

TRABAJO FINAL DE CARRERA J2EE

Aplicación Web para la gestión de una tienda de peces exóticos



Alumno: Francisco Menacho García

Consultor: Albert Grau

13 de Enero de 2014

Agradecimientos

Me gustaría dedicar este proyecto a muchas personas por la paciencia, ayuda o tiempo que no les he podido prestar. En especial quiero dedicárselo a mi pareja, M^a José ya que sin ella esto no hubiera sido posible.

A los demás, mis padres, mi hermana, mi sobrina, mis suegros, gracias por haberme ayudado a hacer esto posible y apoyado en todo momento y por supuesto a mi tutor del trabajo fin de grado ya que sin duda alguna si él esto no hubiera sido posible.

Índice

Capítulo 1. Introducción.....	6
1.1 Justificación del TFC y contexto en el cual se desarrolla: punto de partida y aportación del TFC.....	6
1.2 Objetivos del TFC.....	7
1.2.1 OBJETIVOS GENERALES DEL TFG	8
1.2.2 OBJETIVOS ESPECIFICOS DEL TFC	9
1.3 HARDWARE Y SOFTWARE PROPUESTOS	10
1.3.1 Hardware.....	10
1.3.2 Software.....	10
1.4 ESTRUCTURA DEL PROYECTO.....	10
1.4.1- Descripción de las actividades.....	11
1.4.2- Planificación TFC - PAC 1.....	11
1.4.3- Análisis diseño y prototipo – PAC 2.....	11
1.4.4- Desarrollo J2EE – PAC 3.....	11
1.4.5- Entrega final: Elaboración Memoria, EAR y Presentación TFC	12
1.5.- Planificación.....	12
1.5.1 - DIAGRAMA DE GANTT.....	13
1.5.2 - FECHAS CLAVES DEL PROYECTO.....	14
1.5 Productos obtenidos.....	14
1.6 Breve descripción de los otros capítulos de la memoria.....	14
Capítulo 2. Análisis funcional.....	15
2.1 Descripción del funcionamiento del sistema.....	15
2.2. Análisis de requisitos funcionales.....	16
2.2.1- Casos de uso y sus Fichas.....	17
Diagrama general de los casos de uso.....	17
2.2.2 Fichas de gestión de casos de uso	18
2.3. Análisis de requisitos no funcionales.....	26
2.3.1 Calidad.....	26
2.3.2 Coste.....	27
3.- Diagrama de clases principal.....	30
3.1 Diagrama de clases.....	30
4.- Diagrama relacional de base de datos.....	32
5.- Diagrama de estados.....	33
5.1 DIAGRAMA DE ESTADOS PARA PECES EXÓTICOS.....	33
5.2.- DIAGRAMA DE ESTADOS PARA ACCESORIOS.....	34
5.3 DIAGRAMA DE ESTADOS PARA ACUARIOS.....	35
6.- Prototipos de las pantallas.....	35
6.1- Pantalla de Acceso.....	35
6.2.- Pantalla de registro.....	38
6.3.- Pantalla del administrador.....	39
6.4.- Gestión de peces exóticos.....	40
6.5.- Alta de pez exótico.....	41
6.6.- Modificación pez exótico.....	43
6.7.- Eliminación pez exótico.....	44
6.9.- Pedido con usuario registrado.....	46
6.10.- Listado de pedidos.....	47

6.11.- Mostrar detalle del pez exótico.....	47
7.- Diagrama de arquitectura.....	48
7.1 PostgreSQL.....	50
7.2 Jboss.....	51
7.3 Instalación del Conector Java JDBC PostgreSQL a Jboss.....	53
8. Conclusiones.....	56
8.1 Conclusiones personales.....	56
8.2 Justificación y razonamiento de las decisiones adoptadas durante el proyecto.....	57
8.3 Experiencias o conclusiones extraídas de este TFG.....	58
9. Bibliografía.....	60

Índice de figuras y tablas

Figura 1. Arquitectura en 3 capas	8
Figura 2. Diagrama de Gantt	13
Figura 3. Diagrama de casos de uso.....	17
Tabla 1. Temporarización	14
Tabla 2. Caso de uso de la acción login	18
Tabla 3. Caso de uso para añadir un acuario	18
Tabla 4. Caso de uso para eliminar un acuario	19
Tabla 5. Caso de uso para modificar un acuario	19
Tabla 6. Caso de uso para consultar un acuario	20
Tabla 7. Caso de uso para añadir un pez exótico	20
Tabla 8. Caso de uso para eliminar un pez exótico	21
Tabla 9. Caso de uso para modificar un pez exótico	21
Tabla 10. Caso de uso para consultar un pez exótico	22
Tabla 11. Caso de uso para añadir un accesorio	22
Tabla 12. Caso de uso para eliminar un accesorio	23
Tabla 13. Caso de uso para modificar un accesorio.....	23
Tabla 14. Caso de uso para consultar un accesorio	23
Tabla 15. Caso de uso para añadir un usuario registrado	24
Tabla 16. Caso de uso para salir de la sesión	24
Tabla 17. Caso de uso para realizar un pedido	25
Tabla 18. Caso de uso para consultar pedidos	25
Figura 4. Arquitectura Modelo-Vista-Controlador (MVC)	28
Figura 5. Diagrama de clases	30
Figura 6. Diagrama de base de datos	31
Figura 7. Diagrama de estados peces exóticos	33
Figura 8. Diagrama de estados de accesorios	33
Figura 9. Diagrama de estados de acuarios	34
Figura 10. Pantalla de acceso	35
Figura 11. Pantalla de registro de usuarios	36
Figura 12. Pantalla del administrador	36
Figura 13. Pantalla del gestión de peces exóticos	37
Figura 14. Pantalla de alta de peces exóticos	38
Figura 15. Pantalla de modificación de peces exóticos	38
Figura 16. Pantalla de eliminación de peces exóticos	39
Figura 17. Pantalla de acceso de usuarios registrados	40
Figura 18. Listado de peces exóticos de un usuario registrado	40
Figura 19. Pantalla de realización de pedidos	41
Figura 20. Pantalla de listado de pedidos	41
Figura 21. Pantalla de desalle de peces exóticos	42

Capítulo 1. Introducción

La motivación principal en la realización de este proyecto es diseñar y programar una aplicación de gestión distribuida útil para ayudar a la empresa Aqua Net a satisfacer sus necesidades.

Se me ocurrió realizar un proyecto web para la gestión de dicha empresa por un conocido (el jefe de la misma), ya que es una necesidad importante en los tiempos actuales que discurren. Además para mí, ha supuesto la rampa de salida para el conocimiento y la práctica en el manejo del conjunto de tecnología del que se compone.

1.1 Justificación del TFC y contexto en el cual se desarrolla: punto de partida y aportación del TFC

Las aplicaciones distribuidas web han cogido mucho protagonismo en los últimos años ya sea para darse publicidad o para hacer de puente entre otros sectores.

En el mundo actual en el que las empresas son cada vez menos centralizadas, es mayor la necesidad de que la información se envíe de forma simple a todas las personas del planeta.

Para ello, el uso de una red como Internet se consigue que con un cliente (navegador como Internet Explorer, Mozilla Firefox o Chrome) cualquier persona pueda acceder a la información contenida en ella (información con referencia a ferreterías, investigación, e infinidad de

aplicaciones más). También es importante ver que este tipo de aplicaciones son válidas incluso en Intranet porque aún así proporciona un escenario simple.

Por tanto, la construcción de una aplicación de gestión que ayuden a la gestión de la empresas, puede ser un buen punto de partida para nuevas empresas que abran sus puertas y requieran una herramienta para controlar y gestionar ,por ejemplo, sus productos y stocks.

Este TFC se basa en la necesidad de crear una aplicación web que permita la consulta y gestión por Internet de todo lo relacionado con una empresa de acuarios exóticos(peces exóticos, productos, accesorios, etc.) empleando las tecnologías oportunas para conseguir un tiempo de respuesta lo más bajo posible en las interacciones de los usuarios con la aplicación y una buena gestión de la empresa. Todo esto haciendo uso de la plataforma J2EE.

A nivel personal, creo que la realización del presente TFC me permitirá por un lado, poder agrupar en un solo proyecto los conocimientos adquiridos a lo largo de la Ingeniería en diferentes asignaturas, como Programación orientada a objetos, Bases de Datos ó Ingeniería del Software y por otro, la oportunidad de poder introducirme en una arquitectura no muy conocida para mí, como es J2EE.

La tecnología a utilizar en el proyecto será en principio el frameworks de JSF, JPA, Hibernate, Facelets, EJB.

En conclusión, la idea para el TFG es realizar una aplicación de gestión para una empresa de acuarios exóticos, el cual pueda ayudar a la empresa, agilizando la gestión de stock de sus productos, facturación, etc... Además, se espera un nivel aceptable de seguridad y unos buenos tiempos de respuesta a las peticiones realizadas por los usuarios y por otra parte, la posibilidad de desarrollar un producto completo y el aprendizaje de nuevas tecnologías que me han llegado a motivar.

1.2 Objetivos del TFC

1.2.1 OBJETIVOS GENERALES DEL TFG

El objetivo principal de este TFC es profundizar en el uso de la tecnología Java y adquirir conocimientos sobre una arquitectura distribuida en 3 capas con J2EE mediante el desarrollo del proyecto que se llamará GestAquarium.

Posible arquitectura en 3 capas (Presentación, negocio y acceso a datos):

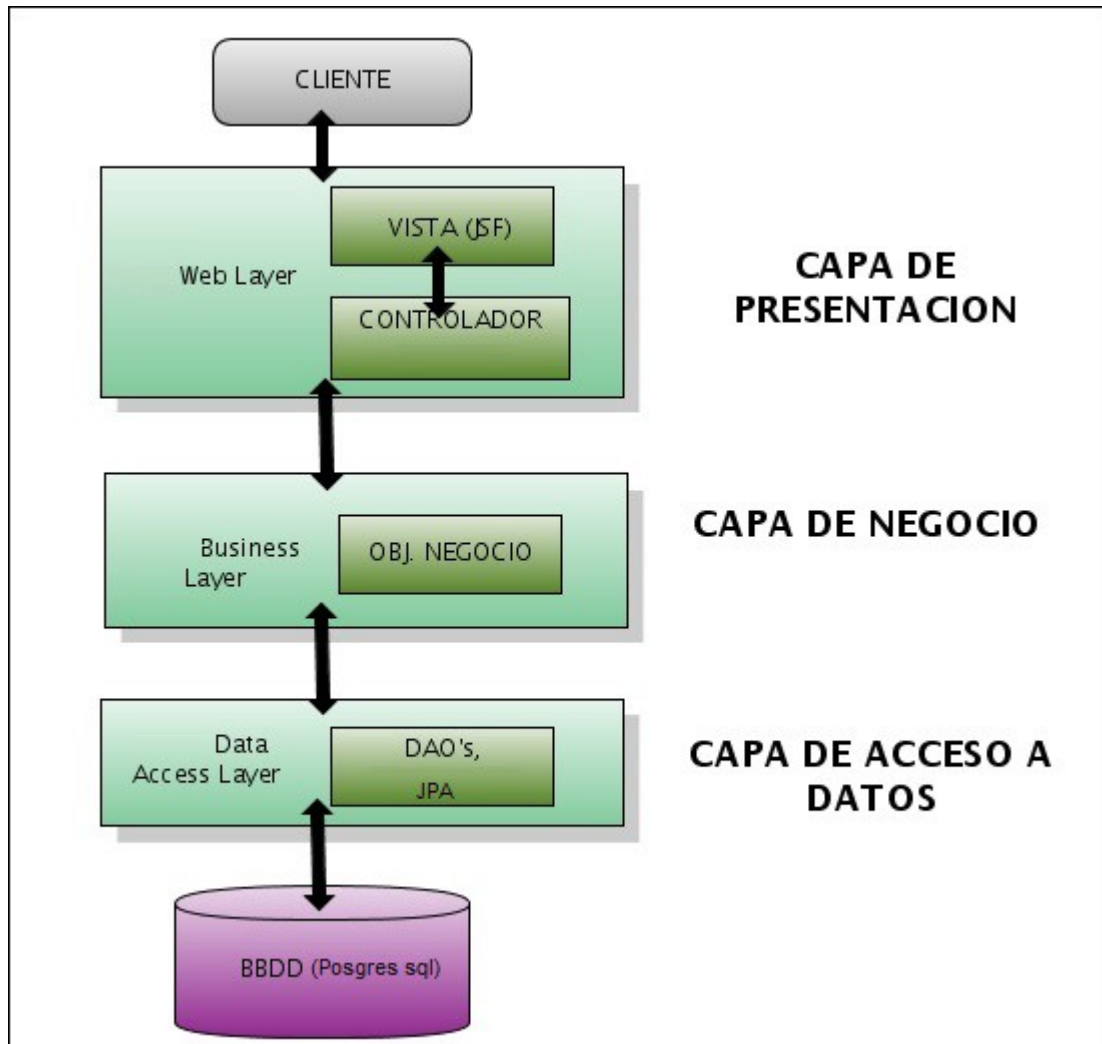


Figura 1. Arquitectura 3 capas

Se tendrá que desarrollar los siguientes objetivos:

³⁵/₁₇ - Elaboración de un Plan de Trabajo.

³⁵/₁₇ - Análisis, diseño e implementación de una aplicación similar a las existentes en el mundo de las aplicaciones de gestión web, pero utilizando la

orientación a objetos.

³⁵₁₇ - Patrones de diseño aplicables a la arquitectura J2EE.

³⁵₁₇ - Comunicación con bases de datos relacionales.

³⁵₁₇ - Frameworks: JSF, JPA, etc.

³⁵₁₇ - Elaborar una Memoria Final del Proyecto y su correspondiente

³⁵₁₇ Presentación Grafica

1.2.2 OBJETIVOS ESPECIFICOS DEL TFC

Alguno de los objetivos específicos o finales del TFC son los resultados que debemos de implementar para que la empresa de acuarios "Aqua Net", consiga la gestión de su empresa, rentabilidad y eficacia deseada.

Para ello, se realizaran los siguientes objetivos específicos:

- Crear una aplicación de gestión distribuida en 3 capas para la empresa especializada en acuarios exóticos con el que se pueda gestionar, al menos, la siguiente información:

³⁵₁₇ Gestión de usuario(Usuario registrado).

³⁵₁₇ La identificación se realizará mediante login inicial.

³⁵₁₇ Gestión de acuarios(alta, baja, modificación y consulta).

³⁵₁₇ Gestión de peces exóticos para acuarios (alta, baja, modificación y consulta).

³⁵₁₇ Gestión de accesorios para acuarios (alta, baja, modificación y consulta).

³⁵₁₇ Pedidos (no venta) de acuarios, peces exóticos y accesorios (Usuario registrado)

1.3 HARDWARE Y SOFTWARE PROPUESTOS

1.3.1 Hardware

Se utilizará en este proyecto el siguiente hardware:

³⁵₁₇ Ordenador portátil Asus : Intel Core I7 de 8 nucleos a 2,3 Ghz, 8 Gb de memoria , 1 Tb de HDD (Disco Duro) con Sistema Operativo (SO) Windows 7 Service Pack (SP) 1.

1.3.2 Software

Se utilizará en este proyecto el siguiente software:

³⁵₁₇ **Sistema de Gestión de Bases de Datos Relacionales** (SGBDR): Postgres SQL v9.3

³⁵₁₇ **Diagramas (análisis)**: Magic Draw Personal Edition

³⁵₁₇ **Desarrollo Java**: IDE Eclipse Kepler con JDK 1.6, frameworks como JSF, JPA, EJB, etc.

³⁵₁₇ **Desarrollo SQL**: pgAdmin III

En principio no se descarta usar otro IDE como NetBeans para el desarrollo de la aplicación, si surgiera la necesidad de hacerlo así.

Por ejemplo para los diagramas E/R, podría ser perfectamente factible usar el Enterprise Architect ya que es un excelente programa utilizado para la representación de la información.

Por último, la planificación que se encuentra en este mismo documento se ha desarrollado con Software Libre (SL) OpenProj.

1.4 ESTRUCTURA DEL PROYECTO

Para el desarrollo de la aplicación se ha intentado seguir el enfoque de una empresa de desarrollo de software. Esto implica que no sólo es importante el resultado final, sino que éste debe haberse acotado previamente de manera clara, debe generarse la documentación adecuada a cada fase y deben alcanzarse la

consecución de los objetivos en los plazos establecidos.

Para ello, se ha seguido un ciclo de vida en cascada clásico en el que se han previsto y realizado la siguiente descripción de actividades:

1.4.1- Descripción de las actividades

Las principales actividades que componen el proyecto son las siguientes:

1.4.2- Planificación TFC - PAC 1

Esta tarea comprende las siguientes actividades:

- ³⁵₁₇ Elaboración del plan de trabajo de este TFC
- ³⁵₁₇ Entrega del plan de trabajo de este TFC
- ³⁵₁₇ Análisis (Inicio Diagrama de clases)

1.4.3- Análisis diseño y prototipo - PAC 2

Esta tarea comprende las siguientes actividades:

- ³⁵₁₇ Objetivos: Completar el análisis de tareas comenzada en el plan de trabajo, el diseño y prototipo.
- ³⁵₁₇ Entregables: Documentación que incluye perfiles de usuario, análisis de tareas, diagrama de clases, diagramas de casos de uso, diagramas de componentes y el prototipo.

La entrega de esta documentación facilitará cumplir los objetivos finales del proyecto, ya que se estudian los potenciales clientes del proyecto, se analizan todas las tareas detalladamente y se obtiene un prototipo de alta fidelidad del proyecto.

Todas estas tareas sirven como entrada de datos para la siguiente fase.

1.4.4- Desarrollo J2EE - PAC 3

Esta tarea comprende las siguientes actividades:

Objetivos: Implementar la solución del proyecto y documentación complementaria.

Entregables: Código fuente, script de inicio, script de creación de tablas y documentación complementaria.

Es la fase final del proyecto en la que se obtiene el producto final. Puede requerir iterar con las fases anteriores en caso de detectar problemas en la elaboración del producto.

1.4.5- Entrega final: Elaboración Memoria, EAR y Presentación TFC

Objetivos: Finalizar el proyecto y documentarlo.

Entregables: Memoria y power point de presentación del proyecto.

En esta fase se finalizará la fase anterior en caso de no haberlo hecho y se presentará nuestro producto al tribunal.

Esta tarea comprende las siguientes actividades:

³⁵₁₇ Elaboración de la memoria de este TFC

³⁵₁₇ Fichero EAR o WAR

³⁵₁₇ Elaboración de la presentación de este TFC.

³⁵₁₇ Entrega de la memoria y presentación de este TFC.

1.5.- Planificación.

El proyecto comenzó el día 20/09/2014 y finalizará el día 13/01/2014.

Para realizar la planificación completa del diagrama de Gantt se ha utilizado una aplicación libre como es OppenProj.

La distribución detallada de las tareas, su duración estimada, así como su plazo temporal se puede apreciar en el siguiente punto:

1.5.1 - DIAGRAMA DE GANTT

A continuación se mostrará el diagrama de Gantt realizado con el software especializado Openproj:

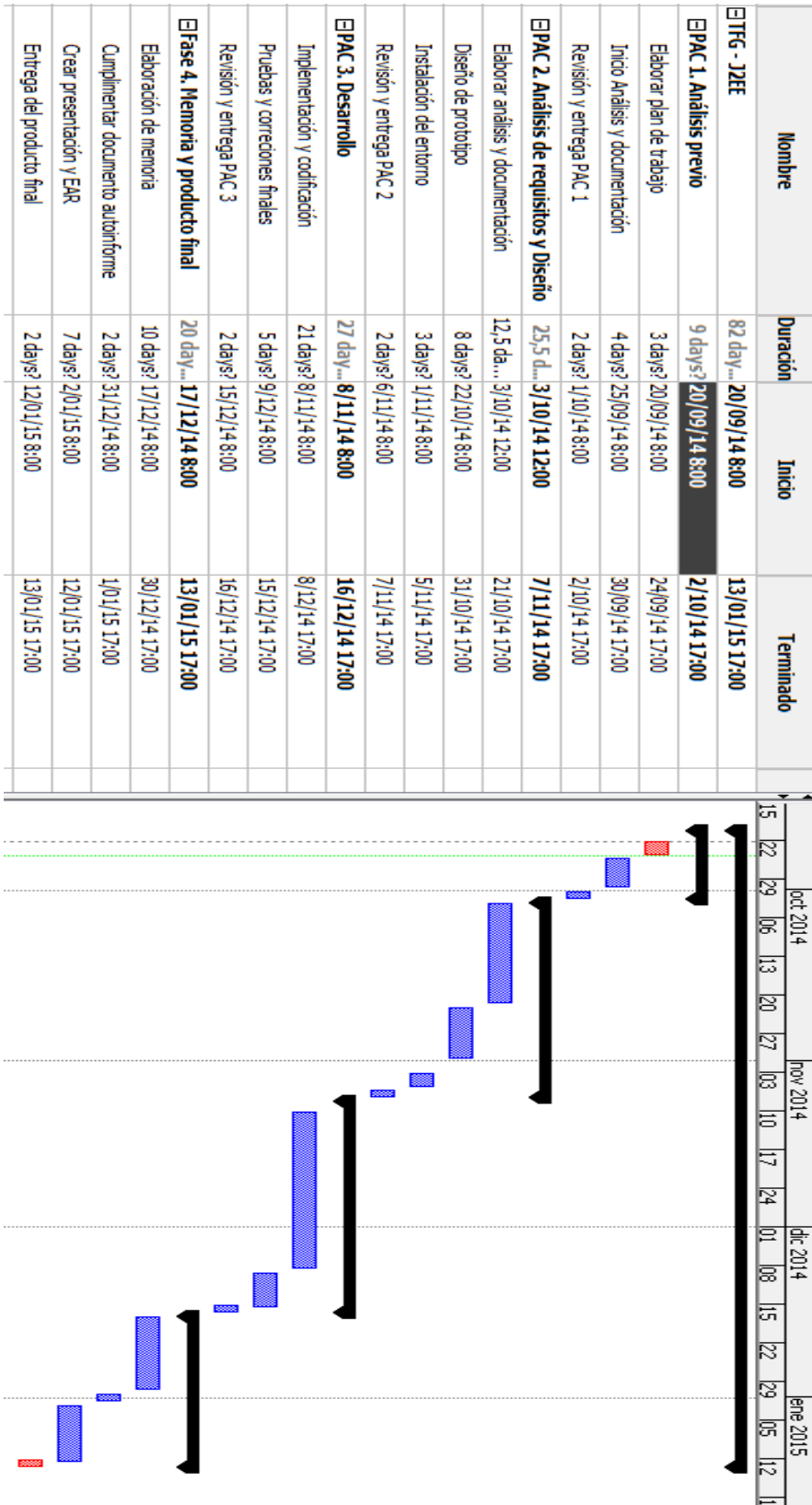


Figura 2. Diagrama de gantt

1.5.2 - FECHAS CLAVES DEL PROYECTO

A continuación se muestra la relación de fechas claves del proyecto en la siguiente tabla:

<i>Resumen Temporización</i>	
Inicio	19/09/14
Entrega elanificación TFC (PAC 1)	02/10/14
Entrega Análisis Funcional y Diseño técnico (PAC 2)	07/11/14
Entrega Implementación GestAquarium (PAC 3)	16/12/14
Entrega Memoria y Presentación TFC (ENTREGA FINAL)	13/01/14
Fecha finalización	24/01/14

Tabla 1. Resumen de temporización

1.5 Productos obtenidos

Como resultado final, se han obtenido los siguientes productos:

Memoria_MenachoGarcia_Francisco.pdf, que es la presente memoria.

Presentacion_MenachoGarciaFrancisco.zip, que contiene la presentación virtual del TFC en LibreOffice Impress.

Producto_MenachoGarcia_Francisco.zip, que es la aplicación web resultante y que contiene el archivo .ear para poder importarlo directamente al entorno de desarrollo integrado y el scripts sql de inicio para el administrador.

1.6 Breve descripción de los otros capítulos de la memoria

En los próximos capítulos de la memoria, se comentaran las fases de análisis funcional, diseño técnico e implementación, que se han llevado a cabo para desarrollar la aplicación.

Habrà un capítulo donde se detallará la valoración económica del proyecto. Seguidamente, en otro capítulo se

plasmarán las conclusiones personales donde se hará una justificación y razonamiento de las decisiones adoptadas durante el proyecto. Se especificarán qué experiencias o conclusiones he extraído de este TFC.

Y, finalmente, otros tres capítulos más dedicados a líneas futuras de la aplicación, glosario de la memoria y bibliografía consultada.

Capítulo 2. Análisis funcional

En este capítulo se realiza un análisis teórico del sistema desarrollado. Se hace una primera descripción del funcionamiento del sistema y a continuación se detallan tanto los requisitos funcionales del sistema como los no funcionales.

2.1 Descripción del funcionamiento del sistema

La aplicación que se ha planteado construir en este proyecto pretende cubrir la necesidad del control de gestión de producto y pedidos para la empresa *Aqua Net*.

La aplicación debe permitir gestionar productos, tales como Peces exóticos, accesorios o acuarios. Además debe permitir acceder con dos tipos de perfiles:

- Perfil Administrador
- Perfil Usuario registrado

Se ha tenido en cuenta el rol de cada usuario para controlar qué acciones se puede hacer sobre el sistema.

Un usuario registrado no debe tener acceso a la gestión de productos de la aplicación. Por eso, se ha implantado un sistema de control de usuarios para la aplicación.

Por último, también se ha controlado la seguridad en el acceso a la aplicación mediante autenticación de usuarios. De esta manera, ingresando una URL de cualquier página de acción de la aplicación ésta avisará al usuario de que no tiene el perfil o rol suficiente para poder acceder a esa opción de menú.

Hecha esta breve descripción del sistema se analizará a continuación más en detalle los requisitos.

Existen dos tipos de requisitos:

- **Funcionales.** Describen el comportamiento que ha de adoptar el sistema. Que acciones ha de desarrollar, como las debe llevar a cabo y qué resultado se debe dar para cada una de ellas.
- **No funcionales.** Especifican cómo se debe comportar el sistema en términos de rendimiento, costo, escalabilidad... Son por tanto requisitos que no entran en la funcionalidad, pero son imprescindibles para construir una aplicación efectiva.

2.2. Análisis de requisitos funcionales

La aplicación está compuesta por diferentes subsistemas. Para analizar los requisitos funcionales se detallará cada uno de estos subsistemas mediante la explicación de lo que ofrecen, acompañados de sus casos de uso y las especificaciones de estos últimos.

- Gestión de Peces exóticos (alta, baja, modificación y consultas)
- Gestión de Acuarios alta, baja, modificación y consultas)
- Gestión de Accesorios exóticos alta, baja, modificación y consultas)
- Registrar usuario
- Identificarse en el sistema

- Log out sistema
- Realizar Pedido (Alta del pedido y consultas)

2.2.1- Casos de uso y sus Fichas

Diagrama general de los casos de uso

En el siguiente diagrama se visualizan los casos de usos principales que componen la aplicación. En el cual se puede apreciar que el diagrama está dividido en 5 "paquetes": Exotic Fish, Accessory, Aquarium, Customer y Order, los cual explicaremos debidamente en las fichas de casos de uso.

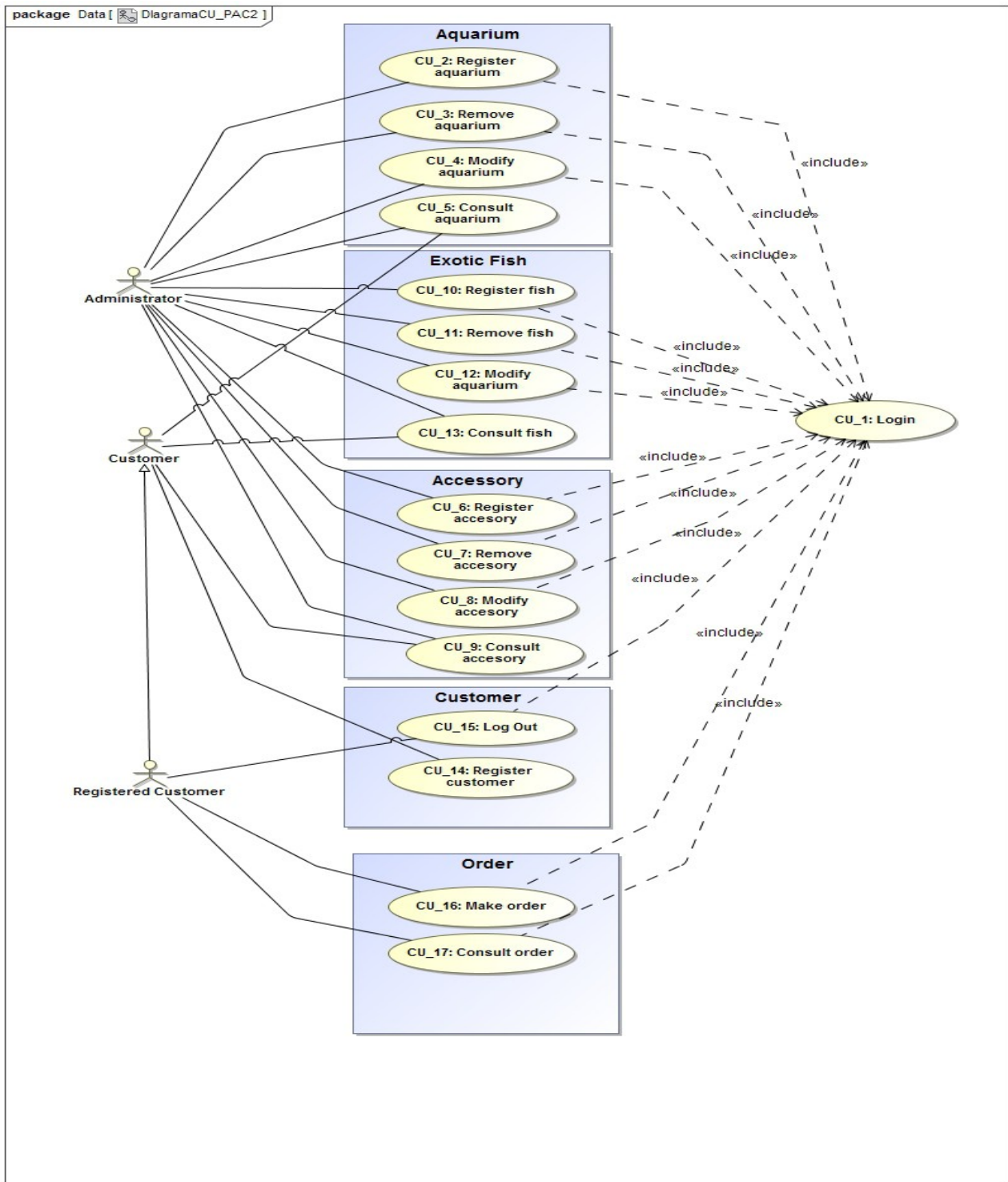


Figura 3. Diagrama de casos de uso

2.2.2 Fichas de gestión de casos de uso

CU 1: LOGIN	
Descripción	Permite a un usuario del sistema identificarse en el mismo para poder acceder con unos permisos en concreto.
Actores	Administrador, Usuario registrado
C.U Relacionados	--

Pre Condición	<i>No estar identificado en el sistema</i>
Post Condición	--
Proceso	<ul style="list-style-type: none"> • <i>Estar posicionado en la página principal de la aplicación.</i> • <i>Introducir usuario y password.</i> • <i>Pulsar sobre el botón aceptar.</i>
Alternativas y excepciones	<ul style="list-style-type: none"> • <i>Si actualmente está identificado en el sistema no podrá identificarse de nuevo.</i> • <i>En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario.</i>

Tabla 2. Caso de uso de la acción login

CU 2: REGISTER AQUARIUM	
Descripción	<i>A través de este caso de uso se permite la funcionalidad de alta de un acuario en el sistema con una cualidades específicas del mismo.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de acuarios exóticos a través del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los acuarios dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el botón de "Alta acuario", situado en la parte superior de la tabla.</i></p> <p><i>Cumplimentar el formulario con los datos a dar de alta.</i></p> <p><i>Pulsar sobre el botón "Alta".</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • <i>En caso de intentar dar de alta un acuario sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta.</i> • <i>En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario.</i>

Tabla 3. Caso de uso para añadir un acuario

CU 3: REMOVE AQUARIUM	
Descripción	<i>A través de este caso de uso se permite dar de baja un acuario en el sistema.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<i>En primer lugar debe entrar en la sección de gestión de acuarios exóticos a través del enlace o imagen del mismo.</i>

	<p><i>A continuación se mostrará un listado de los acuarios dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el icono "eliminar", situado en una columna de la tabla.</i></p> <p><i>Pedirá confirmación de eliminación y en caso de aceptar se eliminará dicho acuario.</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • En caso de intentar eliminar un acuario tenga asociado un pedido avisará de que actualmente tiene pedidos y no será eliminado. • En el caso de pulsar sobre cancelar en la ventana emergente de eliminación no se eliminará y volverá al listado de acuarios.

Tabla 4. Caso de uso para eliminar un acuario

CU 4: MODIFY AQUARIUM	
Descripción	<i>A través de este caso de uso se permite la funcionalidad de modificar un acuario en el sistema con una cualidades específicas del mismo.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	<i>--</i>
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de acuarios exóticos através del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los acuarios dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el icono "modificación", situado en una columna de la tabla.</i></p> <p><i>Mostrará el acuario a modificar con los datos cargados en el formulario .</i></p> <p><i>Modificar los campos que sean necesarios</i></p> <p><i>Pulsar sobre el botón "Modificar".</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • En caso de intentar modificar un acuario sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta. • En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario.

Tabla 5. Caso de uso para modificar un acuario

CU 5: CONSULT AQUARIUM	
Descripción	<i>A través de este caso de uso se permite consultar los acuarios dados de alta en el sistema.</i>
Actores	<i>--</i>
C.U Relacionados	<i>--</i>
Pre Condición	<i>--</i>
Post Condición	<i>--</i>
Proceso	<i>En primer lugar debe entrar en la sección de gestión de acuarios exóticos através</i>

	<p>del enlace o imagen del mismo.</p> <p>Mostrará el listado de acuarios.</p>
Alternativas y opciones	--

Tabla 6. Caso de uso para consultar un acuario

CU 10: REGISTER FISH	
Descripción	<i>A través de este caso de uso se permite la funcionalidad de alta de un pez exótico en el sistema con una cualidades específicas del mismo.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de peces exóticos a través del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los peces dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el botón de "Alta Pez", situado en la parte superior la tabla.</i></p> <p><i>Cumplimentar el formulario con los datos a dar de alta.</i></p> <p><i>Pulsar sobre el botón "Alta".</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • En caso de intentar dar de alta un pez exótico sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta. • En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario.

Tabla 7. Caso de uso para añadir un pez exótico

CU 11: REMOVE FISH	
Descripción	<i>A través de este caso de uso se permite dar de baja un pez en el sistema.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de peces exóticos a través del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los peces exóticos dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el icono "eliminar", situado en una columna de la tabla.</i></p> <p><i>Pedirá confirmación de eliminación y en caso de aceptar se eliminará dicho pez.</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • En caso de intentar eliminar un pez que tenga asociado un pedido avisará de que actualmente tiene pedidos y no será eliminado. • En el caso de pulsar sobre cancelar en la ventana emergente de eliminación no se eliminará y volverá al listado de peces exóticos.

Tabla 8. Caso de uso para eliminar un pez exótico

CU 12: MODIFY FISH	
Descripción	<i>A través de este caso de uso se permite la funcionalidad de modificar un pez exótico en el sistema con una cualidades específicas del mismo.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de peces exóticos a través del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los peces dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el icono "modificación", situado en una columna de la tabla.</i></p> <p><i>Mostrará el pez exótico a modificar con los datos cargados en el formulario .</i></p> <p><i>Modificar los campos que sean necesarios</i></p> <p><i>Pulsar sobre el botón "Modificar".</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • En caso de intentar modificar la información de un pez exótico sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta. • En caso de cumplimentar el formulario de algún campo numérico con letrás saltará el validador avisando al usuario.

Tabla 9. Caso de uso para modificar un pez exótico

CU 13: CONSULT FISH	
Descripción	<i>A través de este caso de uso se permite consultar los peces exóticos dados de alta en el sistema.</i>
Actores	--
C.U Relacionados	--
Pre Condición	--
Post Condición	--
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de peces exóticos a través del enlace o imagen del mismo.</i></p> <p><i>Mostrará el listado de peces exóticos.</i></p>
Alternativas y opciones	--

Tabla 10. Caso de uso para consultar un pez exótico

CU 6: REGISTER ACCESSORY	
Descripción	<i>A través de este caso de uso se permite la funcionalidad de alta de un accesorio en el sistema con una cualidades específicas del mismo.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	<i>--</i>
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de accesorios exóticos através del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los accesorios dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el botón de "Alta Accesorio", situado en la parte superior de la tabla.</i></p> <p><i>Cumplimentar el formulario con los datos a dar de alta.</i></p> <p><i>Pulsar sobre el botón "Alta".</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • <i>En caso de intentar dar de alta un accesorio exótico sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta.</i> • <i>En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario.</i>

Tabla 11. Caso de uso para añadir un accesorio

CU 7: REMOVE ACCESSORY	
Descripción	<i>A través de este caso de uso se permite dar de baja un ccesorio exótico en el sistema.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	<i>--</i>
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de ccesorios exóticos a través del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los accesorios exóticos dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el icono "eliminar", situado en una columna de la tabla.</i></p> <p><i>Pedirá confirmación de eliminación y en caso de aceptar se eliminará dicho accesorio.</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • <i>En caso de intentar eliminar un ccesorio exótico que tenga asociado un pedido avisará de que actualmente tiene pedidos y no será eliminado.</i> • <i>En el caso de pulsar sobre cancelar en la ventana emergente de eliminación no se eliminará y volverá al listado de accesorios exóticos.</i>

Tabla 12. Caso de uso para eliminar un accesorio

CU 8: MODIFY ACCESSORY	
Descripción	<i>A través de este caso de uso se permite la funcionalidad de modificar un accesorio</i>

	<i>en el sistema con una cualidades específicas del mismo.</i>
Actores	<i>Administrador</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	<i>--</i>
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de accesorios exóticos a través del enlace o imagen del mismo.</i></p> <p><i>A continuación se mostrará un listado de los ccesorios exóticos dados de alta en el sistema (en caso de haberlos) y se pulsará sobre el icono "modificación", situado en una columna de la tabla.</i></p> <p><i>Mostrará el ccesorio a modificar con los datos cargados en el formulario .</i></p> <p><i>Modificar los campos que sean necesarios</i></p> <p><i>Pulsar sobre el botón "Modificar".</i></p>
Alternativas y excepciones	<ul style="list-style-type: none"> • <i>En caso de intentar modificar la información de un accesorio sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta.</i> • <i>En caso de cumplimentar el formulario de algún campo numérico con letrás saltará el validador avisando al usuario.</i>

Tabla 13. Caso de uso para modificar un accesorio

CU 9: CONSULT ACCESSORY	
Descripción	<i>A través de este caso de uso se permite consultar los accesorios dados de alta en el sistema.</i>
Actores	<i>--</i>
C.U Relacionados	<i>--</i>
Pre Condición	<i>--</i>
Post Condición	<i>--</i>
Proceso	<p><i>En primer lugar debe entrar en la sección de gestión de accesorios exóticos a través del enlace o imagen del mismo.</i></p> <p><i>Mostrará el listado de ccesorios exóticos.</i></p>
Alternativas y opciones	<i>--</i>

Tabla 14. Caso de uso para consultar un accesorio

CU 14: REGISTER CUSTOMER	
Descripción	<i>A través de este caso de uso se permite dar de alta a un cliente para poder realizar pedidos sobre acuarios, accesorios, peces exóticos, etc</i>
Actores	<i>Customer</i>
C.U Relacionados	<i>--</i>
Pre Condición	<i>--</i>
Post Condición	<i>--</i>
Proceso	<i>En la página inicial de la aplicación pulsar sobre el enlace "Registrar usuario".</i>

	<p><i>Cumplimentar el formulario al que da acceso el paso anterior.</i></p> <p><i>Activar casilla de entiendo términos legales.</i></p> <p><i>Pulsar sobre el botón "Registrar".</i></p>
Alternativas y opciones	<ul style="list-style-type: none"> • En caso de intentar dar de alta un cliente sin cumplimentar los campos obligatorios, avisará al usuario de la información que falta. • En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario. • En caso de intentar dar de alta un usuario sin haber pulsado sobre "Acepto términos legales" avisará al usuario que no es posible realizar el registro hasta que acepte los términos legales.

Tabla 15. Caso de uso para añadir un usuario registrado

CU 15: LOG OUT	
Descripción	<i>A través de este caso de uso se permite desconectar del sistema en caso de estar logado en el mismo. De este forma nos aseguramos de borrar toda la información del usuario del sistema.</i>
Actores	<i>Adminsitrator, Registered Customer</i>
C.U Relacionados	--
Pre Condición	--
Post Condición	--
Proceso	<p><i>Pulsar sobre el icono "Log out" situado en la parte superior derecha de cualquier página.</i></p> <p><i>El sistema volverá a la pantalla inicial de login.</i></p>
Alternativas y opciones	--

Tabla 16. Caso de uso para salir de la sesión

CU 16: MAKE ORDER	
Descripción	<i>Permite a un usuario realizar pedidos de los productos pertenecientes a la tienda de peces exóticos.</i>
Actores	<i>Registered Customer</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<p><i>Entrar en la sección en el que se desea realizar un pedido (peces exóticos, accesorios exóticos, acuarios...)</i></p> <p><i>A continuación se mostrará un listado de producto dados de alta en el sistema (en caso de haberlos) .</i></p> <p><i>Pulsar sobre el icono "Realizar pedido" situado en una de las columnas de la tabla del producto seleccionado.</i></p> <p><i>Mostrará el formulario de pedido el cual se deberá cumplimentar adecuadamente.</i></p> <p><i>Pulsar sobre el botón "Realizar Pedido".</i></p>

Alternativas y opciones	<ul style="list-style-type: none"> • En caso de intentar realizar un pedido de un producto sin cumplimentar los campos obligatorios, avisará al usuario de la información que necesaria. • En caso de cumplimentar el formulario de algún campo numérico con letras saltará el validador avisando al usuario.
--------------------------------	---

Tabla 17. Caso de uso para realizar un pedido

CU 17: CONSULT ORDER	
Descripción	<i>Permite a un usuario realizar consultas sobre los pedidos que ha realizado y el estado de los mismo..</i>
Actores	<i>Registered Customer</i>
C.U Relacionados	<i>Login</i>
Pre Condición	<i>Estar logado en el sistema</i>
Post Condición	--
Proceso	<i>Entrar en la sección de gestión de pedidos .</i> <i>Mostrará un listado de los pedidos de productos realizados por el usuario (en caso de haberlos) .</i>
Alternativas y opciones	--

Tabla 18. Caso de uso para consultar los pedidos

2.3. Análisis de requisitos no funcionales

Éstos son requisitos que especifican criterios que pueden usarse para juzgar la operación del sistema en lugar de sus comportamientos específicos, es decir, de sus requisitos funcionales.

Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.

2.3.1 Calidad

Para el correcto funcionamiento de la aplicación es necesario que ésta se mueva en unos márgenes de calidad adecuados. Analizamos algunos de los más importantes:

- **Integridad y seguridad.** Es necesario asegurar que personas ajenas al programa no puedan acceder a él. Para aquellos usuarios que tengan privilegios de acceso hay que corroborar que sólo tengan cobertura en aquellas partes del aplicativo donde los permisos de su rol tengan vigencia. Además los datos del programa deben ir cifrados para dar seguridad al protocolo http.

- **Flexibilidad y portabilidad.** Como estamos hablando de una aplicación genérica, debe ser fácilmente adaptable a nuevos escenarios. Por tanto, el software debe ser modificable y ampliable. Además debe ser portable para poder aprovechar partes de la infraestructura del destinatario, como por ejemplo la base de datos.
- **Interoperabilidad.** Será posible unir el programa a otros, por ejemplo mediante Web Services, sin grandes cambios en el sistema.

2.3.2 Coste

Se presenta el coste como un factor importante del desarrollo. Se desea construir una aplicación cuyo coste sea bajo, tanto a la hora de implementarla, como a la hora de mantenerla. En este sentido será necesario encontrar tecnologías de software libre para lograr este propósito.

3.6 Base de Datos PostgreSQL

Para la creación y administración de la base de datos, se utiliza la herramienta propia del PostgreSQL, como es pgAdminIII.

Como norma general, PostgreSQL ofrece una integridad de datos más fuerte que MySQL. Además, numerosos tipos de datos son igualmente a su disposición (tablas, XML, etc). Puede incluso crear a sus propios tipos de datos, o crear sus propias funciones para construir peticiones específicas.

Módulos externos le permiten disponer de tipos de datos preparados para expertos, y PostgreSQL le ofrece igualmente un motor de indexación potente y evolutivo, que le permite acelerar las peticiones más frecuentes a sus aplicaciones...

Este tipo de SGBDR está particularmente adaptado a los cálculos científicos y estadísticas, a aplicaciones financieras o a la gestión fina de las transacciones.

3.7 Modelo-Vista-Controlador (MVC)

Las aplicaciones Web pueden ser implementadas de muchas maneras. No se usaría la misma estrategia para desarrollar una aplicación de dos pantallas que una de cien, ni sería conveniente construir de la misma forma una aplicación con una base de datos pequeña que con una grande. Es por ello que una interesante forma de trabajar es adaptar la aplicación a un patrón que se adapte a nuestras necesidades.

Para aplicaciones de pequeña envergadura y siempre teniendo en cuenta que queremos un comportamiento dinámico, lo más útil sería usar un patrón donde las pantallas reflejen directamente el modelo de base de datos. Sin embargo, si queremos muchas pantallas, tenemos una lógica de negocio abundante y la base de datos no es simple, conviene estructurar la aplicación de otra manera.

En caso contrario corremos el riesgo de montar un sistema muy complejo e imposible de mantener. En este sentido, el patrón modelo-vista-controlador empleado en el proyecto es una buena manera de modular la aplicación, separando los datos de la aplicación, de la interfaz del usuario y de la lógica de negocio. Esta forma de trabajar provoca que la configuración sea más extensa pero a nivel conceptual es muy intuitiva y permite diferenciar bien los componentes y actuar sobre ellos de una manera más precisa. Analizamos a continuación un poco más los tres grandes bloques que define este patrón.

- **Modelo.** Es el módulo que representa los datos del programa. Se encarga de manipular estos datos de forma coherente y ofreciéndolos a la aplicación a medida que los va requiriendo. Esta capa no debe tener conocimiento ninguno ni del controlador ni de las vistas.

- **Vista.** Es la representación visual de los datos. Serán en sí las páginas que el usuario contemple y es la capa mediante este interactuará con la aplicación. Aunque a menudo los datos que se representen son el reflejo de la base de datos se hará uso de unos objetos llamados bean para establecer la relación pasando siempre por la capa controlador.

- **Controlador.** Es el tercero de los módulos. El que

proporciona significado a las órdenes del usuario y el que en definitiva crea la relación entre modelo y vista.

Esta capa será la encargada de decidir cuando se debe modificar la base de datos y cuando se deben obtener datos de ella para que los disponga la vista. Además se encarga de la lógica de redireccionamiento de pantallas.

En la siguiente imagen podemos observar el ciclo que realiza una aplicación basada en el modelo-vista-controlador.

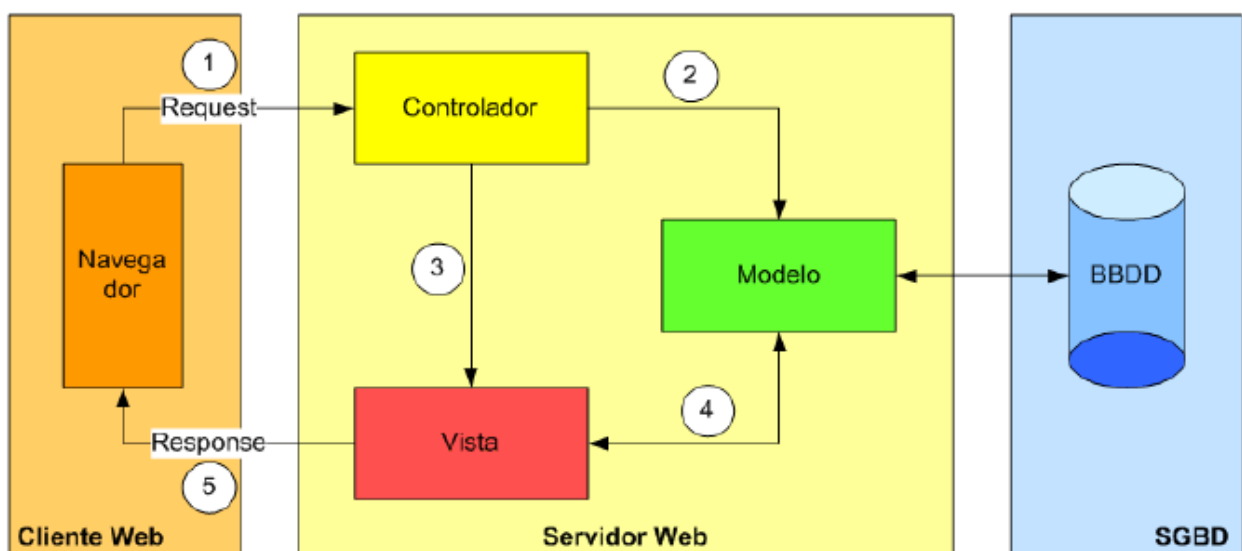


Figura 4. Arquitectura Modelo-Vista-Controlador (MVC)

En primer lugar el usuario debe realizar algún tipo de petición (*request*), es decir, alguna acción en el navegador. El controlador será el responsable de capturar esa petición y de realizar diferentes operaciones. Estas operaciones muy posiblemente necesiten obtener y modificar información de la base de datos por lo que se hará uso de la capa modelo.

Una vez el controlador lleve a cabo sus acciones deberá escoger la pantalla a la que se debe redirigir la petición. Será necesaria la capa vista para ofrecer al usuario los datos de una forma estructurada. El resultado será devuelto (*response*) al cliente para que el usuario pueda de nuevo iniciar el ciclo.

Las principales ventajas que proporciona trabajar con un patrón como este son la separación de los datos de la

base de datos de su representación gráfica y la simpleza a la hora de añadir y modificar nuevas vistas para que se adapten según a las necesidades del modelo. Otras características que ofrece este patrón son la independencia entre las tres capas, la escalabilidad que proporciona al poder actuar en cualquiera de las tres capas y la separación de funciones que facilita el mantenimiento. En contra de este patrón se debe destacar que a veces no es simple separar las capas y definir qué tecnología se usa para cada propósito.

Además trabajar con este método provoca que exista un número más elevado de archivos

3.- Diagrama de clases principal

3.1 Diagrama de clases

En este apartado se puede ver un diagrama con la representación de las clases para el modelo de producto del prototipo.

Los diferentes productos se encontrarán clasificados (tal como ExoticFish, Aquarium, y Accessory) serán gestionados por el usuario administrador. También se puede apreciar como sólo los usuarios registrados pueden realizar pedidos de los productos dados de alta en el sistema. Los pedidos tendrán asociado el producto del pedido en cuestión y el usuario que realiza el pedido junto con un identificador propio del pedido.

Aunque quizás se haya podido modelar a los productos de la tienda como una herencia, no lo he creído conveniente ya que no he visto apenas propiedades suficientes para realizarla. Además para la gestión del mismo me ha resultado más sencillo de utilizar.

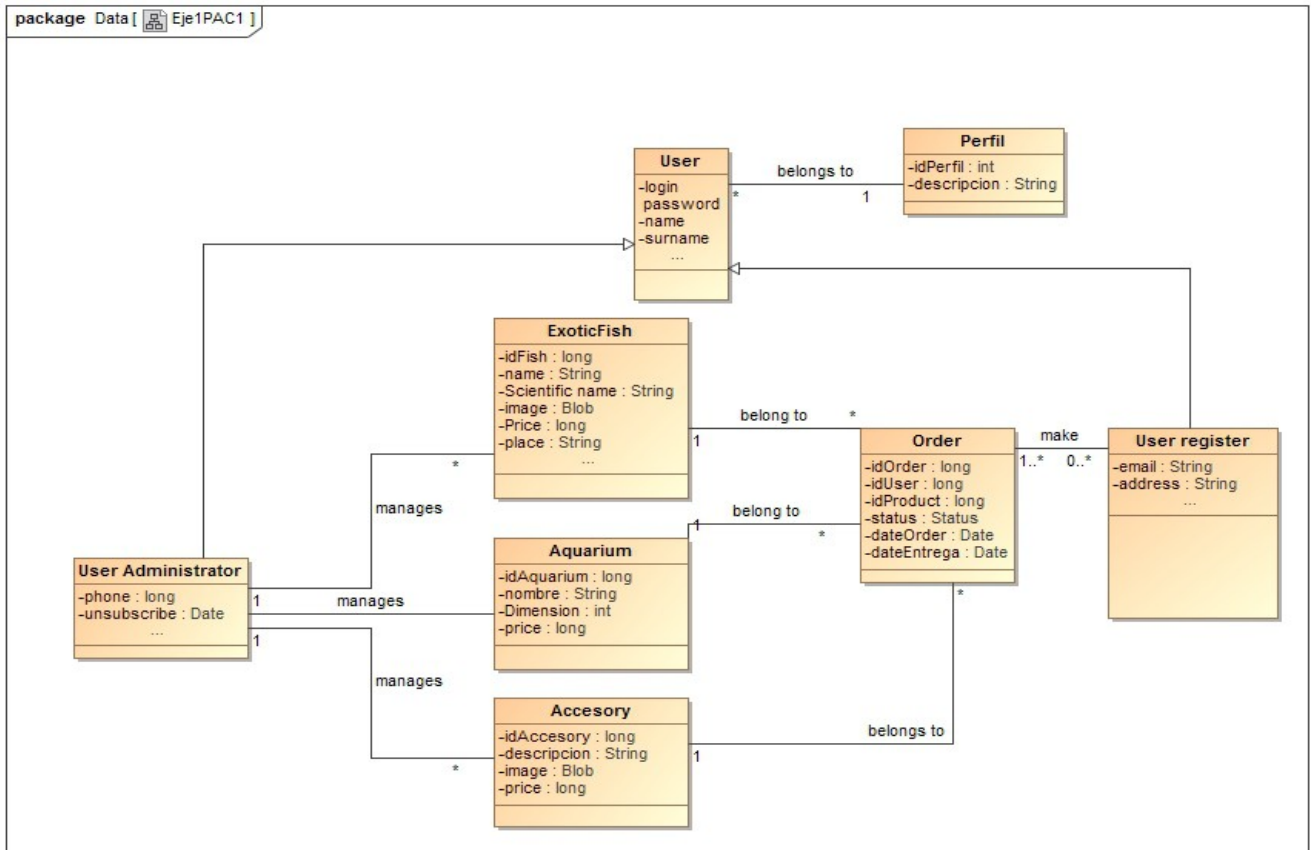


Figura 5. Diagrama de clases

4.- Diagrama relacional de base de datos

En el siguiente esquema de BD relacional se puede apreciar las tablas pertenecientes a la aplicación de GestAquarium. Se ha modelado la herencia de los usuarios con JPA en una sola tabla ya que a nivel lógico de clases es mejor con nua herencia se ha comprobado que a nivel relacional es más eficiente en una sola tabla.

La tabla princial en este caso es orders que es dónde se guardan los pedidos de los usuarios registrados en la aplicación. Alrededor de ellas podemos encontrar las tablas que se relacionan como los accesorios, acuarios, peces exóticos y usuarios.

También se puede observar una tabla Perfil la cual está

vinculada con la tabla usuario. De esta forma podremos saber si el usuario es de tipo administrador, de tipo usuario registrado o simplemente no tiene perfil. Con esto, podremos saber a qué opciones del menú se puede acceder.

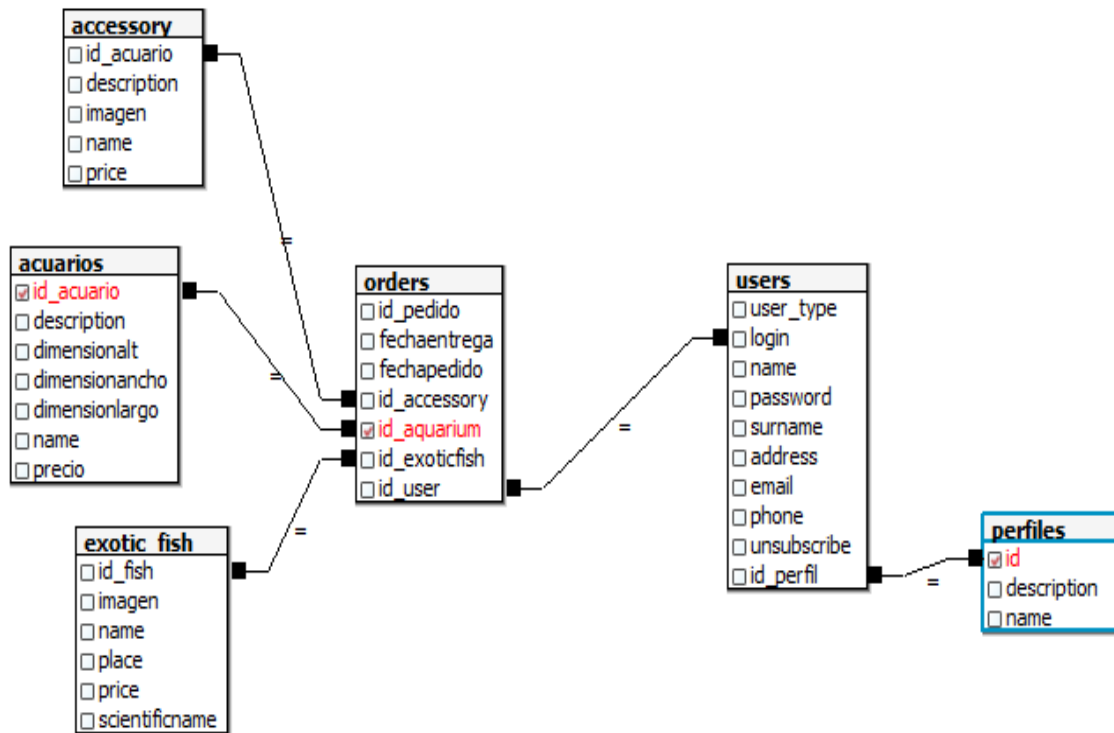


Figura 6. Diagrama de base de datos

5.- Diagrama de estados

El prototipo que se está desarrollando no cuenta con ningún proceso especialmente complejo de diferentes estados. No obstante, es necesario tener claro como se comporta la administración y gestión de algunas entidades. En el siguiente diagrama de estados se puede observar con mas claridad estos puntos:

5.1 DIAGRAMA DE ESTADOS PARA PECES EXÓTICOS

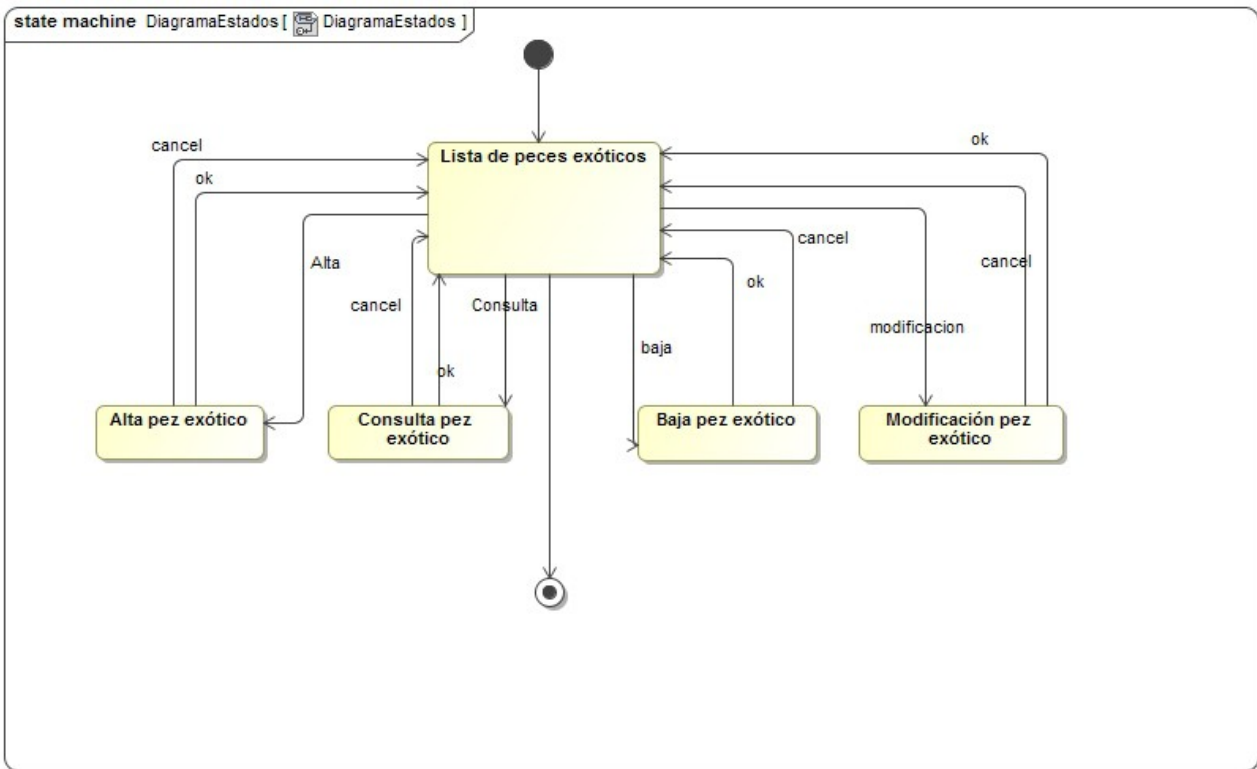


Figura 7. Diagrama de estados peces exóticos

5.2.- DIAGRAMA DE ESTADOS PARA ACCESORIOS

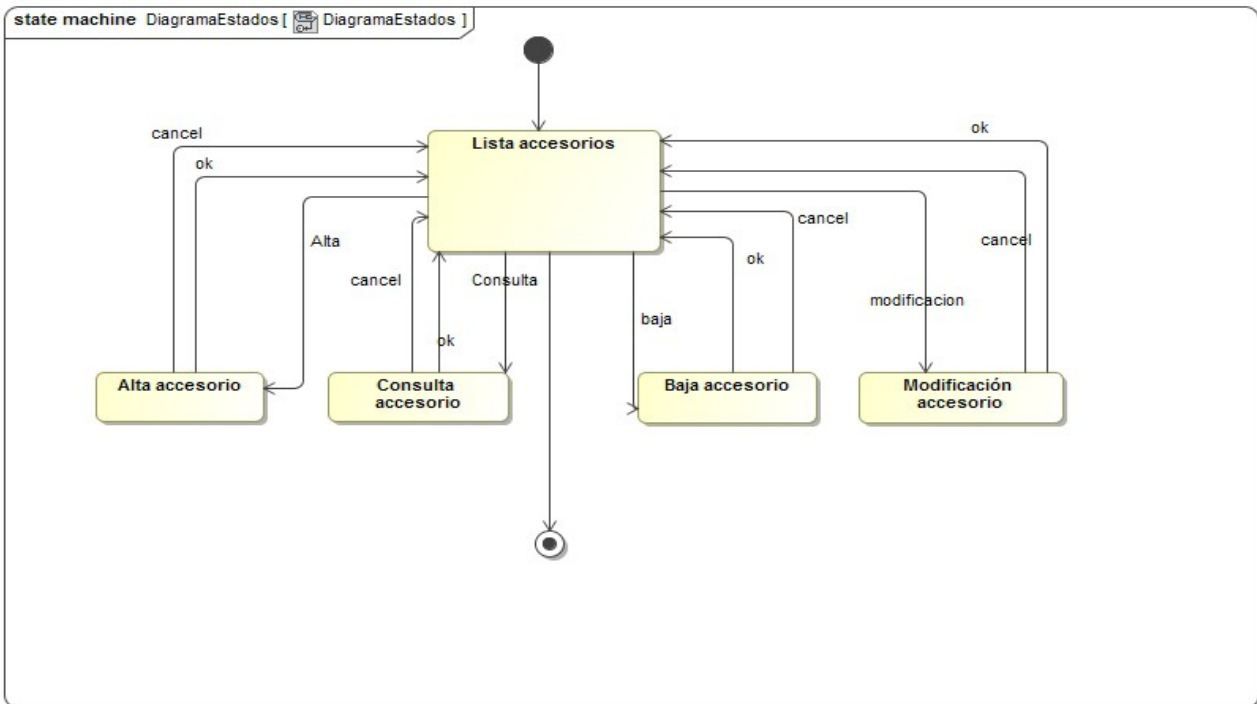


Figura 8. Diagrama de estados de accesorios

5.3 DIAGRAMA DE ESTADOS PARA ACUARIOS

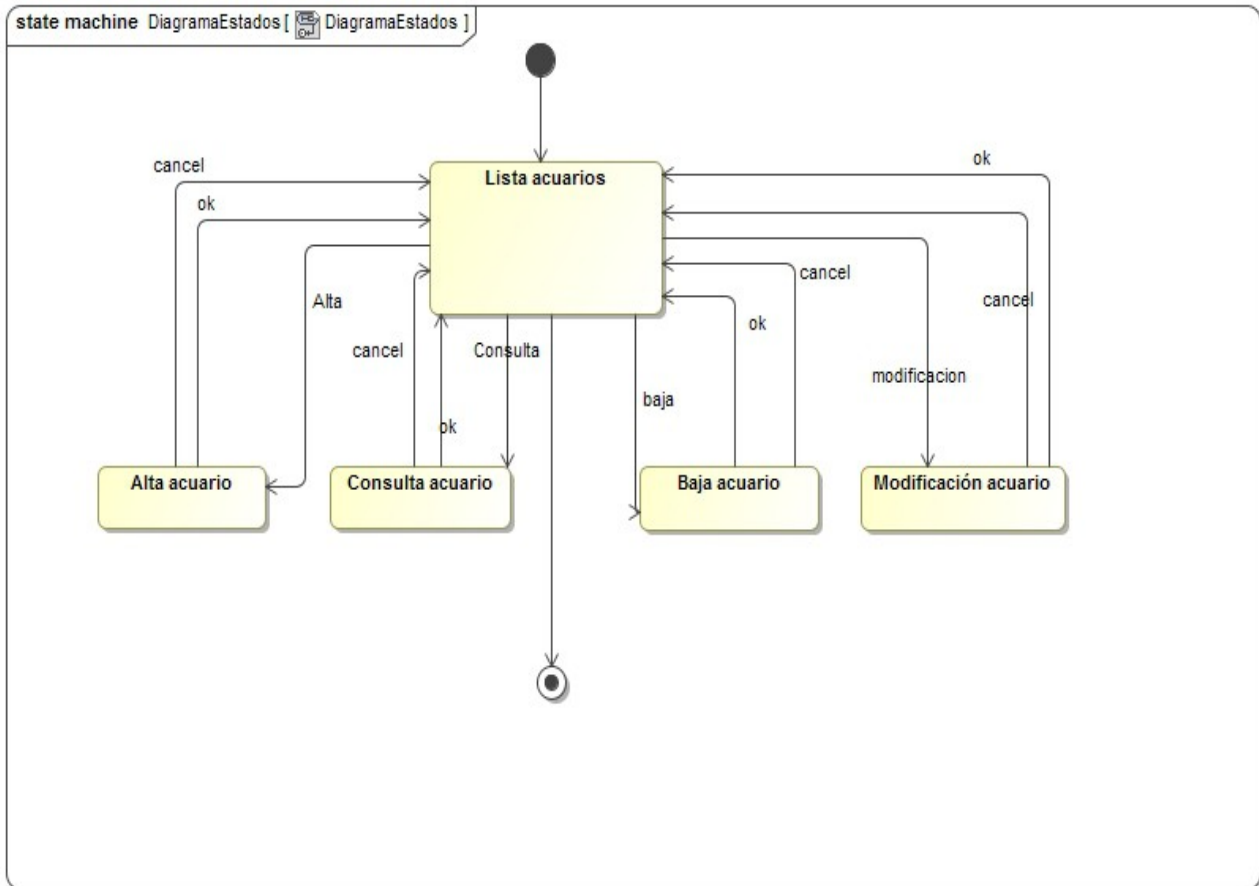


Figura 9. Diagrama de estados de acuarios

6.- Prototipos de las pantallas

En principio, las pantallas que veremos a continuación son las maquetas que se ajustarán lo máximo posible a la realidad de la aplicación. En caso de ser necesario podrán ser modificadas si así lo requiere las necesidades del proyecto GestAquariums.

6.1- Pantalla de Acceso

Esta es la pantalla inicial de la aplicación. Al entrar en ella lo único que podremos ver la pantalla de identificación para los usuarios (tanto administradores, como usuarios registrados).

Además también tendrá la posibilidad de registrarse un usuario que no lo haya hecho con anterioridad a través del botón "Registrar". Hay que tener en cuenta que la

acción registrar solo puede realizar registro de usuarios registrados y nunca de usuarios administradores ya que estos se realizarán a través de una carga inicial mediante script de BD.



Figura 10. Pantalla de acceso

6.2.- Pantalla de registro

Al pulsar en la pantalla inicial anterior sobre el botón registrar, ira al formulario de registro de usuarios. Como se ha comentado con anterioridad, solo dará el alta de usuarios registrados y no de administrador. Los campos estarán validados, y no dejará dar el alta de ningún usuario que no haya cumplimentado debidamente todos los datos del formulario.

Entrar

Registrarse

Usuario:	<input type="text"/>
Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Contraseña:	<input type="text"/>
Email:	<input type="text"/>
Dirección:	<input type="text"/>

Aceptar

Return

- Campo id: Error de validación: se necesita un valor.
- Campo nombre: Error de validación: se necesita un valor.
- Campo apellido: Error de validación: se necesita un valor.
- Campo contraseña: Error de validación: se necesita un valor.
- Campo email: Error de validación: se necesita un valor.
- Campo dirección: Error de validación: se necesita un valor.

Figura 11. Pantalla de registro de usuarios

6.3.- Pantalla del administrador

Al iniciar sesión con un usuario administrador se habilitarán acciones que el usuario registrado no podrá ver, o no podrá ver en su totalidad.

Usuario: Francisco Menacho Garcia

LogOut

Gestión peces exóticos | Gestión de accesorios | Gestión de acuarios

Peces Exóticos

Accesorios

Acuarios

Figura 12. Pantalla del administrador

6.4.- Gestión de peces exóticos

Al acceder con el perfil administrador, en cualquier gestión de productos habilitados, veremos las opciones de Alta del producto, modificación, eliminación o consulta del mismo.

Para que no sea muy engorrosos, se mostrarán estas principales acciones en detalle pero solo para la gestión de peces exóticos.

PECES EXÓTICOS				
id	Nombre	Nombre científico	Precio	Origen
160	Pez angel naranaja2	PEcius naranju2	32	España
165	Pez naranjita	Pecijs naranjitus	7	América

Figura 13. Pantalla de gestión de peces exóticos

6.5.- Alta de pez exótico

Al pulsar en la pantalla anterior en Añadir pez exótico nos llevará a esta otra pantalla en la cual podremos dar de alta el pez exótico.

En ella se puede observar los campos disponibles para dar de alta:

Nombre: Nombre vulgar del pez exótico

Nombre científico: Nombre científico del pez exótico

Precio: Precio de venta

Origen: Origen del pez exótico

Imagen: Imagen del pez

Es importante ver que se ha indicado el campo imagen a parte del formulario porque a la hora de realizar el desarrollo es necesario por cuestiones técnicas que la

codificación del formulario sea diferente al de los otros campos.

Es muy importante realizar la carga de la imagen antes de cumplimentar los demás datos ya que al cargar la imagen reseteará los campos. Por lo que primero cargar la imagen eligiéndola y pulsando en el botón adjuntar, después cumplimentar los datos y finalmente pulsar sobre 'Alta Categoría' para dar de alta al pez exótico.

Nombre:	Pez loro
Nombre científico:	Pecius lorus
Precio:	52
Origen:	Venezuela

Figura 14. Pantalla de alta de peces exóticos

6.6.- Modificación pez exótico

A esta pantalla se accederá a través del listado (de igual manera que para el alta) pero pulsando sobre el link "Modificar" situado en la fila del pez exótico que queremos modificar. De igual forma que en la pantalla anterior, obtenemos los mismos campos, pero esta vez estarán cumplimentados y solo tendremos que modificar los valores que deseemos.

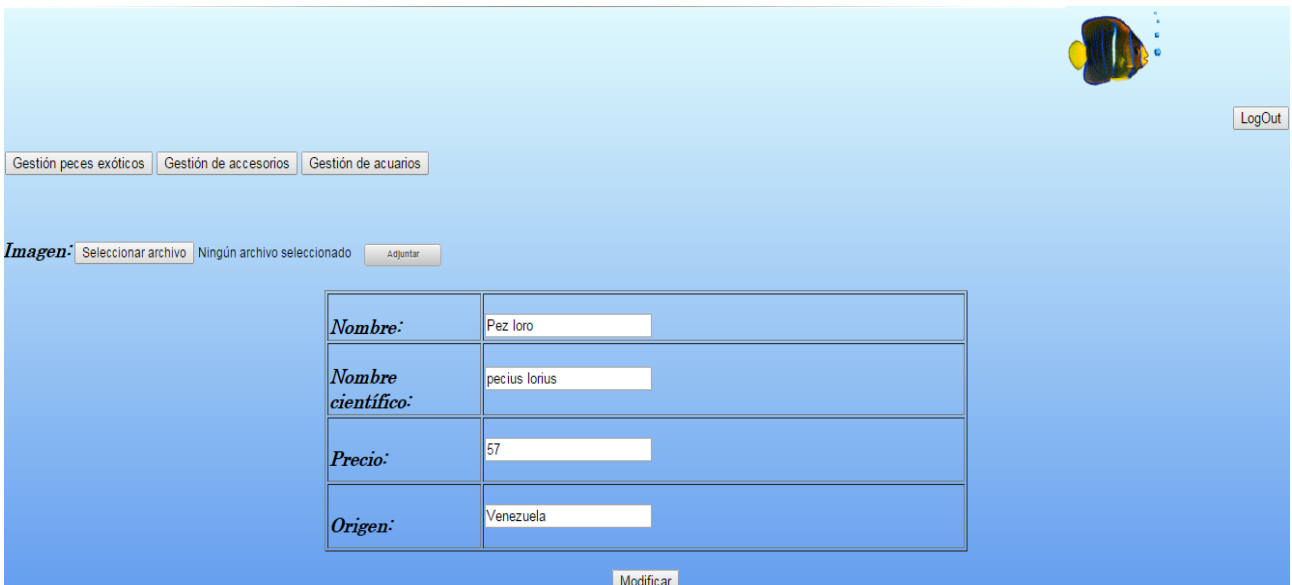


Figura 15. Pantalla de modificación de peces exóticos

Es importante ver que aunque muestre el id, este no será modificable ni tampoco insertable en BD ya que se realizará a través de una secuencia de BD.

6.7.- Eliminación pez exótico

Para la eliminación de un pez exótico, solo tendremos que pulsar sobre el link "Eliminar" situado al final de la fila del pez exótico que se quiera eliminar en cuestión. Al pulsar sobre el link aparecer una pantalla emergente donde deberemos pulsar "aceptar" en caso de que queramos eliminarlo definitivamente o sobre "cancelar" en caso de que queramos no eliminarlo.

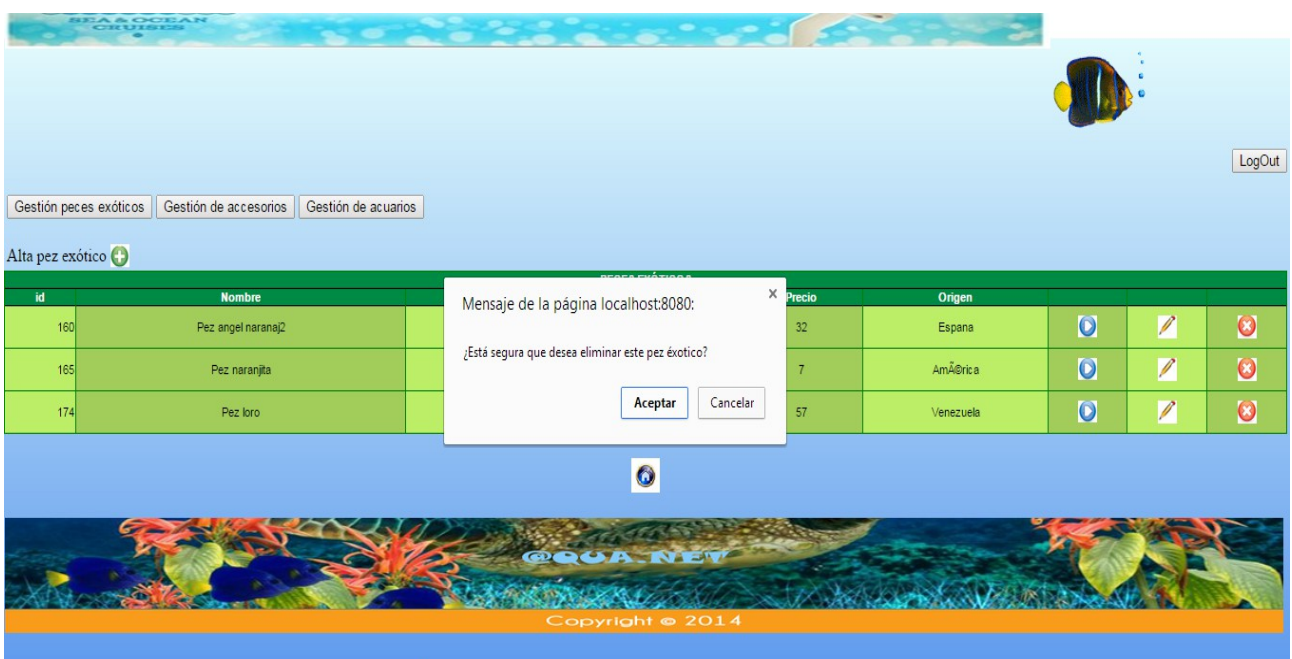


Figura 16. Pantalla de eliminación de peces exóticos

6.8.- Vista con usuario registrado

Una vez logado con un usuario registrado, se puede observar que las opciones de menú ya son diferentes. Aunque también existe acciones para Peces exóticos, Accesorios, Acuarios y Pedidos.



Figura 17. Pantalla de acceso de usuarios registrados

Al entrar en peces exóticos, se verá que solo podrá realizar pedidos y ver la categoría o pez exótico en cuestión. Observar que no aparecen las opciones de alta, modificación, ni eliminación del pez, ya que esto forma parte del usuario administrador.

Uuario: Maria Cuadrado Leon

LogOut

Peces exóticos | Accesorios marinos | Acuarios | Pedidos

PECES EXÓTICOS					
id	Nombre	Nombre científico	Precio	Origen	
160	Pez angel naranja2	PEcius naranju2	32	Espana	
165	Pez naranjita	Pecius naranjitus	7	América	
174	Pez loro	pecius lorius	57	Venezuela	

@QUA.NET

Figura 18. Pantalla de acceso al listado de peces exóticos

6.9.- Pedido con usuario registrado

Para realizar un pedido es sencillo, ya que lo único que se debe realizar es pulsar sobre el vínculo "Realizar pedido" situado en la última columna de pez exótico que se quiera realizar el pedido en cuestión. Al pulsar sobre el vínculo se abra una ventana emergente con dos opciones: Aceptar o Cancelar. La primera opción realizará el pedido del pez exótico que desea y la segunda opción cancelará el pedido. Al tratarse de un pedido contra reembolso y no de una compra directa no será mas complicado ya que quedará constancia del pedido y llegará ala vivienda dónde esté registrado el usuario en cuestión de 7 días.

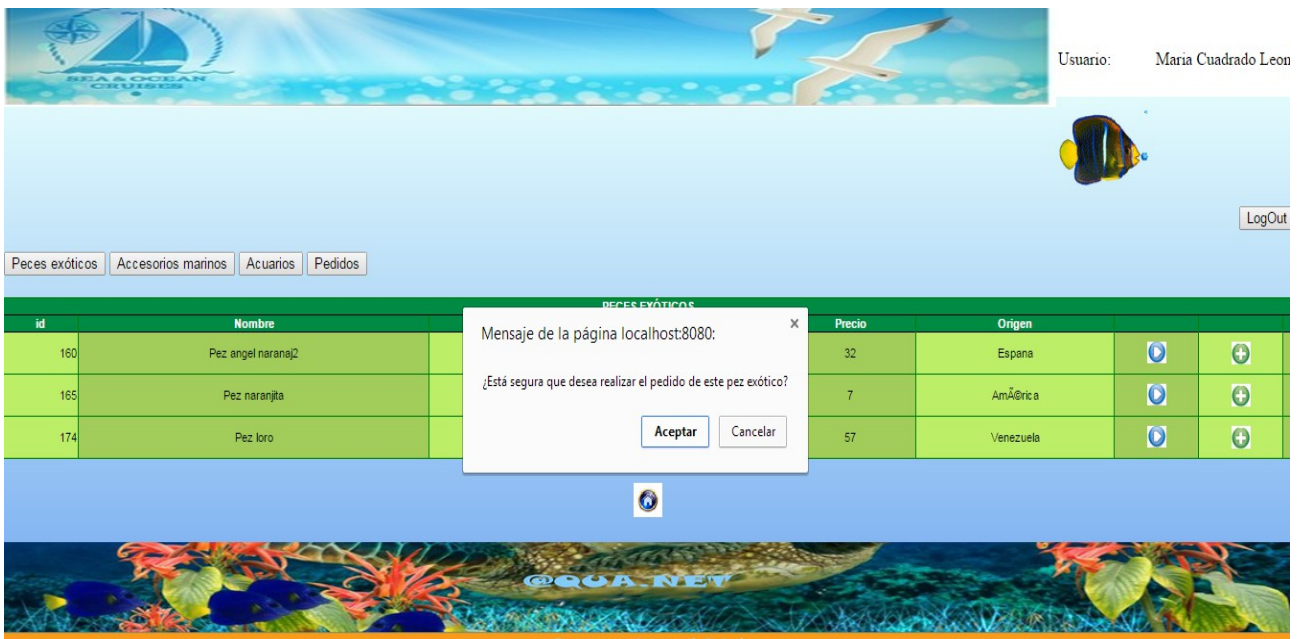


Figura 19. Pantalla de realización de pedidos

6.10.- Listado de pedidos

Para ver los pedidos asociados a un usuario, lo único que debe realizar es pulsar sobre el botón pedidos situado en la parte superior de la tabla, el cual mostrará los pedidos que tiene realizado el usuario logado.

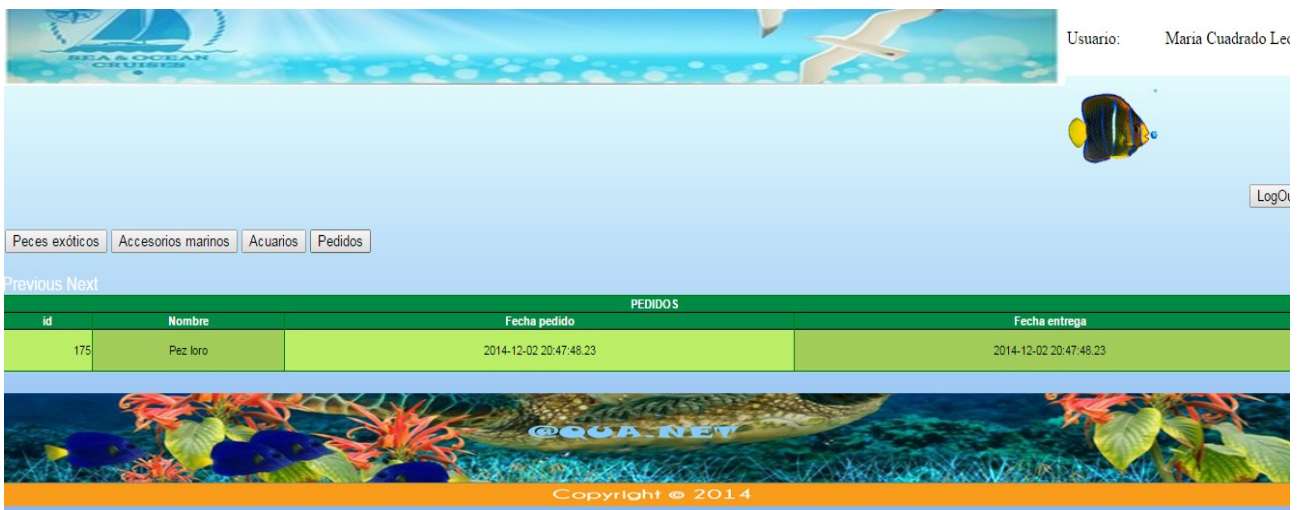


Figura 20. Pantalla de listado de pedidos

6.11.- Mostrar detalle del pez exótico

Para mostrar el detalle de una pez exótico lo único que deberemos realizar tanto para un usuario administrador como para un usuario exótico es pulsar sobre el vínculo "Show Exotic fish" o category.

Este vínculo mostrará en detalle (imagen incluida), del pez exótico seleccionado pero no se podrá en ningún

momento realizar modificación alguna sobre este producto.

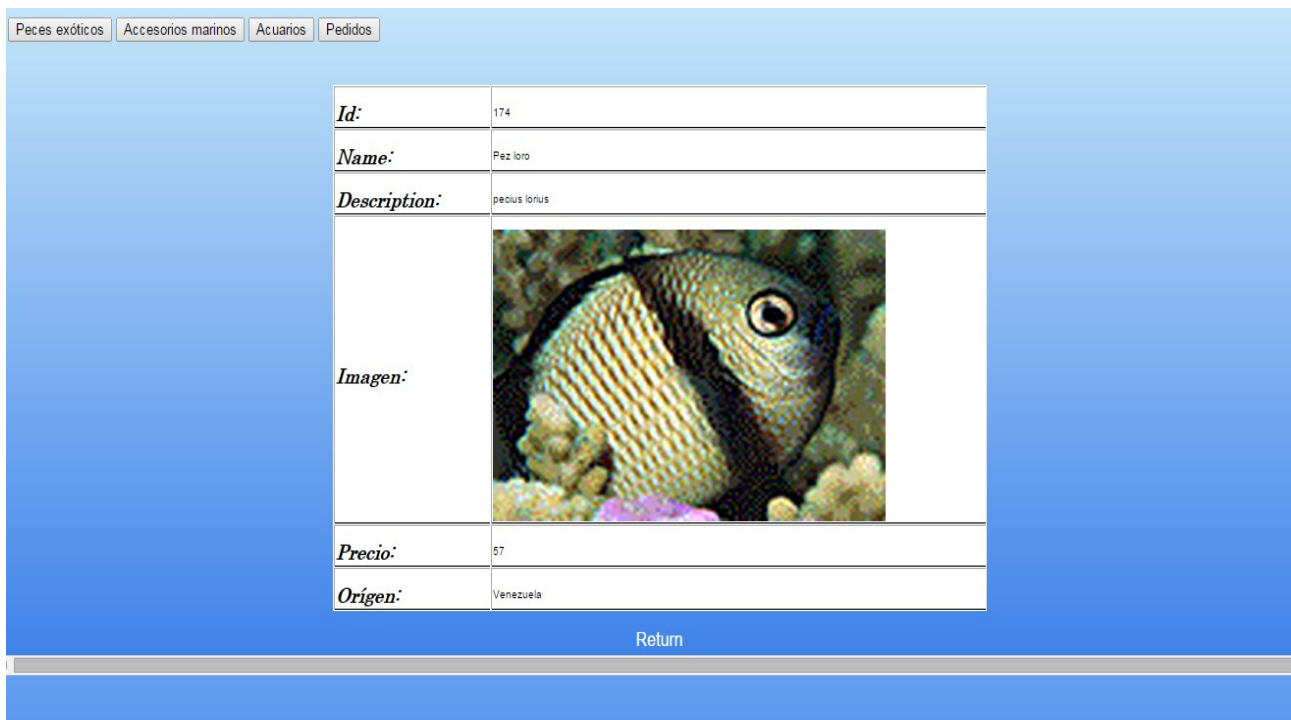


Figura 21. Pantalla de desalle de peces exóticos

7.- Diagrama de arquitectura

Se ha realizado una arquitectura de aplicación por niveles, donde los puntos más importantes que afectan a este proyecto son:

- La aplicación se desarrollara en tres niveles, que son la interface de usuario, la lógica de la aplicación y la base de datos.

El objetivo de la programación por niveles o capas, es la separación de la lógica del negocio de la capa visual, digamos separar la capa de acceso a la información a la capa de presentación de la misma al usuario.

Se se establecerán tres capas distintas:

Una de las mayores ventajas de este estilo de programación es que cada capa funciona autónomamente, por lo que un cambio en alguna de las capas, no afecta al resto. Además, permite una mejor distribución del trabajo a realizar y cada recurso se puede centrar en el área que

mejor gestiona.

- El acceso a la base de datos se realizará a través de JPA contra una base de datos Postgres.
- La creación de la tablas serán autogeneradas con el middleware de JPA.
- La tecnología a utilizar será JSF, JPA y EJB.

El desarrollo de la aplicación GestAquarium se realizara en las siguientes capas:

1. Capa de presentación o Interface de Usuario: Es la parte grafica que se le visualiza al usuario, donde se le comunica la información o se capta la información que introduce.

Por ello, debe ser fácil y entendible de usar. También, se puede aplicar una pequeña gestión de errores de introducción de datos

2. Capa de negocio o lógica de la aplicación: Se ubica la programación que se encargara de gestionar las peticiones y respuestas al usuario. Es donde se establecerán las reglas de negocio que se tienen que llevar a cabo para cada petición y las maneras en que se tratara la información. Digamos que hace de intermediaria entre la parte visual y la base de datos.

3. Capa de datos o Base de datos: Es el almacenamiento de los datos que gestiona la aplicación. Para ello se deberá disponer de un gestor de base de datos que reciba las solicitudes de almacenamiento o recuperación de la información.

· La capa de negocio se distribuirá en paquetes, que se iniciaran con la raíz es.uoc.tfc, más el nombre de la aplicación, nombre paquete y nombre de la clase. T

· Nombramiento de las clases se realizará con la primera letra mayúscula y el resto en minúsculas. Las clases formadas por varias palabras utilizarán mayúsculas para la inicial de cada una de

ellas, por ejemplo es.uoc.tf.paquete.Clase o
es.uoc.tfc.paquete.ClaseDeVariosVocablos

· Nombramiento de los métodos se realizará con todas las letras minúsculas, incluida la inicial.
Las clases formadas por varias palabras utilizarán mayúsculas para la inicial de las segundas palabras, por ejemplo es.uoc.tfc.paquete.Clase.metodo o
es.uoc.tfc.paquete.Clase.metodoDeVariosVocablos

7.1 PostgreSQL

PostgreSQL es una base de datos relacional, distribuida bajo licencia BSD y con su código fuente disponible libremente. Es el motor de bases de datos de código abierto más potente del momento y en sus últimas versiones empieza a no tener que envidiar nada a otras bases de datos comerciales.

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. En los últimos años se han concentrado mucho en la velocidad de proceso y en características demandadas en el mundo empresarial

1. Descarga del software

Hay que descargar el programa desde <http://www.postgresql.org>. Para el proyecto se utilizará la versión PostgreSQL v. 9.3.3-1:

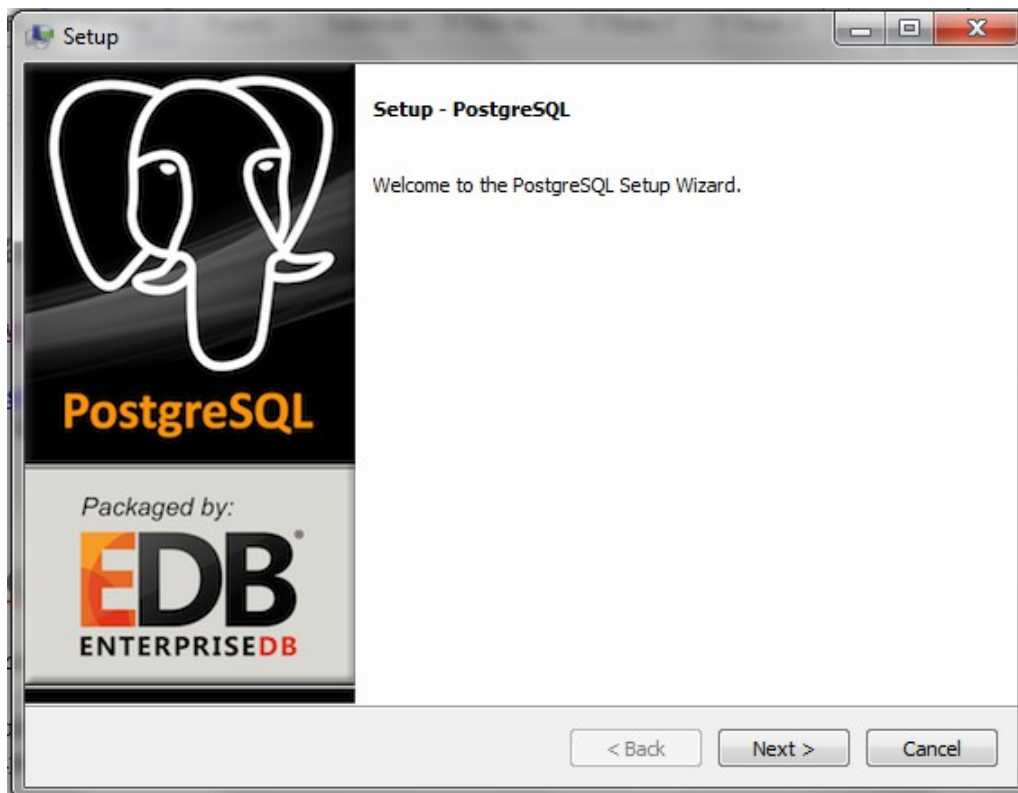
<http://www.enterprisedb.com/postgresql-933-installers-win32> ls=Crossover&type=Crossover

Puede encontrar información sobre el funcionamiento de PostgreSQL en:

<http://www.postgresql.org/docs>.

2. Instalación de PostgreSQL

1. Ejecutar el archivo que nos hemos bajado: postgresql-9.3.3-1-windows.exe.
2. Para el S.O Windows Vista o Windows 7, hay que realizar la instalación como usuario Administrador y desactivar el UAC (Panel de Control --> Centro de Actividades --> Cambiar configuración de Control de cuentas de usuario --> "no notificar nunca").
3. Acepta todas las opciones por defecto:



4. Para poder administrar PostgreSQL con la aplicación "pgAdmin III" (Inicio --> Todos los programas --> PostgreSQL 9.3 --> pgAdmin III). Con la aplicación he podido crear tablas, columnas, asignar permisos, esquemas de BD, etc.

7.2 Jboss

JBoss es un servidor de aplicaciones (Application Server) que implementa la arquitectura Java EE y es de código

abierto. JBoss está desarrollado en Java y por lo tanto, en principio, no debería tener ningún problema para funcionar en cualquier plataforma que soporte Java. Eso sí, requiere disponer de la versión JDK 1.6 o superior.

JBoss lleva incorporado la Base de Datos Hypersonic. Esta BD es la que utiliza por defecto JBoss. Nosotros utilizaremos la BD PostgreSQL, ya que tiene más posibilidades que Hypersonic., Por bajón tendremos que modificar algunos xml y añadir algunos archivos.

En este documento se explica cómo instalar y poner en funcionamiento el JBoss. También se explican los pasos necesarios para desarrollar y desplegar diferentes tipos de EJBs. Esto se hace introduciendo ejemplos sencillos de aplicaciones totalmente funcionales, que podrá probar en su máquina.

Importante: en la documentación que viene a continuación, el nombre de los ficheros correspondientes al software que podrá encontrar en el CD de software libre puede ser ligeramente diferente. Esto se debe a que nosotros utilizaremos las últimas versiones que encontraremos en Internet. En cualquier caso, tenga en cuenta que las explicaciones y los pasos a seguir contenidos en este documento siguen siendo totalmente válidos.

1. Instalación y ejecución de Jboss

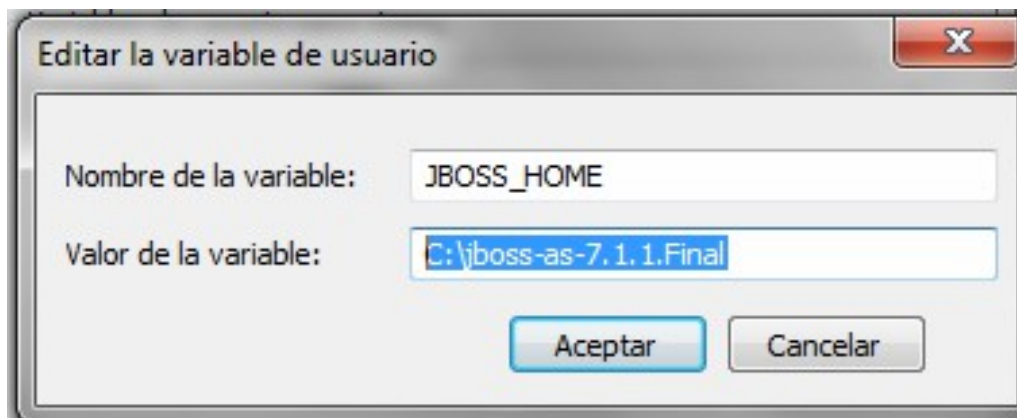
1.1 Instalación de Jboss

La instalación de JBoss es bastante sencilla, debe seguir los siguientes pasos:

1. Bajar y descomprimir el archivo jboss-as-7.1.1.Final.zip que podrá encontrar en la dirección: <http://www.jboss.org/jbossas/downloads>, concretamente en: <http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.Final/jboss-as-7.1.1.Final.zip>
2. Al descomprimir el archivo ya se creará automáticamente la carpeta de instalación de JBoss (siempre con el nombre C: \ JBoss-as-7.1.1.Final). Por tanto, esta carpeta no la debe crear vosotros,

sino que tiene que hacer la descompresión en la carpeta padre que elija (c: \ por ejemplo).

3. Nota: Conviene que la dirección de la carpeta en donde se instalará JBoss no contenga cada espacio en blanco, tipo "C: \ Archivos de Programas \ Jboss-as-7.1.1.Final \", ya que sino dará problemas a la hora de ejecutar JBoss desde Eclipse.
4. Crear una variable del sistema con el nombre JBOSS_HOME y como valor la carpeta donde esta instalado JBoss, por ejemplo: c: \ Jboss-as-7.1.1.Final.



5. Reiniciar el ordenador para los cambios surjan efecto.

7.3 Instalación del Conector Java JDBC PostgreSQL a Jboss

El conector JDBC consiste en un conjunto de clases Java que implementan la especificación dada por SUN de JDBC. Estas clases Java vienen en un archivo .jar. Los pasos a seguir son:

1. Bajar el conector Java JDBC de PostgreSQL desde <http://jdbc.postgresql.org> , concretamente lo encontraremos en el enlace <http://jdbc.postgresql.org/download/postgresql-9.3-1101.jdbc4.jar>

2. Crear la estructura de carpetas: \ postgresql \ main en la carpeta JBOSS_HOME \ modules \ org, así la estructura debe ser, por ejemplo: C: \ JBoss-as-7.1.1.Final \ modules \ org \ postgresql \ main.

3. Copiar el jar postgresql-9.3-1101.jdbc4.jar, corresponden a controlador JDBC de PostgreSQL que nos hemos bajado, en la carpeta JBOSS_HOME \ modules \ org \ postgresql \ main.

4. En la misma carpeta JBOSS_HOME \ modules \ org \ postgresql \ main, crear un archivo con el nombre module.xml con el siguiente texto:

```
<? xml version = "1.0" encoding = "UTF-8"?>
<module xmlns = "urn: JBoss: module: 1.0" name =
"org.postgresql">
<resources>
<resource-root path = "postgresql-9.3-1101.jdbc4.jar" />
</ resources>
<dependencias>
<module name = "javax.api" />
<module name = "javax.transaction.api" />
</ dependencias>
</ module>
```

5. Para que JBoss reconozca este DataSource hay que añadir en el xml JBOSS_HOME \ \ standalone \ configuration \ standalone.xml, concretamente colgando de la etiqueta <datasources>, el siguiente texto:

```
<datasource JTA = "false" JNDI-name = "java: JBoss /
postgresDS" pool-name = "postgresDS" enabled = "true"
use-java-contexto = "true" use-ccm = "false">
<connection-url> jdbc: postgresql: // localhost: 5432 /
postgres </ connection-url>
<driver-class> org.postgresql.Driver </ driver-class>
<driver> postgresql </ driver>
<pool />
```

```
<security>
<user-name> USER </ user-name>
<password> PASSWORD </ password>
</ security>
<statement />
</ DataSource>
```

USER y PASSWORD son el usuario y su password creado a PostgreSQL en el apartado 3.6 de la instalación de PostgreSQL. En mi caso la usuario le he llamado: USER y como password:110608

```
<user-name> USER </ user-name>
<password> 110608</ password>
```

Y colgando de la etiqueta <drives>:

```
<driver name = "postgresql" module = "org.postgresql">
<xa-datasource-class> org.postgresql.xa.PGXADatasource </
xa-datasource-class>
</ driver>
```

Así la etiqueta <datasources> puede quedar:

```
<datasources>
<datasource JNDI-name = "java: JBoss / datasources /
ExampleDS" pool-name = "ExampleDS" enabled = "true" use-
java-contexto = "true">
<connection-url> jdbc: h2: mem: test; DB_CLOSE_DELAY = -1
</ connection-url>
<driver> h2 </ driver>
<security>
<user-name> sa </ user-name>
<password> sa </ password>
</ security>
</ DataSource>
<datasource JTA = "false" JNDI-name = "java: JBoss /
postgresDS" pool- name = "postgresDS" enabled = "true"
use-java-contexto = "true" use-ccm = "false">
<connection-url> jdbc: postgresql: // localhost: 5432 /
postgres </ connection-url>
<driver-class> org.postgresql.Driver </ driver-class>
<driver> postgresql </ driver>
<pool />
<security>
```

```
<user-name> USER </ user-name>
<password> PASSWORD </ password>
</ security>
<statement />
</ DataSource>
<drivers>
<driver name = "h2" module = "com.h2database.h2">
<xa-datasource-class> org.h2.jdbcx.JdbcDataSource </ xa-
datasource-class>
</ driver>
<driver name = "postgresql" module = "org.postgresql">
<xa-datasource-class> org.postgresql.xa.PGXADatasource </
xa-datasource- class>
</ driver>
</ drivers>
</ datasources>
```

6. Arrancar Jboss.

7. En la carpeta del proyecto se encuentra el servidor configurado correctamente listo para arrancar.

8. Conclusiones

8.1 Conclusiones personales

Desde mi punto de vista personal, el proyecto ha constituido un gran reto para mí.

El proyecto final de carrera es la culminación de un largo trabajo, años y años de esfuerzo traducidos en una asignatura en la que se deben poner en práctica todos los conocimientos adquiridos.

Aun teniendo en cuenta la gran variedad de TFGs a escoger, consideré oportuno hacer un proyecto basado en la plataforma J2EE para, en cierto modo, poder ampliar mis humildes conocimientos técnicos sobre alguno de sus muchos marcos de trabajo.

Estos últimos meses han implicado un gran esfuerzo personal; días enteros de duro trabajo, largas noches a

base de café, pocas horas de sueño... pero todo ello ha merecido la pena con creces. Extraigo una enriquecedora experiencia de esta asignatura.

8.2 Justificación y razonamiento de las decisiones adoptadas durante

el proyecto

Hablando desde mi propia experiencia y siendo completamente sincero, he de decir que cuando comencé este TFG me dio un poco de pánico al darme cuenta que mi experiencia con J2EE era poca junto con las diferentes tecnologías que lo componen. No me quedó otra alternativa que adentrarme en ellas y tomarme su aprendizaje como un reto a corto plazo para finalmente poder desarrollar un proyecto de tal envergadura.

Tras investigar sobre la plataforma J2EE y sus diferentes tecnologías decidí utilizar el marco de trabajo de aplicación Java Server Faces (JSF), ya que es completamente novedoso y una tecnología puntera. Éste me facilitaba una sencilla separación de tareas, en capas, implementando un patrón de diseño Modelo-Vista-Controlador (MVC).

Mediante Internet he recopilado muchos manuales, documentación, tutoriales y ejemplos de aplicaciones Web sencillas con JSF que me han sido de mucha ayuda, sobretodo al inicio para poder realizar una buena configuración del proyecto.

Decidí también obtener el libro de Anaya Multimedia, el cual ha sido para mí un pilar imprescindible durante todo el TFG y aseguro que lo seguirá siendo.

A la hora de decidir qué Framework utilizar para la persistencia de datos en la aplicación lo tuve bastante claro, JPA. No fue porque ya había trabajado antes con él sino porque tenía siempre presente el poder utilizarlo, dado que había oído hablar maravillas de él y su actual importancia en el mundo de las conexiones de datos.

Integrar JPA en mi aplicación no me ha sido muy complicado, ya que hay muchos ejemplos en Internet que me

han servido de mucha ayuda. A la hora de mapear las entidades decidí hacerlo directamente con anotaciones y he de decir que no he encontrado ningún tipo de problema, me ha resultado bastante sencillo.

8.3 Experiencias o conclusiones extraídas de este TFG

La experiencia vivida durante el desarrollo de este TFG ha sido para mí muy emocionante, provechosa y me ha motivado muchísimo.

Hasta ahora no tenía claro qué hacer una vez acabara la carrera, pero el tutor comentó de realizar un aplicación distribuida y la verdad es que me entusiasmó la idea. Además, necesitaba esta experiencia para darme cuenta lo que de verdad me gusta y se me da mejor, que es sin duda el desarrollo de aplicaciones Web.

Dado el tiempo tan limitado que hemos tenido para llevar a cabo este proyecto y teniendo presente la inexperiencia que tenía en J2EE, ya que nunca había utilizado ninguna de sus tecnologías, ni tan siquiera taglibs de JSF, EJB, etc. es todo un logro el haber finalizado (aunque todavía le faltan mucho retoques y mejoras) esta aplicación tan satisfactoriamente y con una arquitectura tan completa como es la integración de varias tecnologías, cada una aportando lo mejor de ellas y complementándolas entre sí.

Si tuviera que volver a hacer el proyecto, ahora que ya tengo mejores conocimientos de JSF, EJB, JPA haría las pantallas de listados de forma diferente. Adquiriría conocimientos sobre ajax 4 jsf para poder manejar ajax en las páginas para no tener que renderizar páginas completas cuando solo necesito renderizar una región.

Y sobretodo crearía una pantalla de filtros para cada listado, que aunque es cierto que al principio no habrá muchos productos con el tiempo podría venir bien.

Otra cosa que me gustaría mejorar, es que para guardar una imagen es necesario darle al botón adjuntar y te resetea el resto del formulario. Por lo que lo primero que se debe hacer es adjuntar y después cumplimentar el

resto de campos y guardar. Se podría haber solcuionado utilizando ajax al cargar la imagen, pero tal y comocomento no me dió tiempo a realizar esto.

Y para finalizar, también me hubiera gustado que cuando la persona registrada realiza el pedido llegase un correo al indicado en el campo de registro, pero esto ni pude abordarlo.

Estas mejoras funcinoales querría haberla corregido antes de entregar el proyecto pero me faltó tiempo. No obstante, ya que la aplicación servirá para la vida real tengo el compromiso de hacerlo.

9. Bibliografía

<http://hop2croft.wordpress.com/2011/06/23/dont-repeat-the-dao-no-repitas-el-dao/>

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=desarrolloRapidoJava>

<http://openejb.codehaus.org/hello-world.html>

<http://www.informit.com/articles/article.aspx?p=1671224&seqNum=2>

<http://www.mkyong.com/jsf2/jsf-2-0-hello-world-example/>

http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html_single/#session-configuration

<http://es.scribd.com/doc/454457/tutorialHibernate>

http://www.slideshare.net/Emmerson_Miranda/hibernate-32-short-manual-9367150

http://www.dosideas.com/wiki/Consultas_Con_HibernateTemplate

<http://www.aprende-gratis.com/css/curso.php?lec=clases-identificadores>

<http://www.manualdecss.com/codigos-css/indice-codigos.html>

<http://www.primefaces.org/showcase/ui/overlay/dialog/loginDemo.xhtml>

<http://www.javaworld.com/article/2077817/java-se/understanding-jpa-part-1-the-object-oriented-paradigm-of-data-persistence.html>

<http://wiki.eclipse.org/EclipseLink/Examples/JPA>