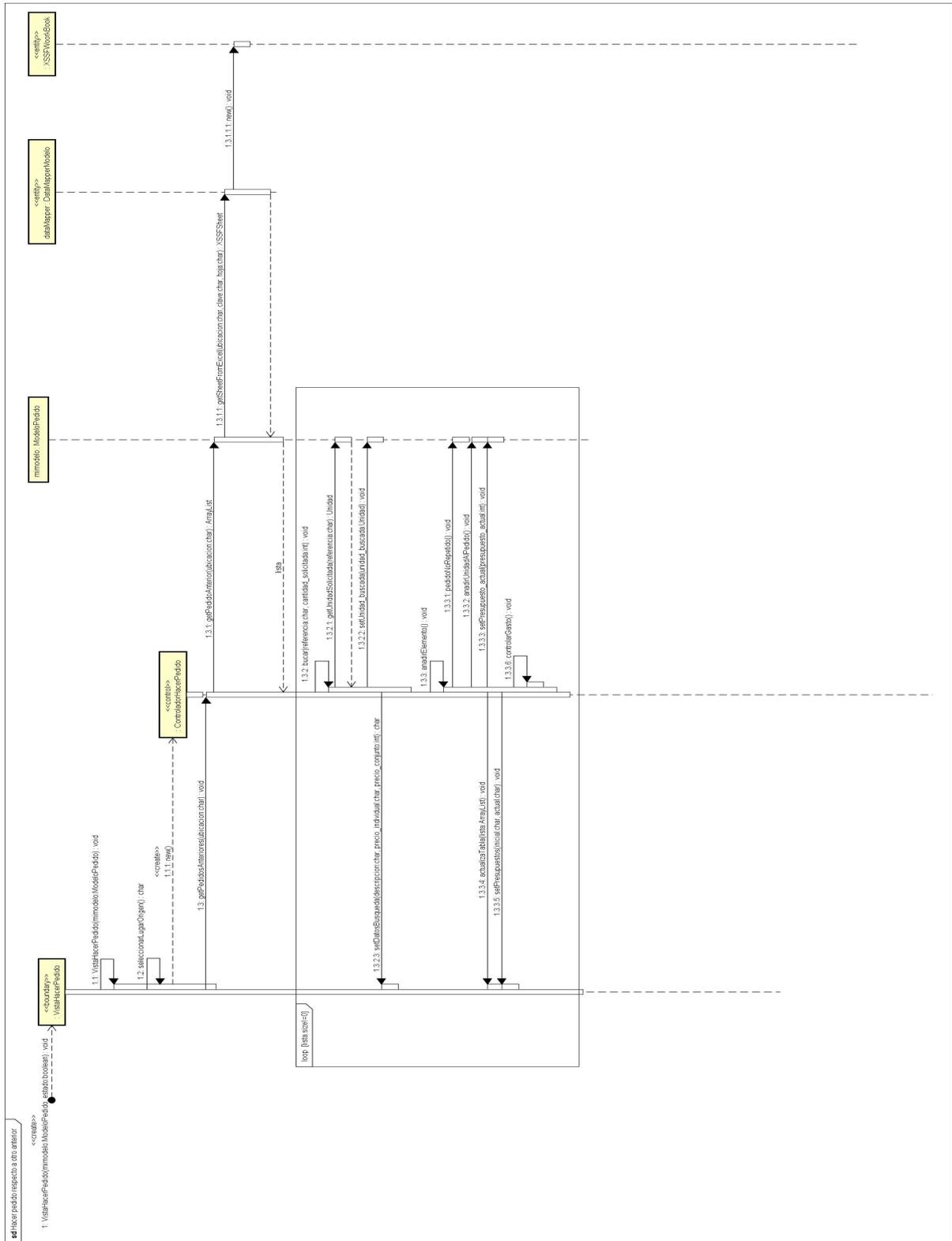
5.3.3.1.1.1. Iniciar sesión



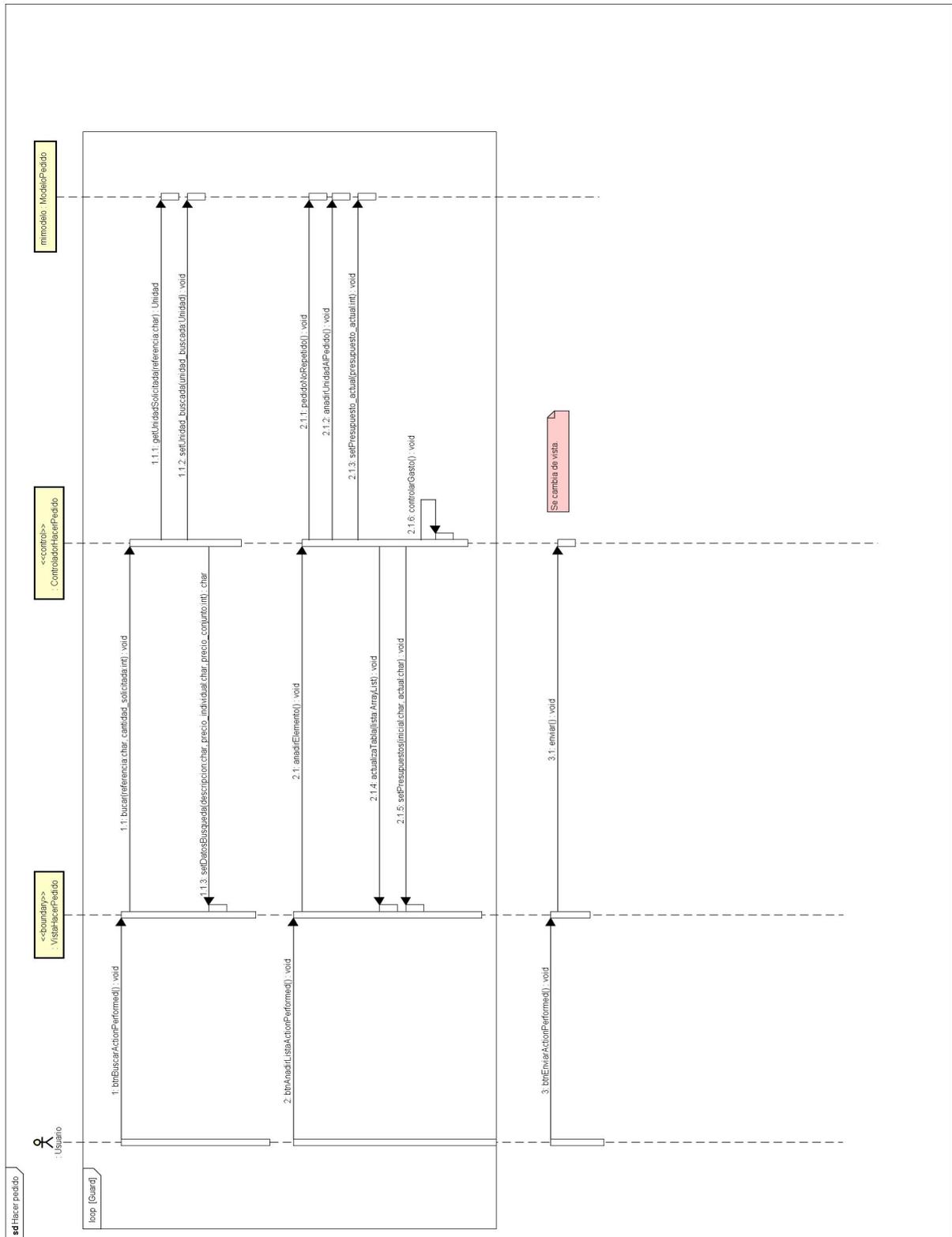
5.3.3.1.1.2. Acceder a los ajustes del sistema



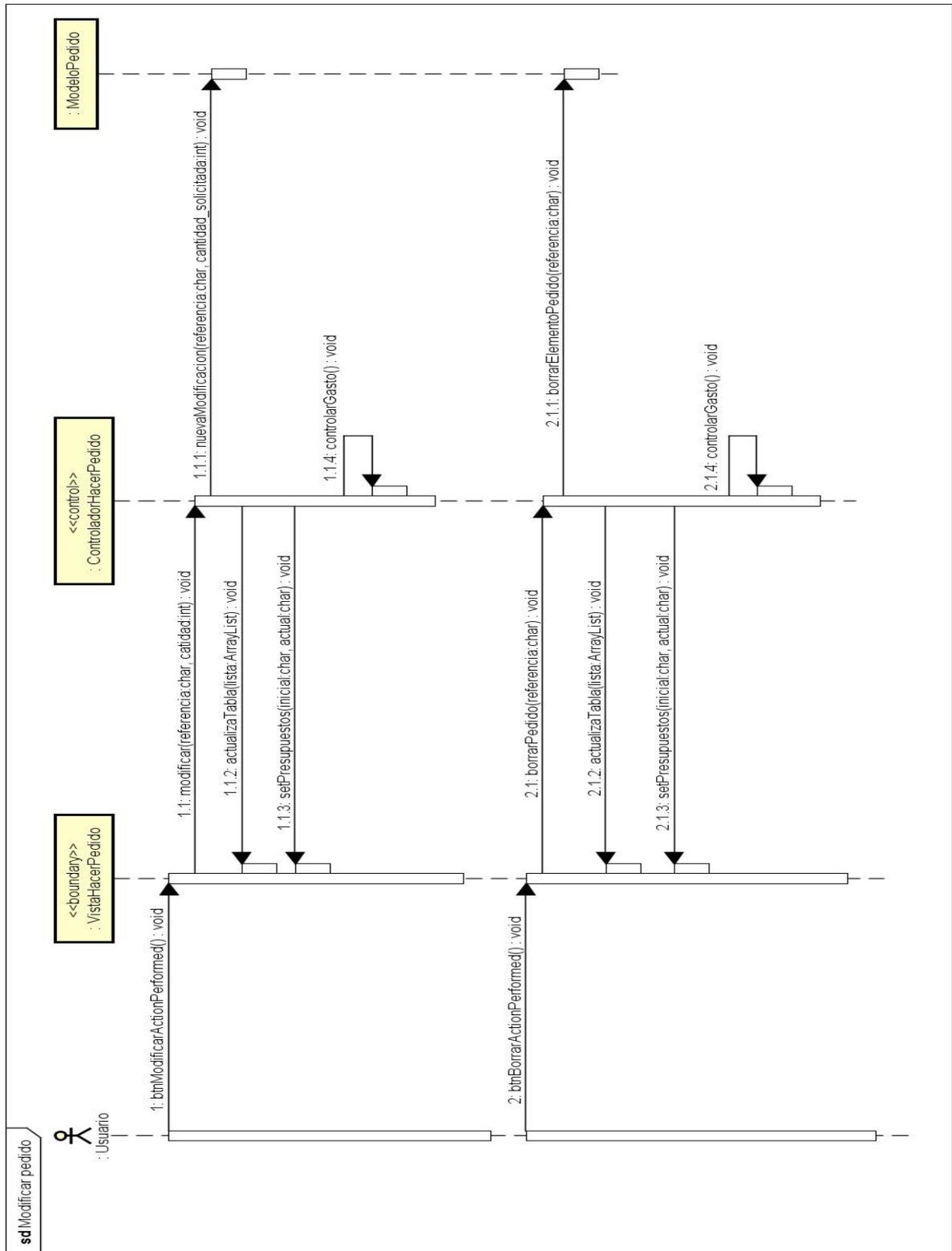
5.3.3.1.1.4. Hacer el pedido respecto a otro anterior



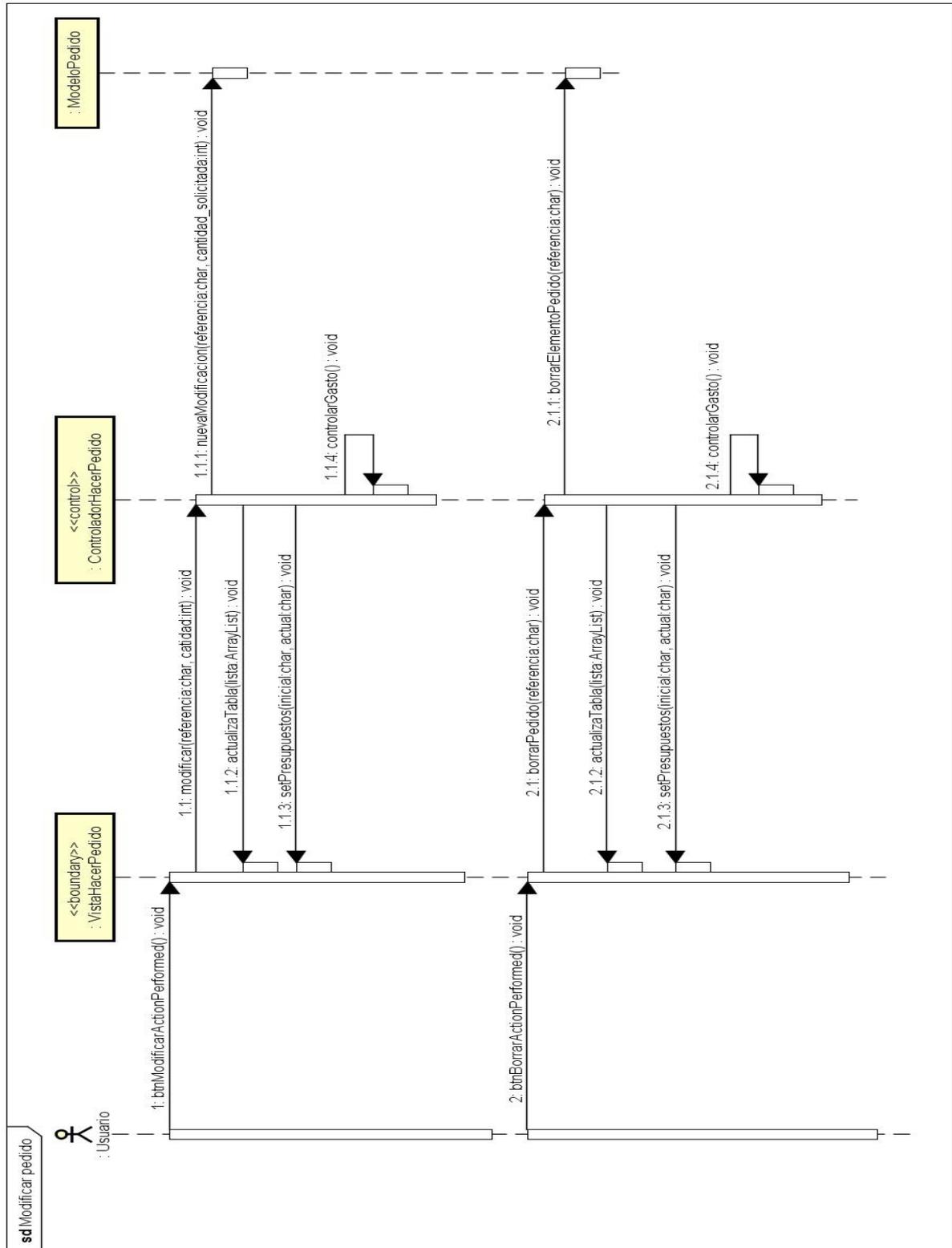
5.3.3.1.1.5. Hacer pedido



5.3.3.1.1.6. Modificar pedido



5.3.3.1.1.7. Ver referencias disponibles



5.3.3.2. Aplicación Envío

5.3.3.2.1. Diagramas de secuencia

5.3.3.2.1.1. Iniciar sesión

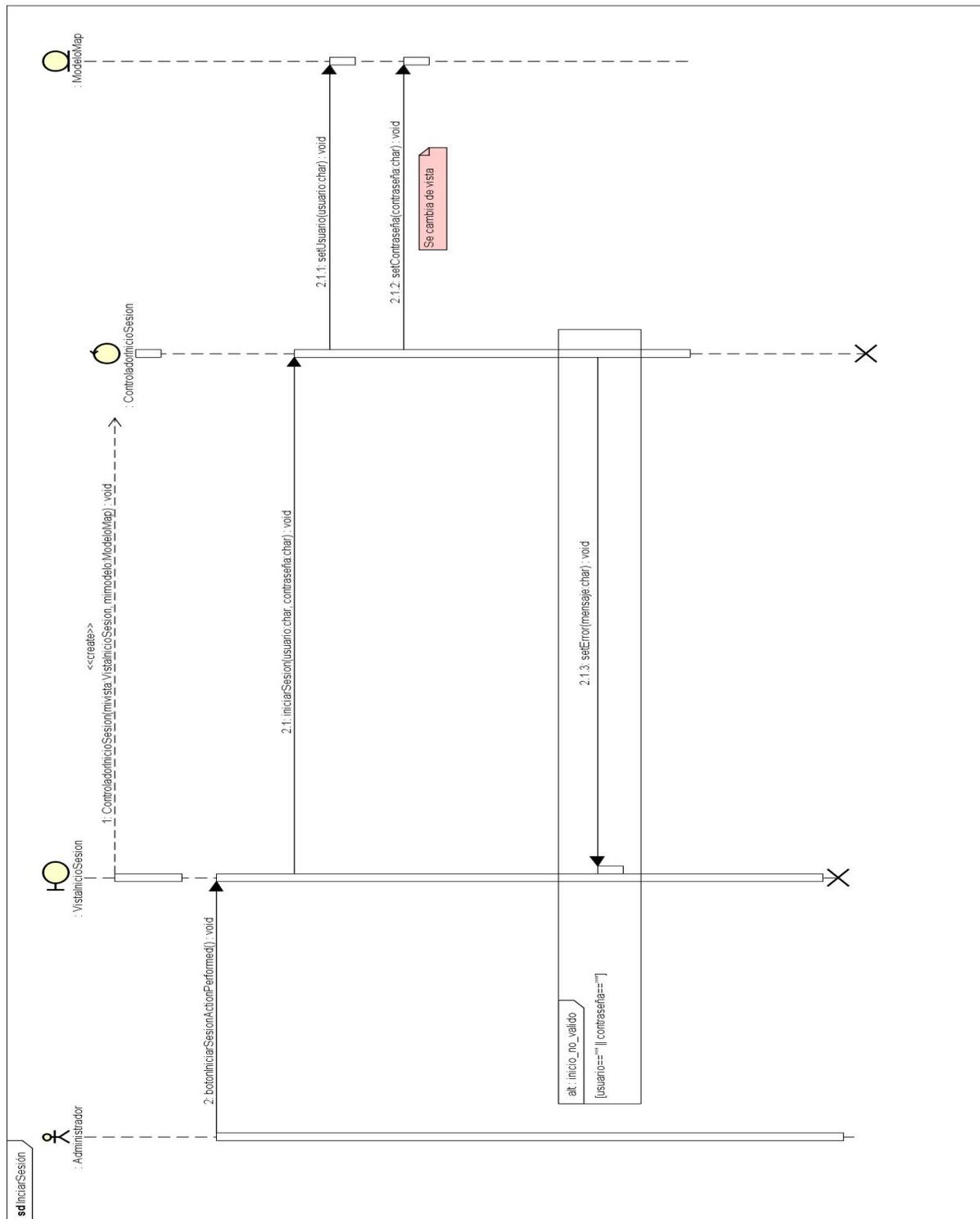


Figura 5.28: Clase HiloFinDePedido

5.3.3.2.1.2. Leer fichero Excel

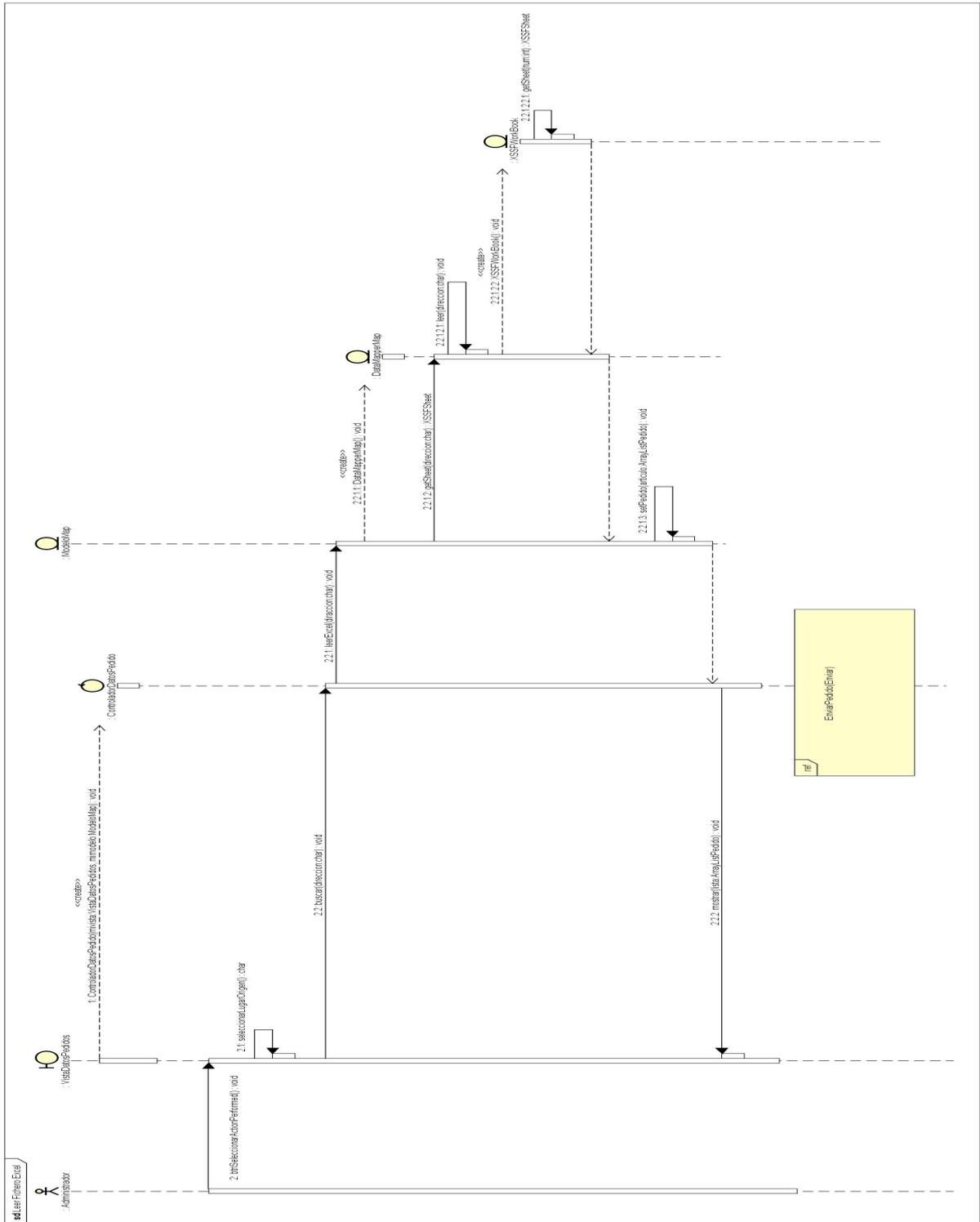


Figura 5.29: Clase HiloFinDePedido

5.4. Diseño de Interfaz de Usuario

A continuación, se presentan unos bocetos simples del esquema general que debía seguir la interfaz de usuario.

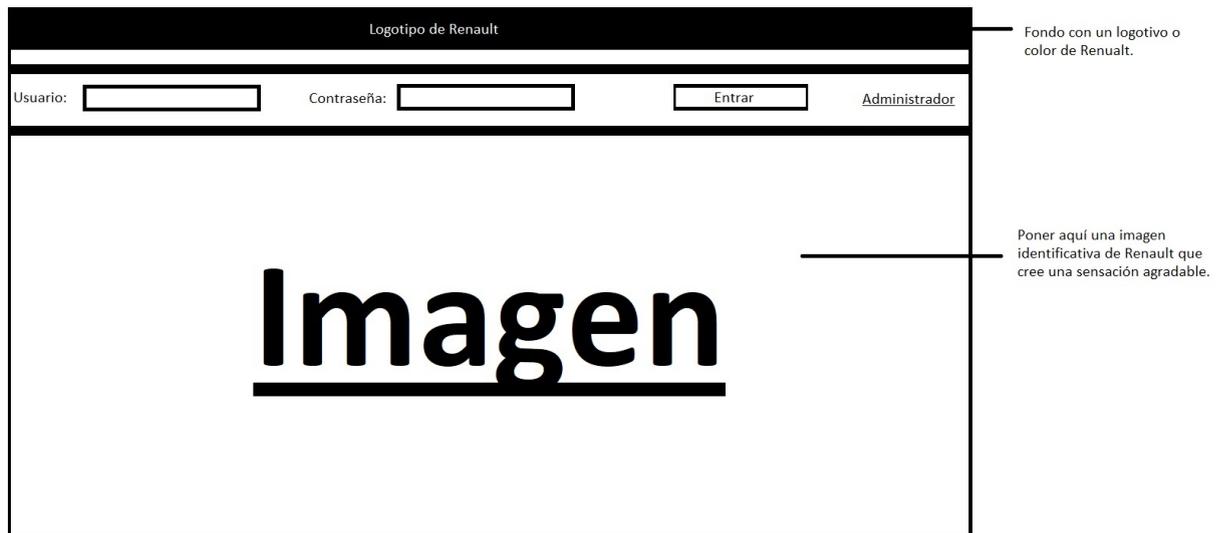


Figura 5.31: Boceto de la pantalla de inicio de sesión

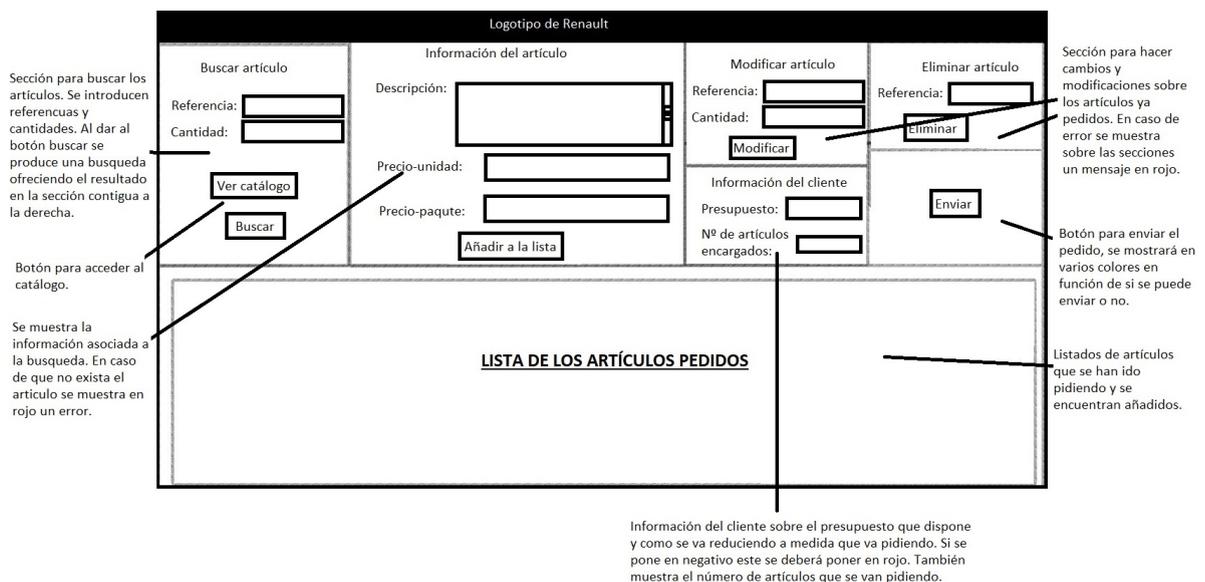


Figura 5.32: Boceto de la pantalla de añadir los artículos

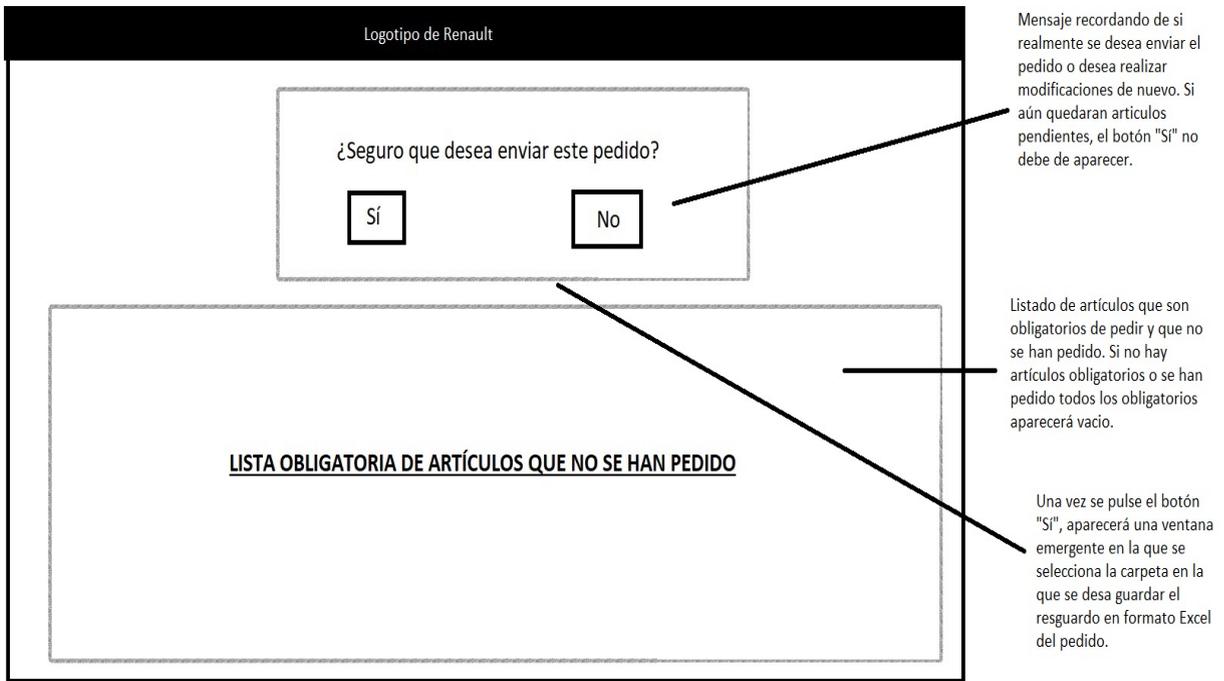


Figura 5.33: Boceto de la pantalla de confirmar el pedido

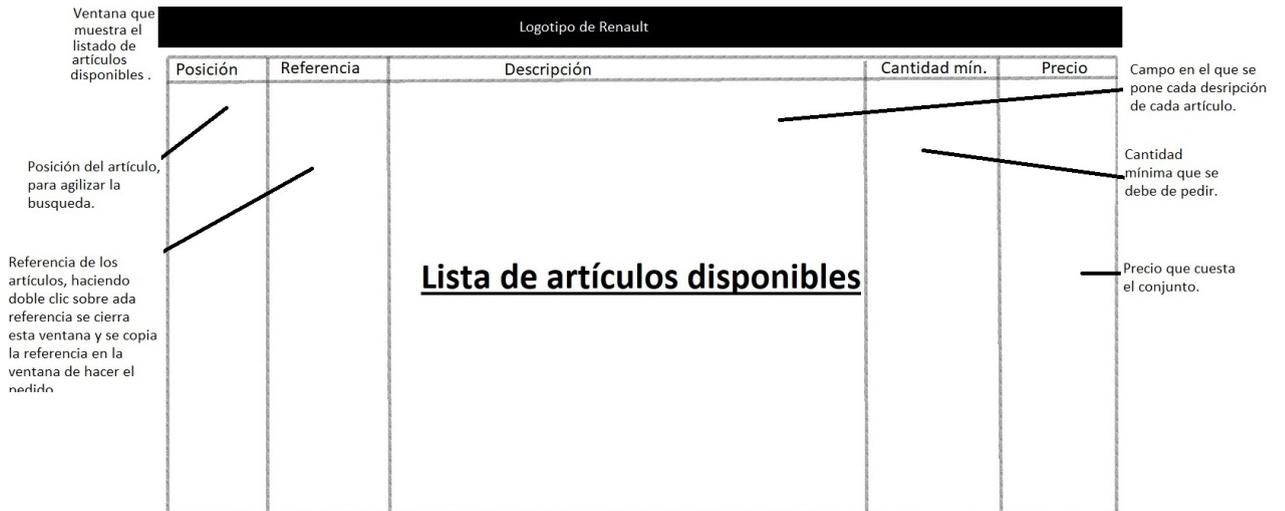


Figura 5.34: Boceto de la pantalla del catálogo

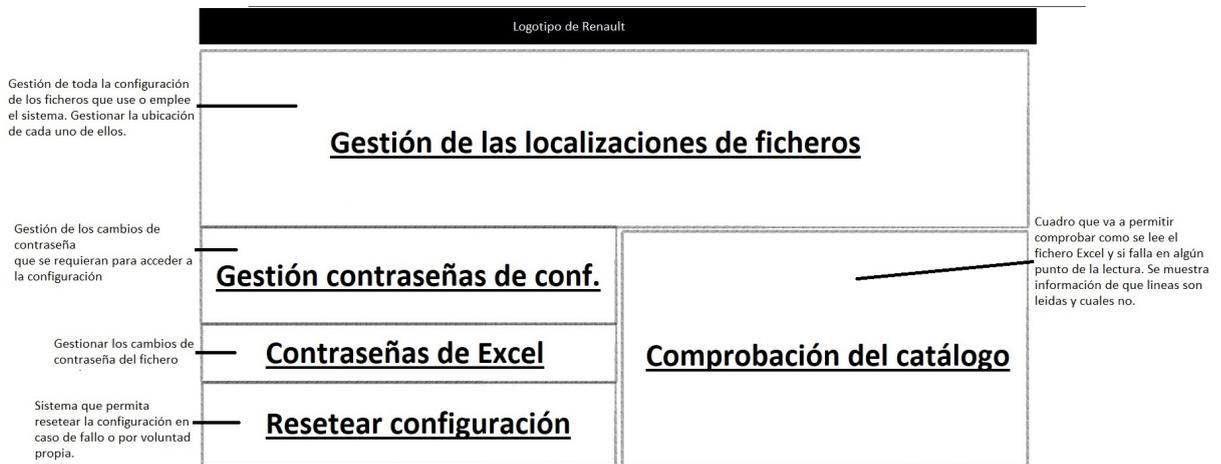


Figura 5.35: Boceto de la pantalla de configuración



Figura 5.36: Boceto de la pantalla de enviar al servidor

Una vez hemos visto cual es la estructura y organización que debe de tener nuestro proyecto, toca hablar de con que herramientas se va a llevar a cabo. Algunas de ellas ya han sido nombradas, debido a la necesidad para poder explicar algunos puntos. En las siguientes secciones se describe cuales son estas herramientas y el motivo que nos ha llevado a usar esas y no otras.

6.1. Java

Java es un lenguaje de programación de alto nivel orientado a objetos. Parte de la sintaxis de Java es tomada de otros lenguajes como C, Cobol o Visual Basic. Java reduce, respecto a los lenguajes mencionados antes, la dificultad en diferentes aspectos a bajo nivel.

Un programa escrito en Java podrá ejecutarse en cualquier sistema que cuente una Máquina Virtual Java (JVM). Aprovechamos esta característica de Java, así la aplicación que desarrollemos puede ser ejecutada en varios equipos de la factoría, independientemente del sistema operativo o de la arquitectura.

AWT o Swing son varias de las bibliotecas gráficas que posee Java. El diseño de interfaces gráficas en Java empleando estas bibliotecas y con ayuda de un entorno de desarrollo integrado (IDE) adecuado resulta bastante sencillo [6].

¿Por qué hemos usado Java? El uso de Java ha sido principalmente y únicamente por que estaba ya instalado por defecto en todos los ordenadores. Poder programarlo en otro lenguaje hubiera sido algo muy difícil de conseguir debido a la seguridad impuesta. Otro valor añadido es el alto conocimiento sobre este lenguaje. Reduciendo los tiempos de aprendizaje a casi cero y pudiendo empezar la labor de programación directamente.

6.2. Microsoft Office Excel

Microsoft Office es un paquete de programas informáticos para oficina desarrollado por Microsoft Corp. Se trata de un conjunto de aplicaciones que realizan tareas ofimáticas, es decir, que permiten automatizar y perfeccionar las actividades habituales de una oficina. Un contenido del paquete Office es Excel. Excel está compuesto por plantillas y hojas de cálculo. Su principal atractivo es la posibilidad de realizar operaciones aritméticas de manera automática, lo que facilita el desarrollo de balances y estados contables.

La decisión de usar esta herramienta fue, en parte, por la seguridad del sistema informático que impedía el uso de cualquier otro software que hiciera la función de base datos, y por otra parte debido al alto conocimiento y uso de esta herramienta en la oficina y que usaban continuamente. La idea de que usaran otra herramienta para ciertos aspectos era una propuesta que fue rechazada, por lo que, se aprovechó el uso que hacían de Excel para usarlo en beneficio del proyecto.

6.3. Apache POI

Apache POI es el resultado de un proyecto cuyo objetivo es crear y mantener una API Java para manipular varios formatos de archivo basados en los estándares Office Open XML (OOXML) y el formato OLE 2 Compound Document (OLE2) de Microsoft. En resumen, puede leer y escribir archivos de Microsoft Office usando Java. También se ofrece soporte para el cifrado de ficheros [13].

HSSF es la implementación Java del proyecto POI del formato de archivo Excel 97-07. **XSSF** es la implementación de Java del Proyecto POI del formato de archivo OOXML (.xlsx) de Excel 2007. HSSF y XSSF proporciona formas de leer hojas de cálculo crear, modificar, leer y escribir hojas de cálculo XLS. Proveen:

- Estructuras de bajo nivel para personas con necesidades especiales.
- Una api de modelo de evento para un acceso eficiente de solo lectura.
- Una api de modelo de usuario completa para crear, leer y modificar archivos XLS.

Hemos empleado esta herramienta por que es una API sencilla y fácil de entender. Además de permitir un amplio abanico de posibilidades de interacciones con los ficheros Excel.

6.4. Windows PowerShell

Windows PowerShell es un lenguaje de scripting y shell de línea de comandos basado en tareas diseñado especialmente para la administración del sistema. Windows PowerShell fue diseñado para usuario avanzados y profesionales para mejorar el control y la automatización tanto de tareas como de aplicaciones ejecutadas en Windows. Windows PowerShell incluye un editor gráfico.

Microsoft, con esta herramienta, trata de ofrecer una version Windows del shell de comando de Linux. Existen ciertas similitudes y comandos identicos que funcionan de la misma manera. No podemos hacer un manual sobre como funciona Windows PowerShell pero en la bibliografía se muestra un enlace a uno [14].

Hemos empleado esta herramienta ya que era la única que nos permitía, con las restricciones existentes, alcanzar nuestros objetivos. No se encontró ninguna alternativa a no hacerlo de esta manera.

6.4.1. Invoke-WebRequest

El comando Invoke Web-Request permite interactuar con una página web. En un entorno Linux, su homólogo sería el comando cUrl. Vamos a ver cual es la capacidad que nos ofrece. Ejecutamos el comando siguiente:

```
Invoke-WebRequest -Uri inf.uva.es
```

Como se puede apreciar en la figura 4, vemos la información que recibimos de ejecutar el comando.

```

PS C:\Users\Santos> Invoke-WebRequest -Uri http://inf.uva.es

StatusCode      : 200
StatusDescription : OK
Content         : <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
                  xmlns="http://www.w3.org/1999/xhtml" lang="es" dir="ltr"><head><meta http...
RawContent      : HTTP/1.1 200 OK
                  Strict-Transport-Security: max-age=15768000,max-age=31536000
                  X-Frame-Options: SAMEORIGIN
                  Upgrade: h2,h2c
                  Connection: keep-alive, Keep-Alive
                  Pragma: public
                  Vary: Accept-Encoding
Forms           : {}
Headers         : [{"Strict-Transport-Security, max-age=15768000,max-age=31536000"}, [{"X-Frame-Options, SAMEORIGIN"}, [{"Upgrade, h2,h2c"}, [{"Connection, keep-alive, Keep-Alive"}]...
Images         : [{"innerHTML=""; innerText=""; outerHTML="<IMG class=logo alt='Escuela de Ingeniería Informática de
                  Valladolid' src='https://www.inf.uva.es/wp-content/ata-images/logo.png'; outerText="; tagName=IMG;
                  class=logo; alt=Escuela de Ingeniería Informática de Valladolid;
                  src=https://www.inf.uva.es/wp-content/ata-images/logo.png"; @{"innerHTML="; innerText=";
                  outerHTML="<IMG title=English alt=English
                  src='/wp-content/plugins/transposh-translation-filter-for-wordpress/img/flags/us.png';
                  outerText="; tagName=IMG; title=English; alt=English;
                  src=/wp-content/plugins/transposh-translation-filter-for-wordpress/img/flags/us.png";
                  @{"innerHTML="; innerText="; outerHTML="<IMG title=Español alt=Español
                  src='/wp-content/plugins/transposh-translation-filter-for-wordpress/img/flags/es.png';
                  outerText="; tagName=IMG; title=Español; alt=Español;
                  src=/wp-content/plugins/transposh-translation-filter-for-wordpress/img/flags/es.png";
                  @{"innerHTML="; innerText="; outerHTML="<IMG title='Charla Indra Minsait"
                  class='attachment-post-thumbnail size-post-thumbnail wp-post-image' alt=""
                  src='https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-107x150.jpg"
                  width=107 height=150 sizes='(max-width: 107px) 100vw, 107px"
                  srcset="https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-107x150.jpg 107w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-213x300.jpg 213w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-546x768.jpg 546w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-728x1024.jpg 728w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-400x563.jpg 400w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-900x1266.jpg 900w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-57x80.jpg 57w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-85x120.jpg 85w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-40x56.jpg 40w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-178x250.jpg 178w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait.jpg 1149w"; outerText=";
                  tagName=IMG; title=Charla Indra Minsait; class=attachment-post-thumbnail size-post-thumbnail
                  wp-post-image; alt=";
                  src=https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-107x150.jpg;
                  width=107; height=150; sizes=(max-width: 107px) 100vw, 107px;
                  srcset=https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-107x150.jpg 107w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-213x300.jpg 213w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-546x768.jpg 546w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-728x1024.jpg 728w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-400x563.jpg 400w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-900x1266.jpg 900w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-57x80.jpg 57w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-85x120.jpg 85w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-40x56.jpg 40w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait-178x250.jpg 178w,
                  https://www.inf.uva.es/wp-content/uploads/2017/10/20171122_IndraMinsait.jpg 1149w)..."}
InputFields    : [{"innerHTML=""; innerText=""; outerHTML="<INPUT onfocus='this.value=''' onblur='this.value='Búsqueda"
                  class='text inputblur' value=Búsqueda name=s; outerText="; tagName=INPUT; onfocus=this.value='';
                  onblur=this.value= Búsqueda; class=text inputblur; value=Búsqueda; name=s"}
Links          : [{"innerHTML="<SPAN>Clases</SPAN>; innerText=Clases; outerHTML="<A
                  href='https://www.inf.uva.es/calendario-de-clases-2017-2018/"
                  data-slimstat="5"><SPAN>Clases</SPAN></A>; outerText=Clases; tagName=A;
                  href=https://www.inf.uva.es/calendario-de-clases-2017-2018/; data-slimstat=5},
                  [{"innerHTML="<SPAN>Exámenes</SPAN>; innerText=Exámenes; outerHTML="<A
                  href='https://www.inf.uva.es/calendario-de-examenes-2017-2018/"
                  data-slimstat="5"><SPAN>Exámenes</SPAN></A>; outerText=Exámenes; tagName=A;
                  href=https://www.inf.uva.es/calendario-de-examenes-2017-2018/; data-slimstat=5},
                  [{"innerHTML="<SPAN>Labs</SPAN>; innerText=Labs; outerHTML="<A title='Acceso Recursos Informáticos de
                  la Escuela"
                  href='https://www.inf.uva.es/instrucciones-para-el-acceso-a-los-recursos-informaticos/"
                  data-slimstat="5"><SPAN>Labs</SPAN></A>; outerText=Labs; tagName=A; title=Acceso Recursos
                  Informáticos de la Escuela;
                  href=https://www.inf.uva.es/instrucciones-para-el-acceso-a-los-recursos-informaticos/;
                  data-slimstat=5}, [{"innerHTML="<SPAN>Clave</SPAN>; innerText=Clave; outerHTML="<A title='Obtener la
                  clave de laboratorios" href='https://www.inf.uva.es/recuperar-clave/"
                  data-slimstat="5"><SPAN>Clave</SPAN></A>; outerText=Clave; tagName=A; title=Obtener la clave de
                  laboratorios; href=https://www.inf.uva.es/recuperar-clave/; data-slimstat=5}...}
ParsedHtml     : mshtml.HTMLDocumentClass
RawContentLength : 155049

```

Figura 6.1: Herramienta Invoke WebRequest

Al usar Invoke-WebRequest , el contenido de la solicitud se nos devuelve en un objeto. Hay muchas maneras de revisar los datos. Disponemos de las siguientes posibilidades:

- RawContent: Este es el contenido devuelto tal como se recibió, incluidos los datos del encabezado.
- Formularios: Esta propiedad contiene cualquier formulario que contenga la página web.
- Headers(cabeceras): Esta propiedad contiene solo la información del encabezado devuelto.
- Imágenes: Esta propiedad contiene todas las imágenes que pudieron ser descubiertas.
- InputFields: Esta propiedad devuelve los campos de entrada descubiertos en el sitio web.
- Links: Esto devuelve los enlaces encontrados en el sitio web, en un formato fácil de recorrer.
- ParsedHTML: Esta propiedad le permite acceder al DOM del sitio web. DOM es la abreviatura de Document Object Model. Piense en DOM como una representación estructurada de los datos en el sitio web.

Invoke WebRequest incluye la siguiente funcionalidad:

- Verificando si un sitio web está activo.
- Descargando archivos.
- Iniciando sesión en páginas web.
- Analizando el contenido que recibimos.
- Utilizando métodos REST.

Invoke-WebRequest también puede funcionar con datos de formulario. Permite la posibilidad de recibir los datos actuales del formulario, actualizar los valores y enviarlos. Para poder verlos hay que escribir “webRequest.Forms”. En la figura 5 se muestra un ejemplo. En él vemos cómo solicitamos una página web, recibimos el contenido y se analiza el contenido de los formularios que contiene. Como solo contiene un formulario, vemos cuales son los campos que contiene.

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Santos> $formulario = Invoke-WebRequest 'aulas.inf.uva.es/login/index.php'
PS C:\Users\Santos> $formulario.Forms

Id      Method Action                                     Fields
--      -
login  post   https://aulas.inf.uva.es/login/index.php {[anchor, ], [username, ], [password, ], [rememberusername, 1]}

PS C:\Users\Santos> $aulas=$formulario.Forms[0]
PS C:\Users\Santos> $aulas

Id      Method Action                                     Fields
--      -
login  post   https://aulas.inf.uva.es/login/index.php {[anchor, ], [username, ], [password, ], [rememberusername, 1]}

PS C:\Users\Santos> $aulas.Fields

Key      Value
---      -
anchor
username
password
rememberusername 1
```

Figura 6.2: Formulario recibido con Invoke WebRequest

Estos campos se pueden rellenar y enviárselos al servidor. Recibiendo de este la página web resultante de iniciar sesión. La funcionalidad que ofrece esta herramienta es muy amplia y variada, por lo que no se puede resumir en este documento toda su funcionalidad [15]. Simplemente, se ha hecho una breve introducción a ella y mostrado un poco sus resultados.

7.1. Introducción

A continuación se presenta cuál es la estructura de ficheros que se crean en la implementación de ambas aplicaciones, igualmente se enseñarán la formación del fichero de información de Excel.

Como se ha mencionado en el capítulo anterior, el lenguaje empleado para la implementación ha sido Java. Para el desarrollo del código se empleó el IDE NetBeans en un ordenador personal, ya que en los ordenadores de Renault no se permiten instalar programas.

7.2. Estructura de Directorios

En la ilustración que se muestra a continuación, podemos ver cuál es la estructura de directorios establecida.



Figura 7.1: Estructura de directorios de la aplicación de pedidos del JU

En la siguiente imagen, podemos ver qué clases han sido introducidas en las carpetas.

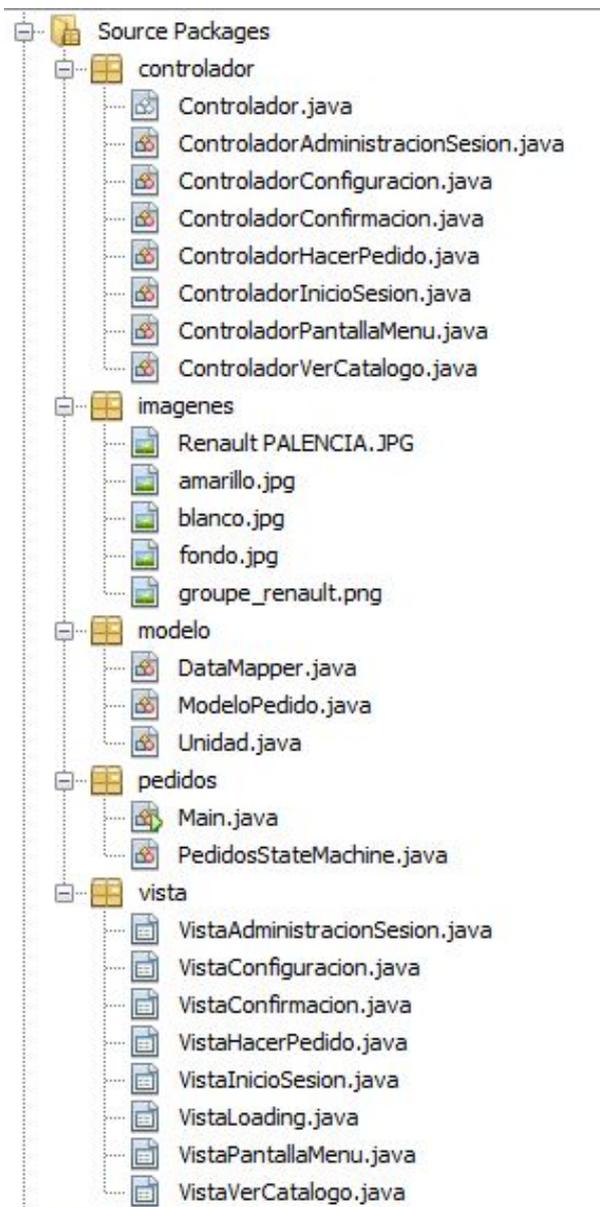
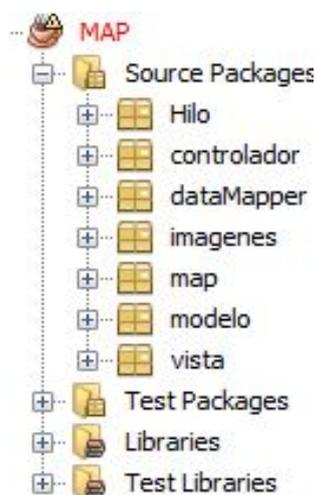


Figura 7.2: Estructura de directorios ampliada



114
Figura 7.3: Estructura de directorios de la aplicación de envío de pedidos al centro de pedidos

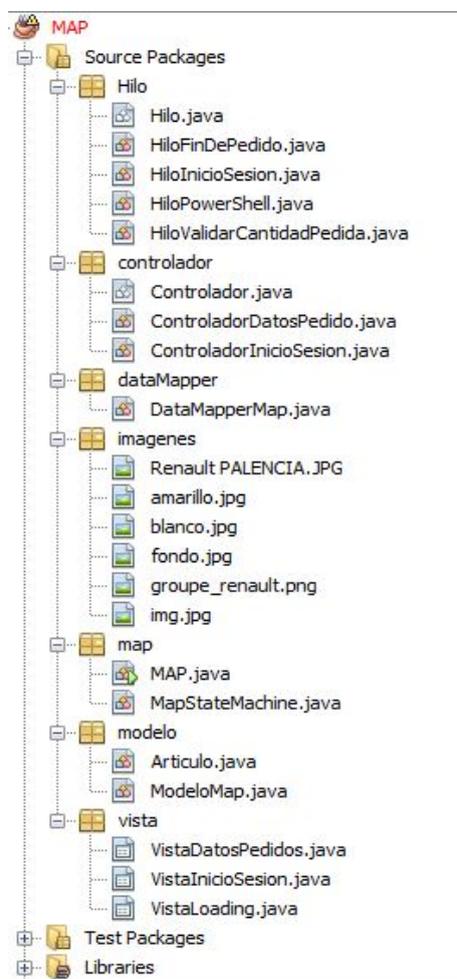


Figura 7.4: Estructura de directorios ampliada

7.3. Seguridad en las Contraseñas

En este apartado haremos mención importante al tratamiento de las contraseñas de las cuentas de usuario. Por nuestra parte, solo la aplicación formulario requiere de un tratamiento, la aplicación envío no entra dentro de nuestra responsabilidad.

En toda aplicación informática que trate con cuentas de usuario, las contraseñas no deben usarse en plano por motivos obvios de seguridad. Éstas suelen tratarse mediante cifrado con claves o valores hash (función de resumen). Nuestra aplicación trabaja con función hash mediante el sistema SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro). Dentro de las diferentes versiones de este algoritmo, concretamente empleamos un cifrado SHA-1.

Esta función podemos dividirla en dos partes, una parte relacionada con el almacenamiento y otra parte relacionado con la comprobación.

- Almacenamiento: en ningún momento se deben guardar las contraseñas en texto plano. Éstas son sustituidas mediante su valor hash. El coordinador, al insertar un usuario nuevo, introduce la contraseña en texto plano en el fichero Excel. Seguidamente, pulsa un botón creado específicamente para ello, y mediante una macro, transforma la contraseña en su valor hash correspondiente y lo almacena. Posteriormente elimina el valor plano de la contraseña. Para

que un usuario pueda recuperar su contraseña, deberá hablar con el coordinador y determinar una nueva contraseña.

- **Comprobación:** cuando un usuario desea iniciar sesión deberá introducir su usuario y contraseña. Las contraseñas siempre deben cifrarse a la hora de trabajar con ellas. En nuestro caso, se calcula el valor hash de la contraseña mediante SHA. Posteriormente, se procede a compararla con el valor almacenado dentro del fichero Excel.

Mediante estas técnicas realizamos un correcto tratamiento de las contraseñas cumpliendo con una política segura.

7.4. Estructura del Fichero Excel

A continuación se pronunciará sobre la estructura del fichero Excel. Dicho fichero aporta información a la aplicación por la que el JU realiza sus pedidos, es decir, la aplicación formulario. Se realizan estos comentarios debido a la importancia funcional que aporta a la aplicación. En la aplicación envío no existe la necesidad de disponer de un fichero que le provea de información.

El fichero Excel está formado por varias páginas, las cuales recogen información que permiten el correcto funcionamiento del programa. Este fichero se encuentra creado previamente a la aplicación, independientemente de su ejecución. El archivo Excel es usado por la aplicación como fuente de información. El formato es fijo, no debe variarse para permitir un correcto funcionamiento de la aplicación. En caso de variación, pueden presentarse errores en la obtención de datos, provocando un funcionamiento incorrecto de la aplicación. A continuación se muestra cuáles son esas hojas del fichero Excel y como son interpretadas por el software desarrollado. El contenido de las siguientes imágenes ha variado para no mostrar información comprometida de los productos empleados por el departamento.

7.4.1. Usuarios

	A	B	C	D	E	F	G
1	USUARIOS	CONTRASEÑAS	UET	HASH	<div style="border: 1px solid white; padding: 10px; width: fit-content; margin: auto;"> Crear hash </div>		
2	usuario1	-	1	f0578f1e7174b1a41c4ea8c6e17f7a8a3b88c92a			
3	usuario2	-	2	8be52126a6fde450a7162a3651d589bb51e9579d			
4	usuario3	-	3	de2a4d5751ab06dc4f987142db57c26d50925c8a			
5	usuario4	-	4	2db4c1811f424582a90f8d7ee77995cf018d9443			
6	usuario5	-	5	9e5ca6b0ffb417997ffb844c76f9c24bbc20fe88			
7	usuario6	-	6	b8de6df1561cd7ab6af36d0cb706b2168b4f4c69			
8	usuario7	-	7	8a337f7233833c6cd96370d64adb55f49b30db2			
9	usuario8	-	8	2c186a6f0dda5aa237254943899d4fee0ab15dd0			
10	usuario9	-	9	e74e5641aa1bea180e9a50cf7f269e6f0e28e341			
11	usuario10	-	10	17005c1048dda67c3f735014f33625dfd0c89d32			
12	usuario11	-	11	0808f4f387948ca6e398f4030e8ee19d74c5883e			
13	usuario12	-	12	4e05d4fa6439a3daf2b853e3df1858d42e861df1			
14	usuario13	-	13	e23e98b3ba8e88821c3d15d8584656069e220db8			
15	usuario14	-	14	55e28fa0e928e7427b2d45301b1953655ec8587d			
16							
17							
18							

Figura 7.5: Hoja usuarios

Existen 4 columnas, asociadas a los nombres de usuarios, a la contraseña, al código hash de la contraseña y a la UET. Si se desea realizar alguna modificación (añadir, eliminar o modificar algún usuario) debe de hacerse sobre esta hoja. Cuando se inserta un nuevo usuario se debe añadir el nombre, contraseña y UET correspondientes en la primera fila libre. El siguiente paso es pulsar el

botón situado a la derecha de la pantalla. Dicho botón obtiene el valor hash mediante el algoritmo SHA de la contraseña. Dicho valor es guardado en una cuarta columna mientras que el valor original es eliminado.

7.4.2. Fechas

	A	B
1	Fechas Validas	
2	Desde	Hasta
3		10
4		

Figura 7.6: Hoja fechas

En esta hoja se colocan los días en los que se puede hacer pedido de los artículos. Si se desea modificar el día de inicio o el día del final tan solo hay que cambiar el día.

7.4.3. Presupuesto

	A	B	C	D
1	Usuarios	Presupuesto	€/Vehículo	Nº Vehiculos previsto
2	usuario1	1368	0,0684	20000
3	usuario2	1368	0,0684	20000
4	usuario3	1368	0,0684	20000
5	usuario4	1368	0,0684	20000
6	usuario5	957,6	0,04788	20000
7	usuario6	3168	0,1584	20000
8	usuario7	3168	0,1584	20000
9	usuario8	3168	0,1584	20000
10	usuario9	2217,6	0,11088	20000
11	usuario10	1728	0,0864	20000
12	usuario11	1728	0,0864	20000
13	usuario12	1728	0,0864	20000
14	usuario13	1209,6	0,06048	20000

Figura 7.7: Hoja presupuestos

En esta hoja se asocia el nombre del usuario y su presupuesto. El software solo lee las dos primeras columnas, las columnas C y D son para el administrador, para que pueda calcular más fácilmente el presupuesto, las cuales se multiplican y ponen el resultado en B.

7.4.4. Referencias

	A	B	C	D	E	F
1	Zona	Denominación	Referencia	Proveedor	Precio	Pedido M
2	Mástico	Descripción del producto 1	R0000001	Proveedor A	0,79	100
3	Mástico	Descripción del producto 2	R0000002	Proveedor A	0,79	100
4	Mástico	Descripción del producto 3	R0000003	Proveedor A	0,79	100
5	Mástico	Descripción del producto 4	R0000004	Proveedor A	0,79	100
6	Mástico	Descripción del producto 5	R0000005	Proveedor A	1	100
7	Mástico	Descripción del producto 6	R0000006	Proveedor A	1	100
8	Mástico	Descripción del producto 7	R0000007	Proveedor A	1	100
9	Mástico	Descripción del producto 8	R0000008	Proveedor A	1	100
10	Mástico	Descripción del producto 9	R0000009	Proveedor A	1	100
11	Mástico	Descripción del producto 10	R0000010	Proveedor B	1	20
12	Mástico	Descripción del producto 11	R0000011	Proveedor B	1	20
13	Mástico	Descripción del producto 12	R0000012	Proveedor B	1	20
14	Mástico	Descripción del producto 13	R0000013	Proveedor B	115,35	4
15	Petoque	Descripción del producto 14	R0000014	Proveedor B	115,35	4
16	Petoque	Descripción del producto 15	R0000015	Proveedor B	115,35	4
17	Petoque	Descripción del producto 16	R0000016	Proveedor B	115,35	4
18	Petoque	Descripción del producto 17	R0000017	Proveedor B	115,35	4
19	Petoque	Descripción del producto 18	R0000018	Proveedor B	115,35	4
20	Petoque	Descripción del producto 19	R0000019	Proveedor B	115,35	3
21	Petoque	Descripción del producto 20	R0000020	Proveedor B	115,35	3
22	Petoque	Descripción del producto 21	R0000021	Proveedor A	115,35	3
23	Petoque	Descripción del producto 22	R0000022	Proveedor A	256,02	1
24	Petoque	Descripción del producto 23	R0000023	Proveedor A	256,02	1

Figura 7.8: Hoja referencias

En esta hoja podemos ver las referencias disponibles, existen varias columnas en las cuales se leen diferentes datos. En la columna A se puede poner la zona de aplicación dentro del departamento de Pintura, aunque no es una información utilizada en el programa, solo sirve para tener mejor ordenados los datos. En la columna B la denominación o descripción del artículo, en la columna C vemos la referencia asociada. La columna D contiene el proveedor, dato no necesario para el programa. La columna E contiene el precio, si no contiene precio, el programa interpreta como que ese artículo no está disponible, de igual manera sucede en la columna F, que contiene el pedido mínimo de cada artículo, si la cantidad es 0 se traduce como que el artículo no está disponible.

La selección de los colores fue puesta por el coordinador, para poder realizar una búsqueda visual más rápida.

7.4.5. Detalles

1	USUARIOS	REFERENCIAS QUE EL USUARIO DEBE PEDIR					
2	usuario1	R0000020					
3	usuario2	R0000021	R0000001	R0000002	R0000005	R0000004	R0000007
4	usuario3	R0000022	R0000001	R0000003	R0000006	R0000005	R0000008
5	usuario4	R0000023	R0000001	R0000004	R0000007	R0000006	R0000009
6	usuario5	R0000024	R0000001	R0000005	R0000008	R0000007	R0000010
7	usuario6	R0000025	R0000002				
8	usuario7	R0000026	R0000003				
9	usuario8	R0000027	R0000004				
10	usuario9	R0000028	R0000001				
11	usuario10	R0000029	R0000001				
12	usuario11	R0000030	R0000002				
13	usuario12	R0000031	R0000003				
14	usuario13	R0000032	R0000004				
15	usuario14	R0000032	R0000005				

Figura 7.9: Hoja detalles

En la hoja Detalles se encuentra una lista de artículos que son de pedido obligatorio para un usuario en concreto.

7.4.6. Ubicación

	A	B
1		
2		plantillaJU.xlsx
3		plantillaadministrador.xlsx
4		informacion.xlsm
5		C:\Users\Santos\Desktop\
6		
7		937bfaea6b875d17a48b0e4b499c346e56c4ca1c
8		

Figura 7.10: Hoja ubicación

En la hoja Ubicación encontramos una serie de datos usados por el programa para almacenar información acerca de ubicaciones, ficheros, estados o valores hash de las contraseñas.

7.4.7. Pedidos

	A	B
1		Gente que ha realizado pedidos:
2		usuario1-6-2017
3		usuario2-6-2017
4		usuario3-6-2017
5		usuario4-6-2017
6		usuario5-6-2017

Figura 7.11: Hoja pedidos

En la hoja de pedidos realizados se almacena una lista de usuarios que ya realizaron un pedido ese mes para así poder saber quiénes han repetido o quienes no un pedido.

CAPÍTULO 8

INSTALACIÓN Y PRUEBAS

8.1. Sistema Operativo

- SO: Windows 7, Windows 8 y Windows 10.
- Herramienta Microsoft: Microsoft Excel 2010.
- Lenguaje: Java 7-8
- Windows PowerShell 2016 Microsoft Corporation

8.2. Pruebas de Sistema

Robustez

- Desaparición del fichero de arranque.
- Desaparición del fichero de información.

8.3. Pruebas de Caja Negra

A continuación se realizarán las pruebas de caja negra, en ella se estudia cual es la respuesta del sistema ante la entrada de información textual y cual es la salida que genera, de forma ajena a cual sea el funcionamiento interno del sistema.

Inicio de sesión – Usuario y contraseña

- Usuario

Usuario		
Motivo	Entrada válida	Entrada no válida
Número de caracteres	A N>0	B N<1

Tabla 8.1: Inicio de sesión: campo usuario

- Contraseña

Ejemplo Usuario		
Ejemplo	Clase	Resultado
Santos	A	Correcto
	B	Incorrecto

Tabla 8.2: Prueba usuario

Contraseña		
Motivo	Entrada válida	Entrada no válida
Número de caracteres	A $N > 0$	B $N < 1$

Tabla 8.3: Inicio de sesión: campo contraseña

Ejemplo Contraseña		
Ejemplo	Clase	Resultado
abcd1234	A	Correcto
	B	Incorrecto

Tabla 8.4: Prueba contraseña

Hacer pedido – Referencia y cantidad

- Referencia

Referencia		
Motivo	Entrada válida	Entrada no válida
Número de caracteres	A $N > 0$	B $N < 1$

Tabla 8.5: Hacer pedido: campo referencia

Ejemplo Referencia		
Ejemplo	Clase	Resultado
R100640408	A	Correcto
	B	Incorrecto

Tabla 8.6: Prueba referencia

- Cantidad

Cantidad		
Motivo	Entrada válida	Entrada no válida
Número de caracteres	A $N > 0$	B $N < 1$
Contenido	C Sin letras	D Con letras

Tabla 8.7: Hacer pedido: campo cantidad

Ejemplo Cantidad		
Ejemplo	Clase	Resultado
24	A	Correcto
	B	Incorrecto
12	C	Correcto
As12	D	Incorrecto

Tabla 8.8: Prueba cantidad

9.1. Objetivos Alcanzados

Al comienzo del presente documento, se analizaban una serie de objetivos que se trataban de alcanzar al finalizar el mismo. Repasaremos a continuación esos objetivos y en que grado se han cumplido:

- Respecto a la funcionalidad del software sí se ha conseguido alcanzar un producto que satisface las necesidades planteadas por los clientes. Alcanza en su totalidad una gran usabilidad e integrabilidad. Finalmente, se ha respetado no saturar al sistema durante su ejecución.
- La facilidad de aprendizaje y de uso es relativamente alta.
- Se han mejorado los conocimientos sobre las herramientas utilizadas, tanto Java, Excel y Windows PowerShell. También con las herramientas de desarrollo como NetBeans.
- El proyecto ha sido realizado bajo las pautas de RUP, cumpliéndose los diferentes pasos que se deben de realizar en cada una de las etapas.
- Se ha realizado un proceso totalmente profesional, desde las reuniones con los clientes, siguiendo por la planificación, análisis y codificación.

9.2. Dificultades Encontradas

El objetivo planteado ha sido alcanzado con éxito y cumple con las expectativas propuestas. Sin embargo, varios han sido los obstáculos que se han interpuesto en su desarrollo. Una primera dificultad fue saber como llevar a cabo el trabajo. Las ideas planteadas no eran factibles en el sistema que se imponía, se trató de hablar con los responsables informáticos de Renault de solicitar permisos para poder realizar otra planificación a la realizada pero fue denegada dicha posibilidad.

Otro obstáculo fue la seguridad, intentar imponer un sistema seguro fue un reto que finalmente fue solucionado mediante cifrado de ficheros y comprensión por parte de los administradores, que nunca habían usado esa funcionalidad.

Un último obstáculo fue realizar un sistema rápido y sin muchos tiempos de demora. Los trabajadores que utilizan este sistema, a menudo, no tienen mucho tiempo y su paciencia no es muy grande debido al entorno laboral. Por lo que el sistema debía de ser rápido y ágil en los tiempos de respuesta y una interfaz fácil y sencilla.

9.3. Conocimiento Adquirido

En este apartado se comentará cuales han sido los conocimientos adquiridos la realización de este proyecto.

La realización de este Trabajo de Fin de Grado ha permitido enfrentarme y resolver un problema de gran envergadura. La solución ofrecida se ajusta a la calidad esperada por el cliente.

En segundo lugar, llevar una investigación sobre las posibles implementaciones aporta gran conocimiento y experiencia. Posibles tecnologías, líneas de trabajo y terminologías han enriquecido mi conocimiento.

9.4. Futuros Objetivos y Ampliaciones

Al finalizar este proyecto se vio que resultaba de gran utilidad, no solo para el departamento de Pintura sino también para otros departamentos de la factoría. Por lo que fue planteado, en un tiempo futuro, realizar otra serie de herramientas con una funcionalidad de filosofía similar. Con esta implementación se muestra como se puede reducir el tiempo y el gasto en el departamento de Pintura. Con este pretexto se intentará obtener mejores recursos para poder realizar una nueva reimplementación de este sistema, con una mayor funcionalidad y para otros departamentos de la factoría.

Durante el año 2017, Renault adquirió un lote de tablets para los JU (usuarios de esta aplicación). Ante esta novedad, se planteó la posibilidad de los trabajadores pudieran utilizar la tablet para realizar pedidos. El objetivo planteado es instalar un servidor WEB dentro del departamento para ofrecer un servicio de esta aplicación. Para ello se ofrece la posibilidad de realizar el pedido a través de una página web o a través una App.

- [1] **Informática industrial**,
ISBN: 978-84-9732-614-8,
Carlos Valdivia Miranda
- [2] **Introducción a la Informática**,
ISBN: 978-84-415-3286-1,
Claudia Valdés-Miranda Cros
- [3] **El Proceso Unificado de Desarrollo Software**,
ISBN: 84-7829-036-2
Ivar Jacobson, Grady Booch, James Rumbaugh
- [4] **De Fasa a Renault España S. A. : historia de una empresa, historia de una cultura**,
ISBN: 978-84-8448-935-1,
Enrique Espinel Melgar
- [5] **UML y patrones : introducción al análisis y diseño orientado a objetos y al proceso unificado**,
Craig Larman
ISBN: 978-84-205-3438-1
- [6] **Manual imprescindible Java 7**,
ISBN: 978-84-415-2987-8
Francisco Javier Moldes Teo
- [7] **Historia de Renault**,
www.renault.es/descubre-renault/grupo-renault-espana/estrategia-del-grupo.jsp
Comprobada el 20/12/2017
- [8] **Datos ventas de Renault**,
www.renault.es/media/descubre-renault/att6b3a23c82e064fe599bd119c7ce0177b/RI.RESULTADOS_COMERCIALES.pdf
Comprobada el 20/12/2017
- [9] **Datos factoría Palencia**,
www.renault.es/descubre-renault/grupo-renault-espana/fabricas-espana.jsp *Comprobada el 20/12/2017*
- [10] **Historía de Renault**,
www.renault.es/descubre-renault/el-grupo-renault/la-historia-de-renault
Comprobada el 20/12/2017

- [11] **Historia de Renault**,
www.autobild.es/coches/renault/historia
Comprobada el 20/12/2017
- [12] **WannaCry**,
www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/2169-ccn-cert-id-17-17-codigo-danino-wannacry-1/file.html
Comprobada el 20/12/2017
- [13] **Apache Poi**,
poi.apache.org
Comprobada el 27/10/2017
- [14] **Power Shell tutorial**,
docs.microsoft.com/en-us/powershell/scripting/getting-started/basic-cookbooks?view=powershell-5.1
Comprobada el 20/12/2017
- [15] **Invoke WebRequest**,
docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/invoke-webrequest?view=powershell-5.1
Comprobada el 20/12/2017

APÉNDICE A

APLICACIÓN FORMULARIO: MANUAL DE INSTALACIÓN Y USUARIO

A.1. Manual de Instalación

A.1.1. Introducción

A continuación, se presenta una serie de requisitos y una serie de pasos para que el software desarrollado pueda funcionar correctamente.

A.1.2. Requisitos de Instalación

Este software ha sido desarrollado en lenguaje Java, por lo cual, para poder ser utilizado debe disponerse de la instalación de este. También se utiliza Microsoft Excel, herramienta que va a permitir ver los pedidos realizados. Debido a que este software ha sido desarrollado para un problema específico de Renault, con unas condiciones de trabajo específicas, las condiciones que se citan a continuación serán las que se deben satisfacer para funcionar en ese entorno de trabajo.

- Hardware
 - Velocidad del procesador igual o superior a 500 Mhz.
 - Memoria RAM superior a 256 Kb
 - Disco: espacio libre superior a 2 Gb.
- Software
 - Sistema Operativo Windows 7,8 y 10.
 - Java 7 u 8.
 - Disponer de carpetas compartidas.¹
 - Disponer de permisos de lectura, escritura y ejecución en la carpeta en la que se haya el programa.

¹La idea de este proyecto es que cada JU pueda realizar su pedido desde el ordenador de su puesto u otro. Debido a las limitaciones de seguridad, la única opción para poder utilizarlo desde diferentes equipos, era crear el acceso al programa y los demás archivos necesarios dentro de una carpeta compartida accesible desde diferentes equipos.

A.1.3. Instalación

Para empezar, se dará la suposición de que ya se dispone de un sistema operativo Windows, ya sea 7, 8 o 10. Sobre versiones anteriores no ha sido probado su funcionamiento, por lo que no se garantiza su funcionamiento.

Para poder empezar a funcionar es necesario Java 7 u 8. En el enlace que se presenta a continuación se puede descargar el software y resolver las dudas relacionadas con éste.

www.java.com/es/download/

La herramienta Microsoft Office, la cual contiene Excel, se nos hace indispensable para poder ver los pedidos que se han realizado. Su uso no es gratuito pero hoy en día muchos ordenadores ya disponen de ella. Su descarga se encuentra en el siguiente link.

www.office.com/

En la carpeta que se entrega junto con el presente documento, se encuentran los siguientes ficheros:

- Fichero .jar
- Carpeta lib
- Icono .icon
- Fichero Excel informacion.xlsx
- Fichero Excel plantillaadministrador.xlsx
- Fichero Excel plantillaJU.xlsx

Esta carpeta puede ser copiada íntegramente en cualquier otra carpeta. Para poder ser utilizada por los demás trabajadores deberá ser copiada en una carpeta de acceso compartido desde diferentes equipos.

Una vez copiado se puede proceder a la utilización del software. Como es la primera vez que este se utiliza, detectará este hecho, y creará un fichero oculto con contenido cifrado para poder arrancar. Al ser la primera vez, el programa supondrá que todos los ficheros anteriormente citados se encuentran en la misma carpeta que el fichero .jar. El programa para arrancar toma como referencia el fichero oculto, este debe estar siempre en la misma ubicación que el fichero .jar. Si el programa encuentra el fichero oculto arranca normalmente, en cambio, si este no le encuentra, procederá a crear uno nuevo con toda la configuración original. Cambiar a la configuración original supone eliminar todos los cambios realizados en los ajustes y buscar los ficheros en la misma carpeta que el .jar.

Una vez ejecutado el programa por primera vez, estos ficheros pueden ser movidos de ubicación pero antes el programa debe saberlo, para ello se accede a la configuración del programa y se registran los cambios. Es recomendable aplicar cambios para una funcionalidad más limpia y ordenada. Ajustar el sistema para que nos cree los ficheros en una carpeta específica es una buena idea. También se pueden añadir anotaciones al fichero Excel. También es recomendable crear un acceso directo. Para ello creamos un acceso directo al fichero .jar y le asignamos el icono que viene en la carpeta. Las ventajas que obtenemos así es que podemos poner el acceso directo donde se desee(en otra carpeta, en el escritorio del usuario) para así evitar que el usuario final utilice la carpeta con los ficheros importantes.



Figura A.1: Pantalla principal

A.2. Manual de Usuario

A.2.1. Pantalla Principal

Cuando se ejecuta el fichero .jar el programa empieza a funcionar. Se muestra una ventana en la que se muestra el nombre de la empresa y un botón para pasar a la siguiente pantalla. La funcionalidad que aquí se recoge es meramente informativa. Para pasar a la siguiente pantalla se pulsa sobre el botón Iniciar.

A.2.2. Iniciar Sesión

En la pantalla para iniciar sesión existen dos funcionalidades, una como su nombre indica es para iniciar sesión y la otra es para acceder a los ajustes del sistema.

Para poder iniciar sesión es necesario estar registrado en el sistema, registro que realiza el administrador. Introduciendo el usuario y la contraseña adecuadas nos permite pasar o no a la siguiente pantalla. En caso negativo o de error, se mostrará por pantalla un aviso con los motivos del fallo.

Si en cambio, se desea acceder a la configuración del sistema, solo hace falta hacer clic sobre el texto "¿Eres administrador?".

A.2.2.1. Hacer Pedido Respecto a Otro Mes

Una vez hemos iniciado sesión podemos hacer dos cosas: hacer un pedido desde cero o hacer un pedido similar a otro ya realizado. En la pantalla nos sale dos botones con cada una de las opciones anteriores. Si hacemos clic sobre el botón "Ver pedidos anteriores" se abre un ventana emergente para seleccionar el fichero deseado. Este fichero se corresponde con el fichero generado para el usuario cuando realiza un pedido. Una vez seleccionado, se pulsa el botón abrir y es procesado. Automáticamente se cambia de pantalla y se muestra la pantalla de hacer pedido con los artículos que contenía el fichero.

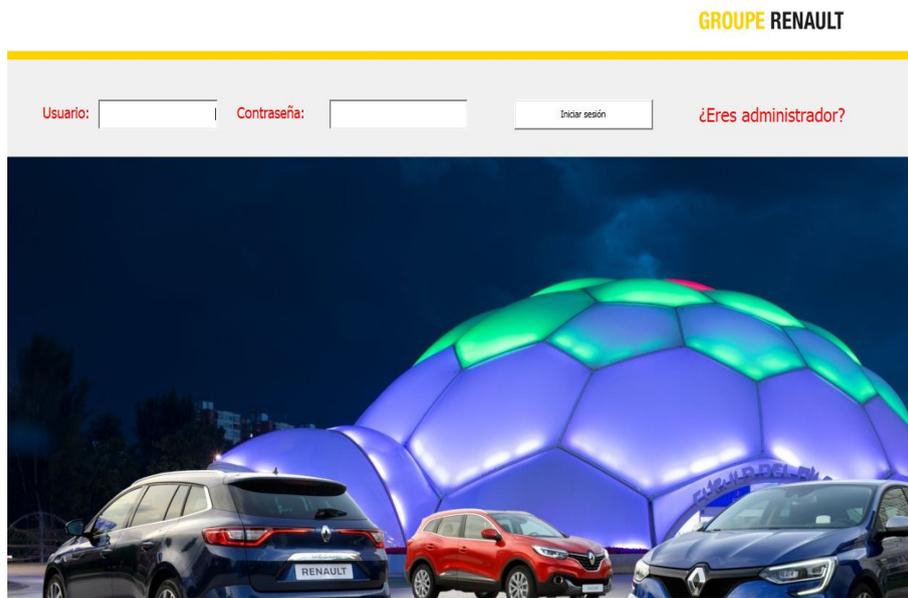


Figura A.2: Pantalla en la que se inicia sesión

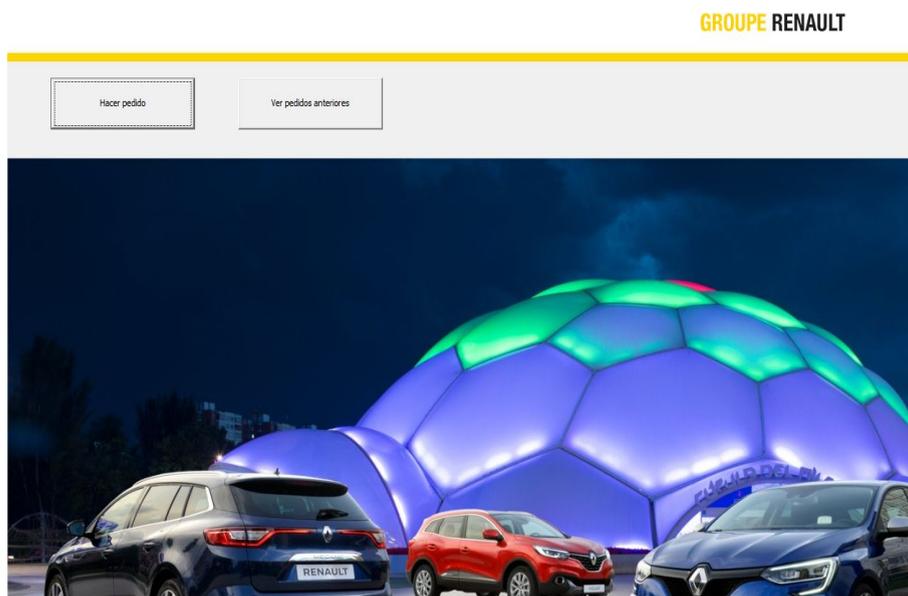


Figura A.3: Menú

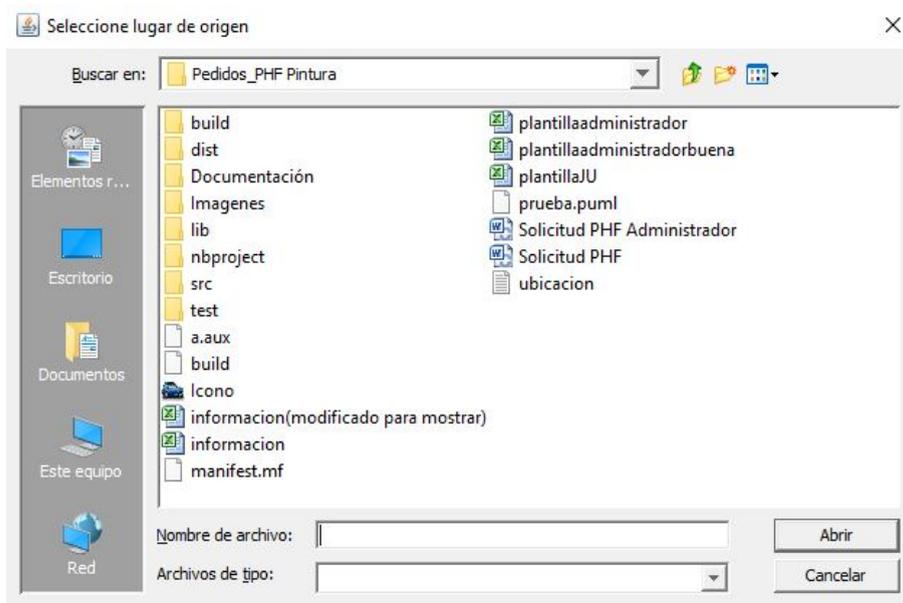


Figura A.4: Pantalla para seleccionar el fichero

A.2.3. Hacer Pedido

Si pulsamos sobre el botón "Hacer pedido" se nos cambia la pantalla y se muestra una nueva en la que podemos buscar artículos o verlos en el catálogo. Una vez añadidos al pedido podemos modificarlos o eliminarlos.

santos Presupuesto inicial: 1000,00 € **GRUPE RENAULT**

Buscar:

REFERENCIA:

CANTIDAD:

Busqueda:

DESCRIPCIÓN:

PRECIO-PAQUETE: 49,68 €

PRECIO-CONJUNTO: 0,92 €

Número de elementos añadidos a la lista: 6

Presupuesto actual: 598,68 €

Quitar elemento

Referencia:

Modificar:

Referencia:

CANTIDAD:

POSICIÓN	ID	DESCRIPCIÓN	CANTIDAD	PRECIO-PAQUETE	PRECIO CONJUNTO
1	R.100640408	Bolsa azul portapiezas	12	12,5	150,00
2	D623078124	Boquilla aplicación mastico en goteron (Robot)	2	32,5	65,00
3	TS53011356	Cinta 50 mm x 25 m. Papel crepe para fijación ficha en aleta (36 rollos)	36	0,52	18,72
4	P93311723A	ESPATULA PLANA MANGO MADERA ANCH.80 L.230	6	1,32	7,92
5	R.100107599	Filtro conico 200mic D:17 lg:135 papel/tejido para filtro pintura en 3 Pasos (Min >>100)	1000	0,11	110,00
6	P93111240A	PINCEL CABO LATA PLANO CERDA PURA N. 18ANCHO 20 VIROLA METALICA MANGO MADERA	54	0,92	49,68

Figura A.5: Pantalla para realizar el pedido manualmente

A.2.3.1. Buscar

Para poder añadir un artículo al pedido primero tenemos que buscarlo. Para ello debemos introducir su referencia y la cantidad en el apartado correspondiente. Si por alguna casualidad la referencia no existe o la cantidad solicitada no es válida se muestra por pantalla un error. Si la referencia y la cantidad es válida, se nos muestra la descripción del artículo, su coste por unidad y el coste total.

A.2.3.2. Ver Catálogo

Puede darse la situación de que no sepamos cual es la referencia del artículo que deseamos. Ante esto existe la posibilidad de ver el catálogo de artículos disponibles, para ello pulsamos el botón de "Ver catálogo" se nos muestra una nueva ventana con el listado de artículos. Podemos volver atrás o si hacemos doble clic sobre la referencia de un artículo, se nos cambiará la ventana por la anterior y se auto rellena el campo referencia con el del artículo seleccionado.

GROUPE RENAULT

Posición	REFERENCIA	DESCRIPCIÓN	CANTIDAD MÍNIMA	PRECIO
1	R100688561	ADAPTADOR MINI EXTREM	1	0.01
2	D623078024	alargador boquilla ciego	1	49.2
3	T803724825	ALCOHOL (BOTELLA DE LITRO)	1	2.5
4	X756001462	BASE ENCHUFE SALIENTE 2P+T (SCHUKO) 10-16A/250V C/TAPA IP55S SERIE PLEKO 5SS MONOBLOC	1	5.65
5	R100596974	Bayeta atrapapolvo impregnada 40x40 cm 70 gr/m2 50% celulosa. 50% poliester. Ep=0,65 mm color verde	25	0.3
6	R100274096	Bayeta de limpieza blanca 32x60 cm 66% viscosa, 33% poliester. Sin bolsa de plastico. HENKEL REF.FAB: LU02 32/60 P3/500	500	0.09
7	R100660581	Bayeta de microfibras de 87% polyester 13% nylon 36 X 36 cm Ep 0,75 mm (3M)	50	0.95
8	P93585190R	Bayeta Tricotada (pequeña)	400	0.2
9	X97413590A	BLOC ESPIRAL 80 HOJAS 105X155MM (OCTAVO) PAPEL BCO 63GR/M2	1	0.31
10	X971141270	Boligrafo escritura normal bic (rojo)	1	0.15
11	R100154042	BOLSA 12 KGS. DE BAYETAS 34X54 CM. 100%	1	83.84
12	R100048782	Bolsa 300 etiquetas autoadhesivas	1	0.44
13	R100640408	Bolsa azul portapiezas	1	12.5
14	P93556520A	Bonete-Boina de borregullo D. 75 mm	75	1.93
15	R100317510	Boquilla 4- 4 pulverizado	1	155.0
16	D623078124	Boquilla aplicación mastico en goteron (Robot)	1	32.5
17	R100453107	boquilla aspiración	1	5.93
18	D623078119	boquilla ciego	1	129.5
19	D623078115	boquilla extruido	1	64.5
20	P93152418A	BOQUILLA PARA APLICACION DE MASTICO EN CHARNELAS	1	58.0
21	D623078054	boquilla pico pato	4	154.0
22	D623078113	boquilla rebatido	1	63.0
23	E831792126	boquilla robot	1	28.9
24	7.711427592E9	Bote spray para Pintura y BDM	6	9.69
25	R100616448	Brazalete Naranja	1	2.6
26	R100616576	Brazalete Negro	1	2.6
27	R100713636	BUSE (PLAT) APLICACION DE MASTICS 0. 3X8	1	212.0
28	R100270047	BUSE ALBUZ SCA 2-4 CORPS INOX EQUIPEE PASTILLE ET HELICE CERAMIQUE	1	155.0
29	D623078125	BUSE APLICATION MASTIC CAPOT AV VEH.X-95(ROBOT)	1	71.0
30	R100615290	BUSE APPLICATION MASTICS D. 10	1	55.0
31	R100617881	BUSE APPLICATION MASTIC EXTREME BAS CARREER D. 10	1	74.0

Volver

Figura A.6: Pantalla en la que se muestra el catálogo

A.2.3.3. Añadir

Una vez realizada la búsqueda y mostrados los datos, podemos estar de acuerdo o no con el artículo. Si estamos conformes pulsamos sobre el botón añadir o sino podemos modificar la búsqueda. Una vez añadido se actualiza el presupuesto actual, el número de artículos y se añade la información de éste a la lista que aparece por pantalla sobre los artículos que ya se han pedido.

A.2.3.4. Modificar

Una vez añadido un artículo tenemos la posibilidad de modificar este. En la sección modificar introducimos la referencia y la cantidad nueva. Pulsando sobre el botón "Modificar" solicitamos al sistema el cambio, si éste es válido lo cambiará, al contrario, si no es válido, lo rechazará mostrando un mensaje de error por pantalla con el motivo del fallo.

A.2.3.5. Eliminar

De igual manera que es posible modificar un artículo ya pedido podemos eliminarlo de la lista. Para ello, dentro de la sección eliminar, introducimos la referencia del artículo y pulsamos el botón “Eliminar”. En caso de que sea rechazado se mostrará un mensaje por pantalla indicando el motivo del fallo.

A.2.4. Enviar, Artículos No Pedidos y Creación del Fichero

Una vez que se han añadido los artículos deseados se procede a su envío. El sistema nos dejará enviarlo si nuestro saldo está en positivo, de lo contrario no dejará enviarlo y se deberá ajustar el pedido. Si el pedido es válido, se procede a su envío. Se cambia la pantalla y sale una nueva que contiene una lista de artículos obligatorios a pedir y que no se encuentran en la lista realizada. En el caso de que quede alguno por pedir, solo se permite volver a la ventana anterior y pedirlo. En caso de que no existe ningún artículo pendiente se nos permite seguir hacia delante o hacia atrás.

Si se procede a seguir hacia delante, se nos pide seleccionar un ubicación en la que se guardará un fichero Excel que simula ser un ticket sobre el pedido realizado.

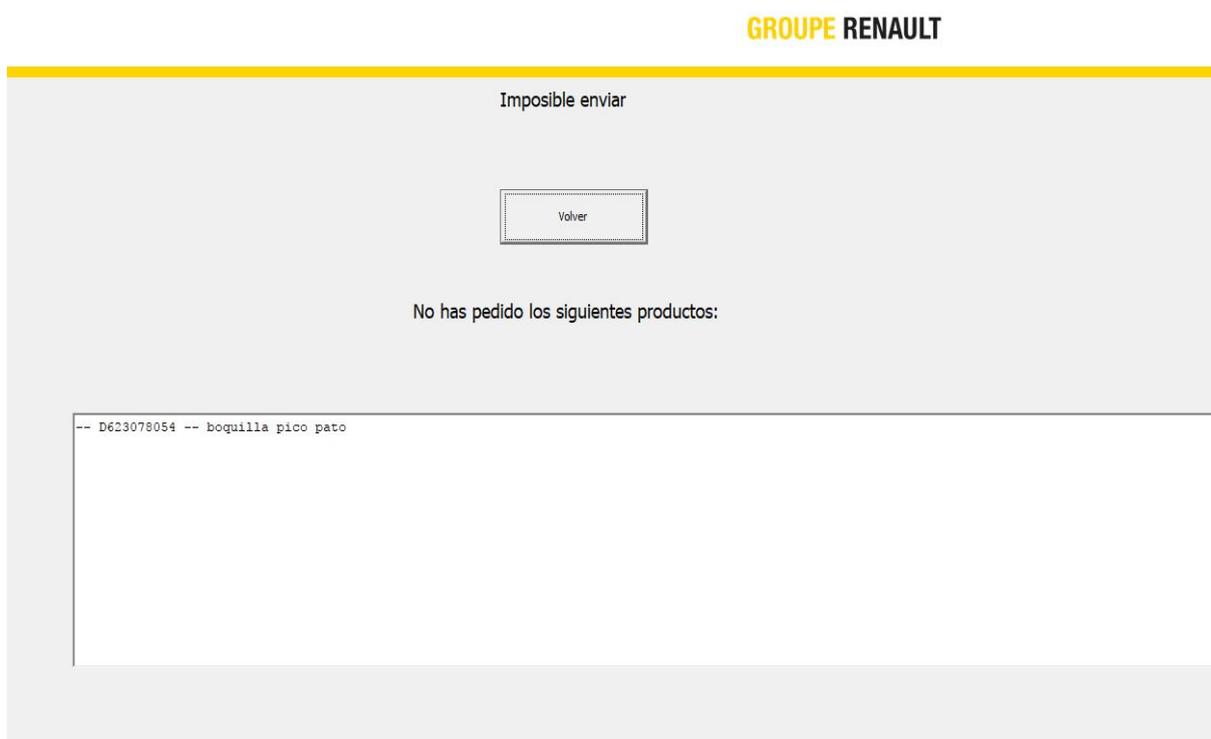


Figura A.7: Pantalla con los artículos pendientes

A.2.5. Acceder a los Ajustes

En la ventana de Iniciar-Sesión se nos mostraba otra alternativa, esta era la de acceder a los ajustes del sistema. Pulsando sobre los botones “¿Eres administrador? se accede a una nueva ventana. En ella se debe introducir una clave para poder acceder a los ajustes. Si la clave introducida es válida permitirá acceder a la siguiente ventana de lo contrario nos mostrará un mensaje de error.

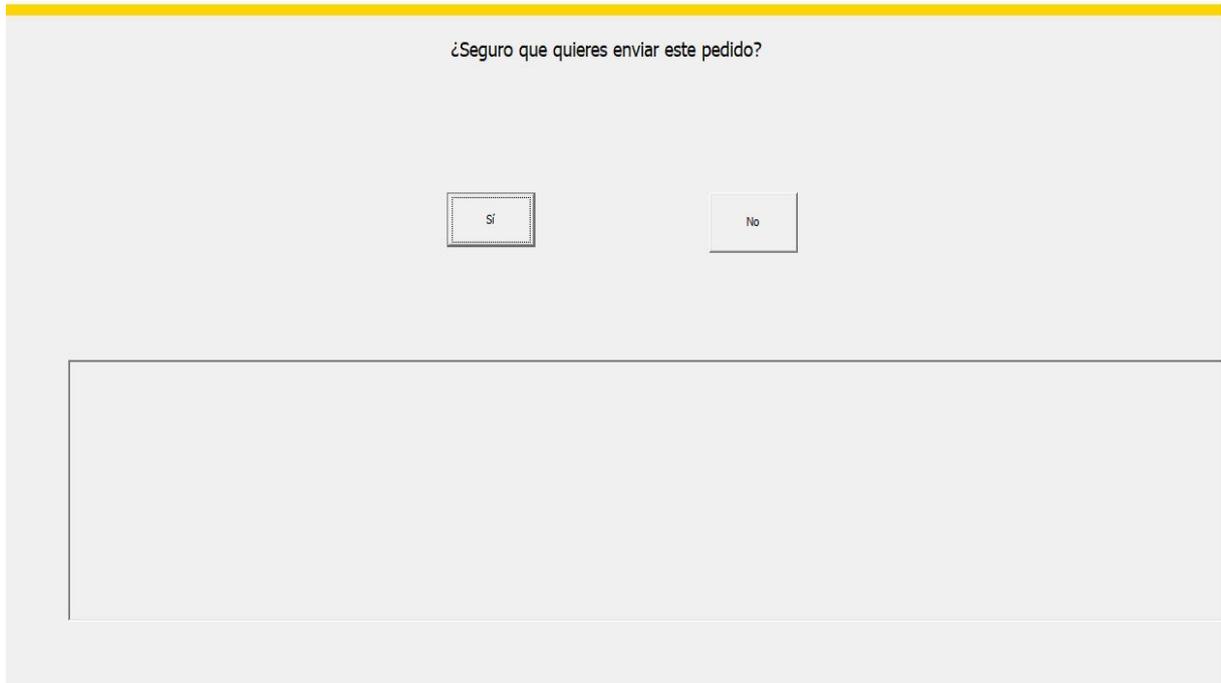


Figura A.8: Pantalla para el envío del pedido



Figura A.9: Pantalla para acceder a los ajustes

A.2.6. Ajustar el Sistema

Una vez accedido con la clave correcta se presenta una nueva ventana con las configuraciones sobre el sistema. Para cambiar las configuraciones debemos realizar los cambios y después sobre el botón guardar.

Localizaciones:

Ubicación plantilla	<input type="text" value="plantillaadministrador.xlsx"/>	<input type="button" value="Seleccionar"/>
Ubicación plantilla JU	<input type="text" value="plantillaJU.xlsx"/>	<input type="button" value="Seleccionar"/>
Ubicación fichero de datos	<input type="text" value="informacion.xlsx"/>	<input type="button" value="Seleccionar"/>
Ubicación fichero gerente	<input type="text" value="C:\Users\Santos\Desktop\"/>	<input type="button" value="Seleccionar"/>

Cambiar contraseña de acceso al menú de configuración:

Contraseña actual:

Nueva contraseña:

Repita contraseña nueva:

Resetear configuración

Cambiar contraseña excel de datos:

Contraseña actual:

Nueva contraseña:

Repita contraseña nueva:

Permitir más de un envío al mes Permitir

Figura A.10: Pantalla para realizar ajustes

APÉNDICE B

APLICACIÓN ENVÍO: MANUAL DE INSTALACIÓN Y USUARIO

B.1. Manual de Instalación

B.1.1. Introducción

En las siguientes secciones se realiza una explicación de que recursos se necesitan para el correcto funcionamiento de la aplicación y de cómo ponerla en funcionamiento. La explicación que se dará será totalmente independiente de la aplicación desarrollada en el proyecto A a pesar de ser una fuente de datos de la otra.

B.1.2. Requisitos de Instalación

El software ha sido desarrollado en Java, por lo que disponer de Java es un requisito esencial para su utilización. Debido a que no es necesario abrir ningún fichero Excel por parte del usuario no es esencial que el usuario disponga de Microsoft Office. Estos requisitos son orientativos, el sistema está programado totalmente en JAVA y usa la herramienta PowerShell del sistema operativo Windows. Para establecer los requisitos mínimos de software se han tomado de referencia los establecidos por Oracle para JAVA 7. Como esta herramienta ha sido diseñada para un uso específico en Renault y este ya contaba con equipos informáticos con un entorno Windows (el cual incluye PowerShell) no se ha visto conveniente establecer como requisitos mínimos los requisitos que necesite Microsoft Windows 7 para funcionar.

- Hardware
 - Memoria RAM superior a 128 MB
 - Espacio en disco de 124 MB para JRE y 2 MB para Java Update
 - Procesador mínimo: Pentium 2 a 266 MHz
- Software
 - Sistema Operativo Windows 7.
 - Java 7 u 8.
 - Disponer de acceso a la red interna de Renault.¹

¹Para poder disponer de acceso a esta subred es necesario ser empleado de Renault y tener un equipo con permiso para acceder a ella. Este permiso es otorgado por el departamento de informática de la empresa, el cual centraliza todas las gestiones.

- Disponer de los permisos necesarios para poder realizar pedidos.²

B.1.3. Instalación

Para empezar, se dará la suposición de que ya se dispone de un sistema operativo Windows 7. Sobre versiones anteriores ha Windows Vista no funciona debido a que estos sistemas operativos no incorporan la herramienta PowerShell. Para versiones superiores a Windows 7 no ha sido probado aunque en principio no debiera dar problema alguno.

Para poder empezar a funcionar es necesario Java 7 u 8. Se supone que los ordenadores de Renault cuentan por defecto con Java y en caso de que no va a ser imposible su instalación por parte del usuario por lo que no es necesario explicar más sobre JAVA. Aunque, por si acaso, en el siguiente enlace se puede obtener el software y resolver dudas acerca de este. En el enlace que se presenta a continuación se puede descargar el software y resolver las dudas relacionadas con este.

www.java.com/es/download/

De igual manera Office no es necesario para la ejecución de este software no tampoco está incluido en los requisitos mínimos. Aunque, por si acaso, en el siguiente enlace se deja información acerca de este.

www.office.com

En la carpeta que se entrega junto con el presente documento, se encuentran los siguientes ficheros:

- Fichero .jar
- Carpeta lib
- Icono .icon

Esta carpeta puede ser copiada íntegramente en cualquier otra carpeta. Para que una persona pueda ejecutar el programa debe disponer de acceso a la carpeta mencionada anteriormente o una copia de ella.

Para poder iniciar el programa tan solo hace falta ejecutar el fichero .jar y seguir navegando por la aplicación. Para poder tener un acceso más rápido al programa, se puede hacer un acceso directo al fichero .jar y colocar este en cualquier parte del equipo. Esto permite reducir el riesgo de eliminar o modificar alguna carpeta o fichero de lib.

B.2. Manual de Usuario

B.2.1. Pantalla de Inicio

Cuando se ejecuta el fichero .jar el programa empieza a funcionar. Se muestra una ventana en la que se muestra el nombre de la empresa y un botón para pasar a la siguiente pantalla, figura B.1. La funcionalidad que aquí se recoge es meramente informativa. Para pasar a la siguiente pantalla se pulsa sobre el botón Iniciar.

²No todos los empleados que tengan acceso a la red interna tienen permiso para realizar pedidos. De igual manera, esta gestión es llevada por el equipo de informática de la empresa.

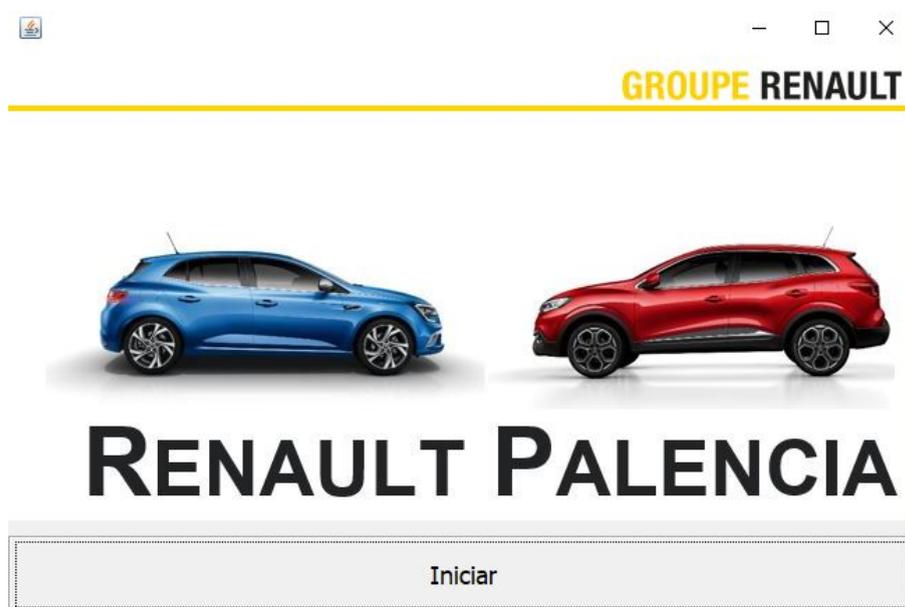


Figura B.1: Pantalla de inicio

B.2.2. Iniciar Sesión

La funcionalidad básica de esta nueva pantalla es para iniciar sesión, figura B.2. En ella se introduce el usuario y la contraseña. Estos datos no son validados en este momento debido a que el servidor no está preparado para esto, pero más adelante sí que son comprobados. Esto no implica que no se deban rellenar estos campos para poder iniciar sesión. En caso de que alguno de los campos quede vacío se mostrará un mensaje avisando al usuario. Para finalizar la pantalla anterior hay que pulsar sobre el botón "Iniciar sesión", los datos son guardados por el programa y enviados más tarde al servidor.

B.2.3. Enviar Pedido

En esta nueva pantalla, figura B.3, que se presenta se procederá a seleccionar el fichero Excel que contiene los datos y a enviarlos.

El primer paso que se debe de dar es seleccionar el fichero Excel que contiene los artículos que deseamos pedir. Para ello hacemos clic sobre el botón seleccionar, el cual nos abre un selector de ficheros. Seleccionamos el fichero Excel y le damos a "Abrir". Automáticamente el fichero es abierto, en caso de contener contraseña nos solicita ésta para poder abrirlo.

Nada más abrirlo, se leen los artículos y se muestran por pantalla para que el usuario sepa cuáles son los que se van a pedir. Se muestran los datos de referencia, cantidad, descripción y centro de gasto asociado de cada uno.

Si estamos de acuerdo con todo el último paso que se debe de dar es pulsar sobre el botón "Solicitar". Una vez pulsado comienza el proceso. En el cuadro de texto colocado a la derecha se muestra información del proceso, información acerca del tiempo que va a tardar, cuantos artículos va a pedir, validación del inicio de sesión, etc. Al acabar se muestra el número asociado al pedido y cuáles son los artículos que se han pedido.

Si en algún momento se produce algún conflicto con algún artículo, se aborta el pedido y se muestra por pantalla información acerca de que se ha abortado el proceso de pedido.

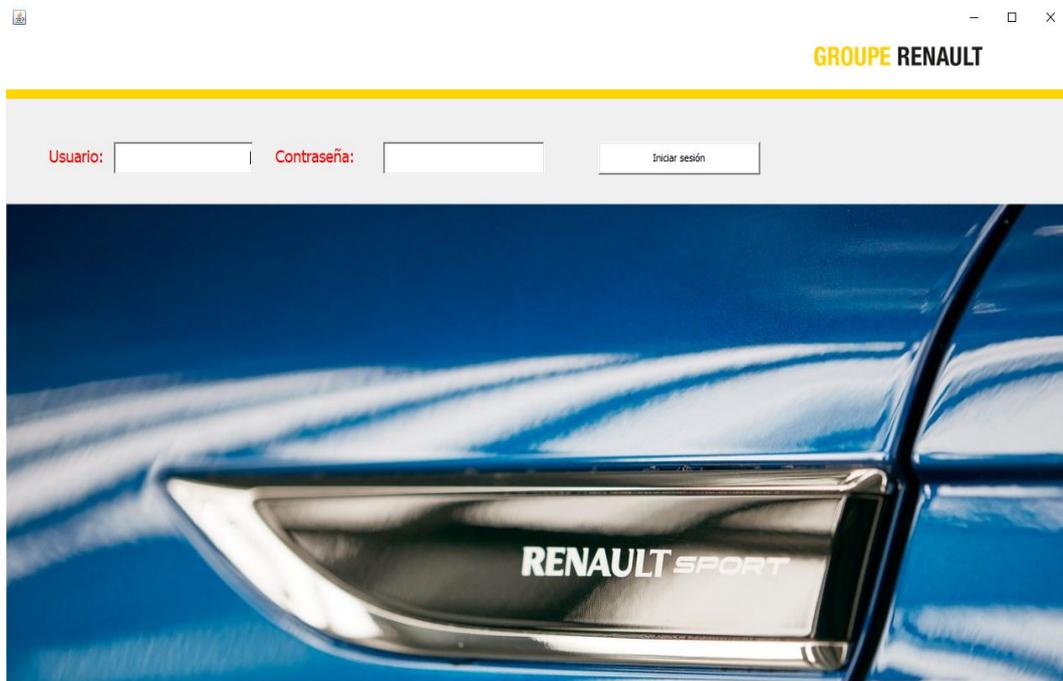


Figura B.2: Pantalla de iniciar sesión

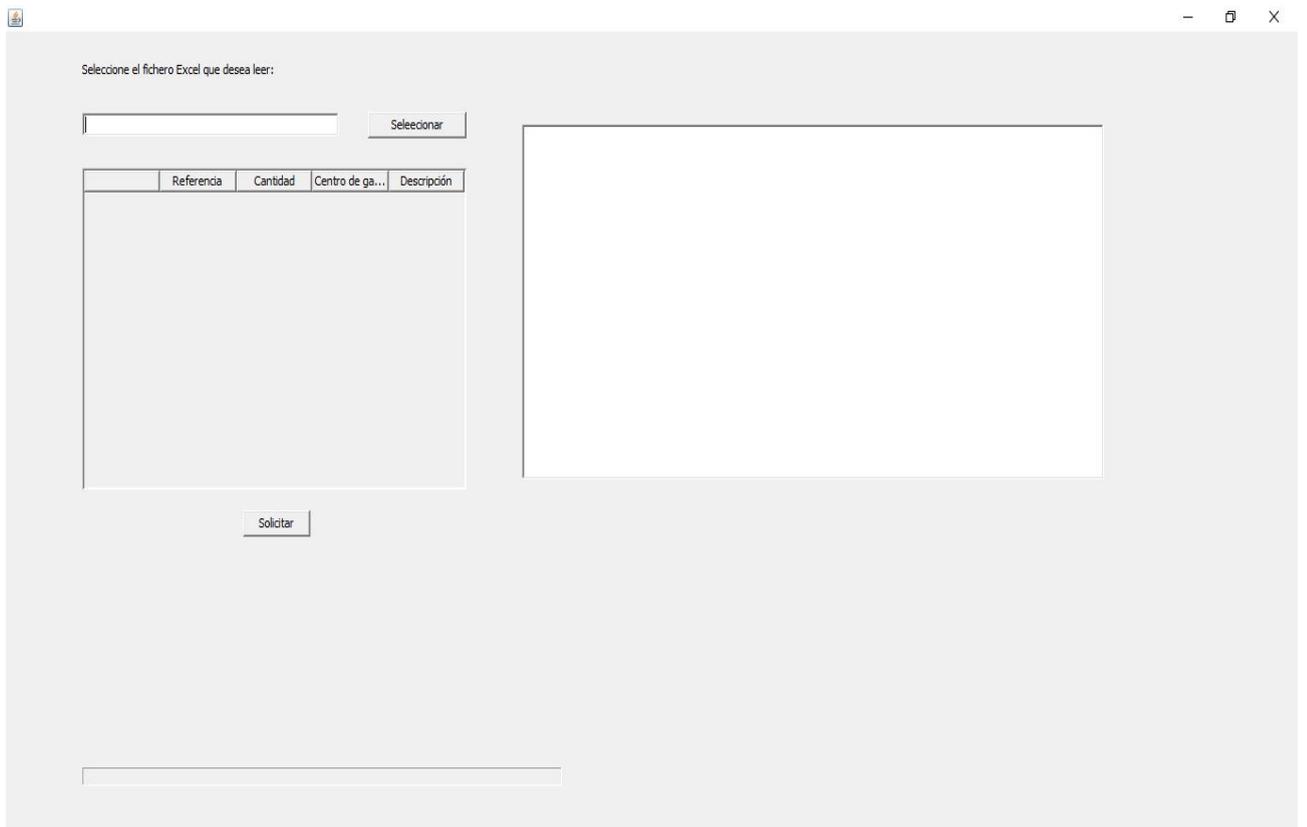


Figura B.3: Pantalla de enviar pedido