

Especificación de Requisitos

ANIMALIA

Modelado del Software

Grupo E

05/10/2023

Integrantes:

Airam Martín Peraza

Álvaro Velasco García

Axel Kahou He Expósito

Carlos Mayorga Santiago

Carlo Dubini Marqués

Clara Sotillo Sotillo

Jorge Calvo Fernández

Mariana Cucicea

Sara Sacó Baños

Sergi Cristóbal Gambau

Sofía Postigo Ruiz

Steven Pérez Salinas

ÍNDICE

1. INTRODUCCIÓN	6
1.1 Propósito	6
1.2. Alcance	6
1.3. Definiciones, acrónimos y abreviaturas	7
1.4 Referencias	10
1.5. Resumen	11
2. DESCRIPCIÓN GENERAL	11
2.1 PERSPECTIVA DEL PRODUCTO	11
2.1.1 Contexto De Implantación	11
2.2 FUNCIONES DEL PRODUCTO	12
2.2.1 Gestión del zoológico	12
2.2.2 Gestión de la tienda	14
3. REQUISITOS ESPECÍFICOS	16
3.1 INTERFACES EXTERNAS	16
3.2 FUNCIONES	17
3.2.1 Empleado	17
3.2.1.1 Alta Empleado	17
3.2.1.2 Baja Empleado	18
3.2.1.3 Modificar Empleado	18
3.2.1.4 Mostrar Empleado	19
3.2.1.5 Listar Empleado	20
3.2.1.6 Listar Empleados por Hábitat	21
3.2.1.7 Listar empleado por especie en Habitat	21
3.2.2 Factura	22
3.2.2.1 Abrir factura	22
3.2.2.2 Cerrar factura	23
3.2.2.3 Mostrar factura	24
3.2.2.4 Listar facturas	24
3.2.2.5 Devolver factura	25
3.2.2.6 Añadir pase a factura	26
3.2.2.7 Quitar pase de factura	27
3.2.2.8 Modificar factura	27
3.2.3 Pase	28
3.2.3.1 Alta pase	28
3.2.3.2 Baja Pase	29
3.2.3.3 Mostrar Pase	30

Especificación de Requisitos Software

3.2.3.4 Listar Pases	30
3.2.3.5 Modificar Pase	31
3.2.3.6 Listar Pases Por Hábitat	32
3.2.4 Hábitat	33
3.2.4.1 Alta Hábitat	33
3.2.4.2 Baja Hábitat	33
3.2.4.3 Mostrar Hábitat	34
3.2.4.4 Modificar Hábitat	35
3.2.4.5 Listar Hábitat	36
3.2.4.6 Vincular Empleado a Hábitat	36
3.2.4.7 Desvincular Empleado de Hábitat	37
3.2.4.8 Listar Habitats por Empleado	38
3.2.4.9 Calcular hábitat con más ingresos	38
3.2.5 Especie	39
3.2.5.1 Alta Especie	39
3.2.5.2 Baja Especie	40
3.2.5.3 Mostrar Especie	41
3.2.5.4 Modificar Especie	41
3.2.5.5 Listar Especies	42
3.2.5.6 Listar Especie por Habitat	43
3.2.6 Animal	44
3.2.6.1 Alta Animal	44
3.2.6.2 Baja Animal	44
3.2.6.3 Modificar Animal	45
3.2.6.4 Mostrar Animal	46
3.2.6.5 Listar Animales	47
3.2.6.6 Listar Animales por Especie	48
3.2.7 Departamento	48
3.2.7.1 Alta Departamento	48
3.2.7.2 Baja Departamento	49
3.2.7.3 Modificar Departamento	50
3.2.7.4 Mostrar Departamento	51
3.2.7.5 Listar Departamentos	51
3.2.7.6 Calcular Nomina Departamento	52
3.2.8 Trabajador	53
3.2.8.1 Alta Trabajador	53
3.2.8.2 Baja Trabajador	54
3.2.8.3 Modificar Trabajador	55

Especificación de Requisitos Software

3.2.8.4	Mostrar Trabajador	56
3.2.8.5	Listar Trabajadores	57
3.2.8.6	Listar Trabajadores por Departamento	57
3.2.8.7	Calcular Sueldo Trabajador	58
3.2.9	Venta	59
3.2.9.1	Abrir Venta	59
3.2.9.2	Añadir producto a venta	60
3.2.9.3	Quitar producto de venta	60
3.2.9.4	Cerrar Venta	61
3.2.9.5	Mostrar Venta	62
3.2.9.6	Listar Ventas	63
3.2.9.7	Listar Ventas por Empleado	63
3.2.9.8	Devolución Venta	64
3.2.9.9	Modificar Venta	65
3.2.10	Producto	66
3.2.10.1	Alta de Producto	66
3.2.10.2	Baja Producto	67
3.2.10.3	Mostrar Producto	67
3.2.10.4	Modificar Producto	68
3.2.10.5	Listar Productos	69
3.2.10.6	Listar Productos por Marcas de Producto	70
3.2.11	Marca de Producto	70
3.2.11.1	Alta Marca de Producto	70
3.2.11.2	Baja Marca de Producto	71
3.2.11.3	Modificar Marca de Producto	72
3.2.11.4	Mostrar Marca de Producto	73
3.2.11.5	Listar Marca de Producto	73
3.2.11.6	Listar Marcas por Proveedor	74
3.2.12	Proveedor	75
3.2.12.1	Alta Proveedor	75
3.2.12.2	Baja Proveedor	76
3.2.12.3	Modificar Proveedor	77
3.2.12.4	Mostrar Proveedor	77
3.2.12.5	Listar Proveedores	78
3.2.12.6	Listar Proveedores por Marca de Producto	79
3.2.12.7	Vincular Proveedor con Marca de Producto	80
3.2.12.8	Desvincular Proveedor con Marca de Producto	80
3.3	Requisitos de rendimiento	81

Especificación de Requisitos Software

3.4 Modelo de Dominio	83
3.5 Restricciones de diseño	85
3.6 Atributos del sistema	85

1. INTRODUCCIÓN

1.1 Propósito

La finalidad de este documento es desarrollar la Especificación de Requisitos Software (SRS) del proyecto *Animalia*. En él se especifican las distintas funcionalidades de la aplicación mediante casos de uso y el modelo de dominio realizado por el equipo. Así mismo, se incluye un primer prototipo de la interfaz correspondiente a alguno de los casos de uso e información de interés, describiendo de manera clara y precisa las características, especificaciones técnicas, requisitos y funcionalidades del producto. Creando una visión clara y rigurosa de los requisitos del software a desarrollar, permitiendo comprobar que se ajusta a los intereses y condiciones o bien para desarrollarla correctamente. Va dirigida tanto a las partes involucradas en el desarrollo del proyecto, como a cualquiera de los usuarios que vaya a hacer uso del producto, el profesor de la asignatura incluido.

1.2. Alcance

El proyecto, de nombre *Animalia*, se divide en 2 grandes partes. En primer lugar, tiene como objetivo el desarrollo de una aplicación de escritorio que facilite el proceso de gestión de un zoológico. Por otro lado, pretende crear una aplicación, también de escritorio, para gestionar una tienda, propia de un zoológico.

- Sistema de gestión del zoológico

La aplicación está orientada al personal de un zoológico en el ámbito administrativo. Así, permite al usuario gestionar los hábitats que se encuentran en el recinto y a su vez los animales que se encuentran en cada uno de ellos según su especie y su ecosistema. Por otro lado, al usuario también se le permite tramitar pases organizados en el zoológico para el entretenimiento de los visitantes con sus respectivos recibos. Por último, la aplicación también permite la administración del personal activo con respecto a sus labores.

- Sistema de gestión de tienda

La aplicación está orientada al personal de la tienda de un zoológico en el ámbito administrativo. De este modo, permite gestionar los productos que se ofrecen según la marca que abastecen sus respectivos proveedores, las ventas que se realizan y los empleados asociados a la actividad comercial conforme al departamento que pertenecen.

Puesto que, para ambas partes del proyecto, el uso de Animalia está destinado únicamente al personal de administración, no existe ningún subsistema de seguridad que determine los roles que representan los usuarios o su nivel de privilegios sobre las operaciones.

1.3. Definiciones, acrónimos y abreviaturas

Capítulo dedicado a la definición de una serie de conceptos necesarios para el correcto entendimiento de la especificación de requisitos, ya que serán citados a lo largo del documento.

CONCEPTO/ABREVIATURA	DEFINICIÓN
Animal	En este proyecto, cada uno de los individuos que pertenecen a una especie y que residen en un hábitat del zoo.
App	Aplicación.
BD	Base de Datos.
Caso de uso	Fragmento de la funcionalidad del sistema que proporciona un resultado de interés para el usuario, que interactúa con él. Contiene la secuencia de acciones que generan dicho resultado, incluyendo posibles alternativas a la secuencia principal.
CIF	Código de Identificación Fiscal.
Cliente	En este proyecto, persona que hace uso de los servicios del zoo (compra pases y visita hábitats) o de la tienda (compra productos).
Departamento	En este proyecto, sección en que está dividida la empresa que constituye la tienda de un zoo, según los distintos ámbitos en los que trabajan los empleados. Se caracteriza por su nombre único.
DNI	Documento Nacional de Identidad.
Empleado	En este proyecto, persona que por un sueldo desempeña los trabajos asociados al propio zoológico según corresponda.

	Puede pertenecer al personal de limpieza o al zoológico. Sus sueldos se calcularán de manera diferente.
Entidad	Objetos o sujetos de la base de datos de los que queremos almacenar información.
Especie	En este proyecto es la clasificación de animales semejantes entre sí que pueden cohabitar un hábitat.
Factura	En este proyecto, la acción de obtener un determinado número de pases para poder entrar a los distintos hábitats, se distinguen por su hora y fecha.
Funcionalidad	Lo que hace el sistema.
GUI	Graphic User Interface.
Hábitat	En este proyecto, lugar donde residen las especies, es cuidado por los empleados y al cual se accede mediante un pase
IEEE	Estándar cuyo objetivo es definir métricas de software y su uso en la evaluación de componentes software. Propone la construcción de modelos de calidad a medida adaptados a cada proyecto.
ID	Identificador único.
Interfaces Externas	Forma de interactuar del sistema con las personas, el hardware del sistema, otro hardware y otro software.
JDK	Java Development Kit.
JPA	Java Persistence API.
Marca	En este proyecto, distinguen en el mercado dos productos similares fabricados por distintas empresas.
MySQL	Sistema de gestión de bases de datos

	relacional.
N/A	No hay, no corresponde en el caso (not applicable).
Pase	En este proyecto, exhibición de especies en un hábitat concreto en un horario específico con un precio establecido.
Producto	En este proyecto, cualquier elemento ofrecido por la tienda de un zoo con su nombre, el stock disponible en un determinado momento y su precio. Puede tratarse de alimentación como palomitas o refresco, con el tipo correspondiente; o merchandising, como una mochila o una taza, con su categoría y si se trata o no de una edición limitada
Proveedor	En este proyecto, empresa que suministra a la tienda las marcas que esta vende, y que tiene un nombre, un CIF que la identifica unívocamente y un teléfono de contacto.
Requisito	Es la descripción de los servicios y restricciones.
SGBD	Sistema Gestor de Base de Datos.
SRS	Especificación de Requisitos Software (Software Requirement Specifications).
Trabajador	En este proyecto, persona que por un sueldo desempeña los trabajos asociados a la tienda de un zoo. Puede trabajar a tiempo completo (jornada laboral de 40h semanales) o a tiempo parcial (por horas).
User-friendly	Si al usuario, especialmente en el ámbito de la informática, le resulta fácil trabajar con ello.

Venta	En este proyecto, ofrecer al público los productos vendidos en la tienda de un zoo por un precio establecido.
-------	---

1.4 Referencias

Recopilación de fuentes utilizadas en la elaboración de la SRS:

- ❖ Transparencias de la asignatura Modelado de Software de la universidad complutense de madrid proporcionadas por el profesor Antonio Navarro Martín.
- ❖ Libros recomendados utilizados:
 - Alur, D., Crupi, J., Malks, D. Core J2EE Patterns: Best Practices and Design Strategies, 2nd Edition. Prentice-Hall PTR, 2003.
 - IEEE Std. 830-1998. Recommended Practice for Software Requirements Specification, 1998.
 - Rumbaugh, J., Booch, G., Jacobson, I. El lenguaje unificado de modelado. Manual de referencia. Addison Wesley, 2004.
 - Pressman, R.S., Maxim, B.R. Software Engineering. A Practitioner's Approach. 9th edition. McGraw-Hill, 2019.
- ❖ Referencias utilizadas en el diseño del modelo:
 - <https://nubika.es/noticias/principales-caracteristicas-zoologicos/>
- ❖ Construcción de los animales separados por especies:
 - <https://es.wikipedia.org/wiki/Animalia>
 - <https://www.biodiversidad.gob.mx/especies/cuantasesp>
- ❖ Referencias para mejor entendimiento entre marca, proveedor y producto:
 - <https://www.biodiversidad.gob.mx/especies/cuantasesp>
- ❖ Referencia de posibles pases de zoológicos y lo que contiene:
 - <https://www.zoosantillanadelmar.com/tu-visita/pase-anual>
 - <https://www.zoomadrid.com/>
- ❖ Definición CIF:
 - <https://www.sumup.com/es-es/facturas/glosario/cif/>
- ❖ Definición de caso de uso y su especificación:
 - <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.3?topic=requirements-defining-use-cases>
- ❖ Definición de User-friendly:
 - <https://rankmyapp.com/es/user-friendly-los-conceptos-basicos-y-como-hacerlo/>

1.5. Resumen

Esta SRS está apoyada en el estándar de IEEE (Institute of Electrical and Electronics Engineers) para el diseño y la especificación de requisitos. Se encuentra dividida en 3 grandes módulos:

- En primer lugar encontramos, la **Introducción**, que describe la información que recopila esta SRS, el propósito del proyecto, el objetivo de la idea, información complementaria referenciada y definición de términos utilizados en la misma.
- En segundo lugar, la **Descripción general**, donde encontraremos los requisitos que se han de implementar en el proyecto, las funciones finales del trabajo, las restricciones y las características del propio usuario.
- En la última sección, **Requisitos específicos**, es la encargada de recopilar gracias a toda la información ya mencionada en los módulos anteriores, los requerimientos del software implementado en detalle. Se puede encontrar las interfaces, el rendimiento, los requisitos especificados y los atributos del sistema.

2. DESCRIPCIÓN GENERAL

En los siguientes apartados se identificarán los requisitos sin contemplar los detalles de estos.

2.1 PERSPECTIVA DEL PRODUCTO

2.1.1 Contexto De Implantación

Animalia es un programa software que se concibe para la gestión de Zoos y de la tienda. El programa es local por lo que los únicos requisitos para que funcione es que se tenga un Sistema Operativo que sea compatible con Java.

Animalia funcionará a través de una base de datos donde se gestionará toda la información de los empleados, los animales y los clientes. No se necesitará ninguna conexión a internet debido a que la Base de datos será local.

La gestión se llevará a cabo en una interfaz gráfica donde el usuario pueda gestionar la información con mayor facilidad.

Otros aspectos de interés acerca del producto son:

- Interfaz del sistema → La aplicación emplea una base de datos relacional externa, donde se almacenan todos los datos en su contexto.
- Interfaz del usuario → La aplicación de escritorio hace uso de las interfaces proporcionadas por la biblioteca gráfica Swing de Java.
- Interfaz hardware → No se presenta ningún tipo de limitación hardware, por lo que la app podrá ser ejecutada en cualquier tipo de equipo informático.
- Interfaz software → Para ejecutar la aplicación se necesita un equipo que contenga instalada la máquina virtual de Java.
- Interfaz de comunicación → La aplicación se conectará con una base de datos MySQL externa.
- Memoria → Se utilizarán los recursos imprescindibles que, de hecho, serán bastante reducidos.
- Requisitos de adaptación → No se precisa de proceso de instalación, únicamente de descarga.

2.2 FUNCIONES DEL PRODUCTO¹

Animalia ofrece numerosas funcionalidades para el usuario, tanto la parte de gestión del zoológico como la de gestión de la tienda. Seguidamente se describen brevemente las funcionalidades dividiéndolas por módulos, para mantener una lista estructurada y clara.

2.2.1 Gestión del zoológico

- **Empleado**
 - Alta Empleado: registra un nuevo empleado.
 - Baja Empleado: establece que un empleado deja de serlo.
 - Modificar Empleado: cambia los atributos de un empleado
 - Mostrar Empleado: muestra todos los atributos de un empleado.
 - Listar Empleados: muestra una lista de todos los empleados del zoológico junto con sus atributos generales.
 - Listar Empleados por Hábitat: muestra una lista de todos los empleados del hábitat insertado.
 - Listar empleados por especie en hábitat: muestra una lista de todos los empleados que trabajan para uno o varios hábitats que tienen la especie especificada.
- **Factura**
 - Abrir factura: abre una factura que se puede modificar.
 - Cerrar factura: registra una factura haciendo que esta ya no se pueda modificar.

¹ ACLARACIÓN: en las funciones de modificación, nunca se va a poder modificar el id de las entidades.

- Mostrar factura: Muestra los atributos de una factura
- Listar facturas: Muestra una lista de todas las facturas con sus atributos.
- Devolver factura: Devuelve los pases vendidos que están dentro de la factura
- Añadir pase a factura: Se añade un pase a la compra.
- Quitar pase de factura: Se retira uno de los pases de la factura.
- **Pase**
 - Alta Pase: Registra una entrada para el zoo.
 - Baja Pase: Desactiva la entrada del zoo.
 - Mostrar Pase: Muestra los atributos del pase.
 - Modificar Pase: Modifica los atributos del pase.
 - Listar Pases: Muestra una lista de todos los pases del zoo con sus atributos respectivamente.
 - Listar Pases por Hábitat: Muestra una lista de los pases de un hábitat en específico con sus respectivos atributos.
- **Hábitat**
 - Alta Hábitat: Se registra un nuevo hábitat.
 - Baja Hábitat: Desactiva un hábitat.
 - Mostrar Hábitat: Muestra toda la información de un hábitat
 - Modificar Hábitat: Modifica la información de un hábitat
 - Listar Hábitats: Muestra una lista de todos los hábitats del zoológico con sus respectivos atributos.
 - Vincular Empleado a Hábitat: Relaciona a un empleado con el hábitat en que trabaja.
 - Desvincular Empleado de Hábitat: Elimina la relación entre un empleado y un hábitat en el que ya no trabaja.
 - Listar Hábitats por Empleado: Muestra una lista con todos los hábitats en los que trabaja un empleado.
 - Calcular el hábitat con más ingresos: muestra el hábitat que ha generado más dinero en función del importe de las facturas correspondientes a los pases recibidos por dicho hábitat en el intervalo de tiempo indicado.
- **Especie**
 - Alta Especie: se registra una nueva especie de animales.
 - Baja Especie: se desactiva la especie en la BD.
 - Mostrar Especie: Muestra los atributos de una especie.
 - Modificar Especie: Permite cambiar los atributos de una especie
 - Listar Especies: Muestra una lista de todas las especies del zoológico con sus respectivos atributos.
 - Listar Especies por Hábitat: Muestra la lista de las especies que pertenecen a un hábitat con sus respectivos atributos.

- **Animal**

- Alta Animal: registra un nuevo animal.
- Baja Animal: desactiva la pertenencia de un animal en el zoológico.
- Modificar Animal: Cambia los atributos del animal.
- Mostrar Animal: Muestra todos los atributos de un animal.
- Listar Animales: Muestra una lista de todos los animales pertenecientes al zoológico con sus respectivos atributos.
- Listar animales por especie: Muestra una lista de todos los animales que pertenecen a una especie en concreto con sus respectivos atributos.

2.2.2 Gestión de la tienda

- **Departamento**

- Alta Departamento : registra un nuevo departamento.
- Baja Departamento : desactiva un departamento.
- Mostrar Departamento: Muestra todos los atributos de un departamento.
- Modificar Departamento: Cambia los atributos de un departamento.
- Listar Departamento: Muestra una lista de todos los departamentos de la tienda, junto con todos sus atributos.
- Calcular Nomina Departamento: Muestra el sumatorio de todos los empleados activos que pertenecen al departamento.

- **Trabajador**

- Alta Trabajador:: registra un nuevo trabajador.
- Baja Trabajador : desactiva un empleado, indicando que no trabaja ya en el zoo.
- Modificar Trabajador: Cambia los atributos del trabajador.
- Mostrar Trabajador: Muestra todos los atributos de un trabajador.
- Listar Trabajador: Muestra una lista de todos los trabajadores de la tienda.
- Mostrar Trabajador por Departamento: Muestra una lista de todos los trabajadores de un departamento, junto a todos sus atributos.
- Calcular Sueldo Trabajador: Calcula el sueldo del trabajador en base al tipo.

- **Venta**

- Abrir Venta: Inicia una venta de un conjunto de productos.
- Añadir producto a venta: Registra nuevos productos en la venta.
- Quitar producto de venta: Descarta productos en la venta.
- Cerrar venta: Finaliza el pedido
- Mostrar venta: Muestra todos los atributos de la venta.

- Listar ventas: Muestra una lista con todas las ventas realizadas y sus atributos.
- Listar ventas por empleado: Muestra una lista con todas las ventas realizadas por un empleado.
- Devolución (necesita cambio): Se encarga de realizar la devolución del dinero al cliente y de los productos correspondientes a la tienda de una venta.
- Mostrar venta por producto: Muestra una lista con todas las ventas de un producto junto con sus atributos.
- **Producto**
 - Alta Producto : registra un nuevo producto.
 - Baja Producto : desactiva un producto indicando que ya no está en la tienda.
 - Modificar Producto: Cambia los atributos de un producto.
 - Mostrar Producto; Muestra todos los atributos de un producto.
 - Listar Productos: Muestra una lista de todos los productos de la tienda con sus atributos.
 - Listar Productos por Marcas de Producto: Muestra una lista con todos los productos de una marca que vende la tienda, junto con sus atributos.
- **Marca de Producto**
 - Alta Marca : registra una nueva marca.
 - Baja Marca: desactiva una marca indicando que esa marca ya no se vende en el zoo.
 - Modificar Marca: Cambia los atributos de una marca.
 - Mostrar Marca: Muestra los atributos de una marca.
 - Listar Marca: Muestra una lista con todas las marcas que vende la tienda y sus atributos.
 - Listar Marcas por Proveedor: Muestra una lista con todas las marcas que proporciona un proveedor a la tienda junto con sus atributos.
- **Proveedor**
 - Alta Proveedor: registra un nuevo proveedor.
 - Baja Proveedor: desactiva un proveedor indicando que ya no proporciona productos al zoo.
 - Modificar Proveedor: Cambia los atributos de un proveedor.
 - Mostrar Proveedor: Muestra todos los atributos de un proveedor.
 - Listar Proveedores: Muestra una lista con todos los proveedores de la tienda y sus atributos.
 - Listar proveedores por marca de producto: Muestra una lista con todos los proveedores que venden una marca a la tienda, junto con sus atributos.
 - Vincular proveedor con marca de producto: Relaciona a un proveedor con una marca que distribuye.

- Desvincular proveedor con marca de producto: Elimina la relación entre un proveedor y una marca que ya no distribuye.

3. REQUISITOS ESPECÍFICOS

3.1 INTERFACES EXTERNAS

A continuación, se adjunta un prototipo de la vista correspondiente a un caso de uso del sistema de gestión del zoológico. En particular, la correspondiente a “Alta Empleado” (Ilustración 1), para que el lector pueda visualizar cómo las funcionalidades descritas de Animalia se traducen en un software real.

El prototipo de la interfaz de usuario para el caso de uso "Alta Empleado" se presenta dentro de un recuadro que simula una ventana de aplicación. En la parte superior, centrado, se indica "(LOGO ANIMALIA)". Debajo de esto, hay una barra de navegación horizontal con cinco botones: "Departamento", "Venta", "Producto", "Marca de producto" y "Proveedor". A la izquierda de la interfaz, hay un menú vertical con los siguientes ítems: "Alta Empleado" (destacado con un fondo azul), "Baja Empleado", "Modificar Empleado", "Mostrar Empleado", "Listar Empleado" y "Listar Empleado por Departamento". El área principal de la interfaz tiene un encabezado de color morado que dice "ALTA EMPLEADO". Debajo de este encabezado, se encuentran los campos de entrada y las opciones para crear un nuevo empleado:

- DNI:** Un campo de texto.
- Nombre:** Un campo de texto.
- Sueldo Base:** Un campo de texto.
- Teléfono:** Un campo de texto.
- Horas extra:** Un campo de texto.
- Euros/hora extra:** Un campo de texto.
- Horas:** Un campo de texto con fondo gris.
- Euros/h:** Un campo de texto con fondo gris.

Además, hay una etiqueta "Tipo" seguida de dos opciones de radio: "Completo" (seleccionada, con un círculo azul) y "Parcial" (no seleccionada, con un círculo gris).

Ilustración 1

3.2 FUNCIONES

3.2.1 Empleado

3.2.1.1 Alta Empleado

Concepto	Descripción
Identificador	M_EM_01
Nombre	Alta Empleado
Autor	Sara Sacó Baños
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • DNI • nombre • sueldo_base • teléfono • tipo={zoológico,limpieza}} • especialidad(zoológico) • tasa(zoológico) • experiencia(zoológico) • suplemento(limpieza) • zona(limpieza)
Salida	Mensaje de éxito/fracaso junto con el id del empleado
Precondición	No hay identificadores repetidos BD.
Postcondición	No hay identificadores repetidos en la BD. Se inserta un nuevo empleado en la BD.
Descripción	Comprueba que los datos introducidos son correctos, si no lo son, se muestra un mensaje donde se informa del error. Si es así, pasa a comprobar si el DNI ya se encuentra en nuestra base de datos. Si el DNI ya existe, entonces se comprueba si está activo. Si está activo, se muestra un error. Si se encuentra inactivo, se reactiva con los datos actualizados. Si el DNI no se encontraba ya en la base de datos se introduce en ella un nuevo empleado.

3.2.1.2 Baja Empleado

Concepto	Descripción
Identificador	M_EM_02
Nombre	Baja Empleado
Autor	Sara Sacó Baños
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_empleado
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si éxito, empleado dado de baja
Descripción	Comprueba que el Id introducido está en el formato correcto. Comprueba que existe un empleado activo en la base de datos con el ID introducido. Si es así, pasa a estar inactivo y se desvinculan los hábitats que tiene asociados. En caso contrario, se informa del error.

3.2.1.3 Modificar Empleado

Concepto	Descripción
Identificador	M_EM_03
Nombre	Modificar Empleado
Autor	Sara Sacó Baños
Prioridad	Baja
Estabilidad	Alta
Origen	GUI

Destino	BD
Entrada	<ul style="list-style-type: none"> • id_empleado <p>Uno o más de los siguientes datos:</p> <ul style="list-style-type: none"> • nombre • sueldo_base • teléfono • DNI • tipo={zoológico,limpieza}} • especialidad(zoológico) • tasa(zoológico) • experiencia(zoológico) • Suplemento(limpieza) • zona(limpieza)
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si éxito, empleado actualizado.
Descripción	Se comprueba que el empleado existe y se encuentra activo, si está activo, se comprueba que los datos a modificar son sintácticamente correctos, y si el campo a modificar es el Dni, se comprueba si no existe en la BD. En caso afirmativo, en base al tipo de Empleado se actualizan los datos del empleado correspondiente y se muestra un mensaje de éxito. En caso contrario se muestra un mensaje indicando el error producido.

3.2.1.4 Mostrar Empleado

Concepto	Descripción
Identificador	M_EM_04
Nombre	Mostrar Empleado
Autor	Sara Sacó Baños
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_empleado

Salida	Se muestra el empleado registrado y activo con los atributos nombre, sueldo_base, teléfono, DNI, tipo={zoológico,limpieza}}, especialidad(zoológico), tasa(zoológico) y experiencia(zoológico) o Suplemento(limpieza) y zona(limpieza) (dependiendo de su tipo)
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se comprueba que el Id introducido está en el formato correcto y es de un empleado existente en la BD. En caso afirmativo, mostrará en pantalla todos los atributos del empleado, es decir si el empleado es de tipo zoológico muestra nombre, sueldo_base, teléfono, DNI, tipo={zoológico}}, especialidad, tasa y experiencia, si no, se muestra nombre, sueldo_base, teléfono, rango, DNI, tipo={limpieza}}, suplemento(limpieza) y zona(limpieza). En caso contrario, mandará un mensaje de error diciendo que el id introducido es incorrecto.

3.2.1.5 Listar Empleado

Concepto	Descripción
Identificador	M_EM_05
Nombre	Listar Empleado
Autor	Sara Sacó Baños.
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Todos los empleados registrados con los atributos id, nombre, sueldo_base, teléfono, rango, DNI, tipo={zoológico,limpieza}}
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Muestra todos los empleados activos e inactivos con sus atributos. En caso de no existir ninguno, se muestra una tabla vacía

3.2.1.6 Listar Empleados por Hábitat

Concepto	Descripción
Identificador	M_EM_06
Nombre	Listar Empleados por Hábitat
Autor	Sara Sacó Baños
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_habitat
Salida	Se muestran todos los empleados relacionados con el hábitat del ID introducido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se comprueba que el id_habitat introducido esta en el formato correcto. Si no lo es, se muestra un mensaje de error. Si es correcto, se muestran los atributos: id, nombre, sueldo_base, teléfono, DNI, tipo{=zoológico,limpieza}}, de los empleados registrados relacionados con el hábitat cuyo id es el introducido. En caso de no existir ninguno, se muestra una tabla vacía.

3.2.1.7 Listar empleado por especie en Habitat

Concepto	Descripción
Identificador	M_EM_07
Nombre	Listar Empleados por Especie en Hábitat
Autor	Jorge Calvo y Steven Pérez
Prioridad	Alta

Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_especie
Salida	Se muestran todos los empleados relacionados con una especie dentro de un hábitat del que está relacionado.
Precondición	No hay identificadores repetidos en la base de datos.
Postcondición	No hay identificadores repetidos en la base de datos.
Descripción	Se verifica que el id_especie introducido esta en el formato correcto. Si no lo es, se muestra un mensaje de error. Si es correcto, se muestran los atributos: id, nombre, sueldo_base, teléfono, DNI, tipo={zoológico, limpieza}}, de los empleados registrados relacionados con el hábitat donde reside la especie cuyo id es el introducido. En caso de no existir ninguno, se muestra una tabla vacía.

3.2.2 Factura

3.2.2.1 Abrir factura

Concepto	Descripción
Identificador	M_F_01
Nombre	Abrir factura
Autor	Axel Kahou He Expósito
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Vista de los casos de Uso M_V_02, M_V_03 y M_V_04

Precondición	No hay ID de facturas repetidos en la base de datos.
Postcondición	No hay ID de facturas repetidos en la base de datos. Si tiene éxito se abre la factura.
Descripción	Se carga la interfaz donde se podrán añadir/quitar pases a la nueva factura, y cerrarla.

3.2.2.2 Cerrar factura

Concepto	Descripción
Identificador	M_F_02
Nombre	Cerrar factura
Autor	Axel Kahou He Expósito
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	N/A
Salida	Mensaje de éxito/fracaso.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si hay éxito, nueva factura insertada en la BD.
Descripción	<p>Realizamos una serie de comprobaciones, primero se comprueba que la compra tiene al menos un pase, segundo que los pases añadidos estén activos, tercero comprobamos que hay cantidad disponible para cada pase. Después actualizamos la cantidad disponible de los pases, luego calculamos el precio total de la factura. Por último, se crea una factura de la venta y se muestra un mensaje de éxito con el id de la factura y el precio total de la factura.</p> <p>En caso de que no cumpla alguna de las comprobaciones, se muestra un mensaje informando del error correspondiente.</p>

3.2.2.3 Mostrar factura

Concepto	Descripción
Identificador	M_F_03
Nombre	Mostrar factura
Autor	Axel Kahou He Expósito
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_factura
Salida	precio_total, fecha_compra, lista de pases asociados a ese id_factura, cada uno con su precio y uds.
Precondición	no hay identificadores repetidos en la base de datos.
Postcondición	no hay identificadores repetidos en la base de datos.
Descripción	Comprueba que el id se encuentra en el formato correcto. Si es así, entonces comprueba que el ID introducido pertenece a una factura introducida en la BD, ya esté activa o inactiva. En caso afirmativo se muestran los atributos de la factura y su listado de pases. En caso contrario, se informa del error.

3.2.2.4 Listar facturas

Concepto	Descripción
Identificador	M_F_04
Nombre	Listar facturas
Autor	Axel Kahou He Expósito
Prioridad	Alta
Estabilidad	Alta

Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Todas las facturas registradas con los atributos fecha_compra, precio_total, id_Factura.
Precondición	No hay identificadores repetidos en la BD
Postcondición	No hay identificadores repetidos en la BD
Descripción	Muestra todas las facturas registradas, activas o inactivas, con sus atributos. En caso de no existir ninguna, se muestra una tabla vacía

3.2.2.5 Devolver factura

Concepto	Descripción
Identificador	M_F_05
Nombre	Devolver factura
Autor	Axel Kahou He Expósito
Prioridad	Baja
Estabilidad	Media
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id de la factura • id pase • número unidades a devolver
Salida	Mensaje de éxito/fracaso.
Precondición	No hay identificadores repetidos en la base de datos.
Postcondición	No hay identificadores repetidos en la base de datos.
Descripción	Se comprueba que el Id de la factura y del pase están en el formato correcto. Si es correcto, se pasa a comprobar que el ID factura es el de una factura que se encuentra en la BD y está activa. En caso de éxito, Se comprueba que el ID pase es de un pase que se encuentra en la

	<p>BD. Posteriormente, comprobamos que el pase estaba incluido en la factura. Luego se procede a actualizar el stock del pase, en caso de devolver unidades de un pase que ya estaba dado de baja, se comprueba si está dado de baja, en ese caso, se actualiza su stock y se reactiva. Se analiza si se ha devuelto una parte de la factura o toda la factura, debido a que si se devuelve una factura completa, esta pasa a estar inactiva y su precio total pasa a ser 0. En caso de no devolver una factura completa, se procede a calcular el nuevo precio total de la factura, se actualiza la cantidad disponible del pase y para finalizar se muestra un mensaje de éxito indicando que se ha devuelto la factura. En caso de error, se muestra un mensaje informando del error correspondiente.</p>
--	--

3.2.2.6 Añadir pase a factura

Concepto	Descripción
Identificador	M_F_06
Nombre	Añadir pase a factura
Autor	Axel Kahou He Expósito
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_pase • unidades del pase
Salida	Mensaje de éxito/fracaso.
Precondición	No hay identificadores repetidos en la base de datos.
Postcondición	No hay identificadores repetidos en la base de datos.
Descripción	<p>Se comprueba que los datos ingresados están en el formato correcto y que el número de unidades no sea superior a 100. En caso afirmativo, se añade el pase a la venta. Si esto se cumple, se comprueba si el pase a añadir no se encuentra ya en la factura, en caso de ya encontrarse, se actualiza su número de unidades y se comprueba que el total de unidades no sea superior a 100. En caso de ser un nuevo pase, se lo añade a la factura con su respectivo número de unidades.</p>

	Caso contrario, se informa del error.
--	---------------------------------------

3.2.2.7 Quitar pase de factura

Concepto	Descripción
Identificador	M_F_07
Nombre	Quitar pase de factura
Autor	Axel Kahou He Expósito
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_pase • número de unidades
Salida	Mensaje de éxito/fracaso.
Precondición	No hay identificadores repetidos en la base de datos.
Postcondición	No hay identificadores repetidos en la base de datos.
Descripción	Comprueba que el Id esta escrito en el formato correcto, caso correcto entonces comprueba que el ID del pase se encuentra dentro de la factura y que el número de unidades a quitar no es superior al numero de unidades que fueron solicitadas al añadir el pase. Caso afirmativo, se actualiza la factura y se informa de que se ha eliminado las unidades del pase. En caso contrario, se informa del error.

3.2.2.8 Modificar factura

Identificador	M_F_08
Nombre	Modificar Factura
Autor	Axel Kahou He Expósito

Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_factura • fecha compra
Salida	Mensaje de éxito/fracaso.
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos
Descripción	Comprueba que el Id está escrito en el formato correcto, Si es correcto, se pasa a comprobar que el ID es el de una factura que se encuentra en la BD y que está activa. Si esto es correcto, se pasa a comprobar el formato de los datos introducidos, si son correctos, se actualiza los datos correspondientes de la factura. En caso de algún error, se muestra un mensaje informando del error correspondiente

3.2.3 Pase

3.2.3.1 Alta pase

Concepto	Descripción
Identificador	M_P_01
Nombre	Alta Pase
Autor	Clara Sotillo Sotillo
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • fecha • hora • precio

	<ul style="list-style-type: none"> • cantidad_disponible • id_habitat
Salida	Mensaje de éxito/fracaso con el ID del pase creado
Precondición	No hay identificadores repetidos ni pases a la misma hora y fecha en la BD
Postcondición	No hay identificadores repetidos ni pases a la misma hora y fecha en la BD. Si tiene éxito, se inserta el pase en la BD.
Descripción	Se comprueba que los datos introducidos se encuentran en el formato correcto, comprueba que existe el id del habitat en la base de datos y esta activo. En caso de ser correcto, se crea un nuevo pase en la BD si los datos introducidos se encuentran en el formato correcto y no existe un pase con esos datos en la BD. En el caso de existir un pase con esos datos pero que se encuentre inactivo, este se reactivará. En el caso de existir y encontrarse activo no se modificará la BD y se mostrará un mensaje de error. En el caso que el id_habitat sea de un habitat que no existe o esta inactivo en la BD, se mostrará un mensaje de error. En el caso de que los datos sean erróneos tampoco se modificará la BD y se mostrará un mensaje de error

3.2.3.2 Baja Pase

Concepto	Descripción
Identificador	M_P_02
Nombre	Baja Pase
Autor	Clara Sotillo Sotillo
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_pase
Salida	Mensaje de éxito o fracaso
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos Si tiene éxito, pase dado de baja

Descripción	<p>Se comprueba que el Id introducido está en el formato correcto. En caso de éxito, se marcará como inactivo un pase cuando el usuario introduzca un id_pase que coincida con un pase en la BD que exista y esté activo.</p> <p>Si no se encuentra activo, no existe en la BD o hay un error de formato, se informa al usuario del error.</p>
--------------------	--

3.2.3.3 Mostrar Pase

Concepto	Descripción
Identificador	M_P_03
Nombre	Mostrar Pase
Autor	Clara Sotillo Sotillo
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_pase
Salida	Se muestra un pase con su información y el id del hábitat
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos
Descripción	<p>Comprueba que el id es ingresado en el formato correcto. En caso de éxito, comprueba que el id del pase se encuentre registrada en la BD. Si el pase existe, entonces se muestran todos los datos del pase. En caso de no existir se muestra un mensaje de error</p>

3.2.3.4 Listar Pases

Concepto	Descripción
Identificador	M_P_04

Nombre	Mostrar Todos los Pases
Autor	Clara Sotillo Sotillo
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Muestra la información de todos los pases: id_pase, id_habitat, fecha, hora, precio, cantidad_disponible, activo.
Precondición	No hay identificadores repetidos en la BD
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se muestra la información de todos los pases registrados en la BD. En caso de no haber pases en la BD, se muestra una tabla vacía

3.2.3.5 Modificar Pase

Concepto	Descripción
Identificador	M_P_05
Nombre	Modificar Pase
Autor	Clara Sotillo Sotillo
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_pase Uno o más de los siguientes datos: <ul style="list-style-type: none"> • fecha • hora • precio • cantidad_disponible

	<ul style="list-style-type: none"> • id_habitat
Salida	Pase con los datos actualizados.
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos Si éxito, pase actualizado
Descripción	Se comprueba que el id y los datos introducidos se encuentran en el formato correcto. En caso de éxito, se comprueba que el id del pase exista y esté activo. Si está activo, y si el campo a modificar son la fecha, hora y id hábitat, se comprueba que no existan en la BD. En caso afirmativo, se comprueba que el id_habitat pertenece a un hábitat que se encuentra en la BD y está activo. En caso de no tener errores, se modificarán los datos distintos que introduzca el usuario. En el caso de que ocurra un error se mostrará un mensaje informando del error correspondiente.

3.2.3.6 Listar Pases Por Hábitat

Concepto	Descripción
Identificador	M_P_06
Nombre	Listar Pase Por Hábitat
Autor	Clara Sotillo Sotillo
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_habitat
Salida	Se muestran todos los pases asociados a ese id_habitat con la información: fecha, hora, precio, id_pase, cantidad_disponible, activo
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos
Descripción	Se comprueba que el ID se encuentra en el formato correcto. Si es correcto, se comprueba si el id introducido existe en la BD de hábitat. Si el id hábitat existe entonces se muestra la información de los pases

	asociados al hábitat con el id_habitat introducido por el usuario. Caso contrario, se mostrará un mensaje de error indicando el error correspondiente.
--	--

3.2.4 Hábitat

3.2.4.1 Alta Hábitat

Concepto	Descripción
Identificador	M_HA_01
Nombre	Alta Hábitat
Autor	Jorge Calvo Fernández
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • Nombre • Tamaño
Salida	Mensaje de éxito con el ID del nuevo hábitat o mensaje de fracaso indicando el error producido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si éxito, se inserta un hábitat en la BD
Descripción	Comprueba que los datos introducidos son correctos, si es así, se comprueba si el nombre, que debe ser único, es de un hábitat inactivo, en cuyo caso, se reactivaría actualizando sus atributos. En caso de ser repetido, no se crearía un nuevo hábitat, mostrando un mensaje de error de que el hábitat ya existe. En caso de no existir, se introduce dentro de la base de datos el nuevo hábitat y se muestra en pantalla el ID generado. En caso contrario se muestra un mensaje donde se informa del error.

3.2.4.2 Baja Hábitat

Concepto	Descripción
Identificador	M_HA_02
Nombre	Baja Hábitat
Autor	Jorge Calvo Fernández
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_habitat
Salida	Mensaje de éxito si el hábitat existe y se da de baja correctamente. mensaje de fracaso en caso contrario, indicando el error producido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si tiene éxito, el hábitat es dado de baja.
Descripción	Comprueba que el id introducido existe y está activo, se comprueba que no tenga pases ni especies activas, si hay alguna de estas activa no se da de baja, en caso contrario, dicho hábitat pasará a estar inactivo y se desvinculan los empleados asociados. Si no es posible realizar la baja lógica, se mostrará un mensaje indicando el error producido.

3.2.4.3 Mostrar Hábitat

Concepto	Descripción
Identificador	M_HA_03
Nombre	Mostrar Hábitat
Autor	Jorge Calvo Fernández
Prioridad	Alta

Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_habitat
Salida	Atributos del hábitat con id_habitat
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se comprueba que el Id introducido está en el formato correcto y es de un hábitat existente, en cuyo caso se mostrarán sus atributos. En caso contrario, se mostrará un mensaje indicando el error.

3.2.4.4 Modificar Hábitat

Concepto	Descripción
Identificador	M_HA_04
Nombre	Modificar Hábitat
Autor	Jorge Calvo Fernández
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_habitat Uno o más de los siguientes datos: <ul style="list-style-type: none"> • nombre • tamaño.
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se comprueba el formato del id, luego se comprueba que el hábitat

	existe y se encuentra activo, si está activo, se comprueba que los datos a modificar son sintácticamente correctos, y si el campo a modificar es el nombre, se comprueba si no existe en la BD. En caso afirmativo se actualizan los datos del hábitat y se muestra un mensaje de éxito. En caso contrario se muestra un mensaje indicando el error producido.
--	--

3.2.4.5 Listar Hábitat

Concepto	Descripción
Identificador	M_HA_05
Nombre	Listar Hábitat
Autor	Jorge Calvo Fernández
Prioridad	Media
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Lista de todos los hábitats registrados.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se consulta en la BD y se devuelve una lista de todos los hábitats registrados en la BD con sus respectivos atributos. Se muestra además, si están activos o inactivos.

3.2.4.6 Vincular Empleado a Hábitat

Concepto	Descripción
Identificador	M_HA_06
Nombre	Vincular Empleado a Habitat

Autor	Jorge Calvo Fernández
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_habitat • id_empleado
Salida	Mensaje de éxito si se vincula correctamente o mensaje de fracaso indicando el error producido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Nueva vinculación creada
Descripción	Se verifica si el formato de los ID introducidos es correcto. Si es correcto se verifica que tanto el hábitat como el empleado existen y están activos. Si es así, se vincula el empleado al hábitat. Se devuelve un mensaje confirmando la vinculación o indicando el error producido.

3.2.4.7 Desvincular Empleado de Hábitat

Concepto	Descripción
Identificador	M_HA_07
Nombre	Desvincular Empleado de Hábitat
Autor	Jorge Calvo Fernández
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_habitat • id_empleado
Salida	Mensaje de éxito si se desvincula correctamente o mensaje de fracaso indicando el error producido.

Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Se realiza una desvinculación Empleado-habitat
Descripción	Se verifica el formato de los ID introducidos. Si es correcto se verifica que tanto el hábitat como el empleado existen, están activos y que el empleado está vinculado al hábitat. Si es así, se desvincula el empleado del hábitat. Se devuelve un mensaje confirmando la desvinculación o indicando el error producido.

3.2.4.8 Listar Habitats por Empleado

Concepto	Descripción
Identificador	M_HA_08
Nombre	Listar Habitats por Empleado
Autor	Jorge Calvo Fernández
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_empleado
Salida	Lista de hábitats vinculados con el empleado de id de entrada
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se verifica el formato de los ID introducidos. Si es correcto se comprueba la validez del ID introducido, mostrándose por pantalla un error en caso negativo. En caso positivo, se accede a la BD mostrando una lista de todos aquellos hábitats con el ID del empleado introducido asociado a ellos.

3.2.4.9 Calcular hábitat con más ingresos

Concepto	Descripción
Identificador	M_HA_09
Nombre	Calcular hábitat con más ingresos
Autor	Alvaro Velasco y Mariana Cucicea
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • Fecha inicio • fecha fin
Salida	El hábitat con más ingresos
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Se comprueba que las fechas introducidas tienen un formato válido. Se calcula el hábitat que más dinero ha generado en función de los pases recibidos y las facturas correspondientes a los mismos en el intervalo de tiempo indicado en la entrada. Si no hay ninguno en ese rango de fechas, se muestra una tabla vacía.

3.2.5 Especie

3.2.5.1 Alta Especie

Concepto	Descripción
Identificador	M_ES_01
Nombre	Alta Especie
Autor	Airam Martín Peraza
Prioridad	Alta
Estabilidad	Alta

Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • nombre de la especie • id_habitat.
Salida	Mensaje de éxito/fracaso .
Precondición	no hay identificadores ni nombres repetidos en la base de datos.
Postcondición	no hay identificadores ni nombres repetidos en la base de datos. Si tiene éxito, nueva especie insertada.
Descripción	Comprueba que los datos introducidos están en el formato correcto, si es así, se comprueba que exista el id de hábitat y esté activo. Se comprueba si el nombre, que debe ser único, es de una especie inactiva, en cuyo caso, se reactivaría actualizando sus atributos. En caso de ser repetido y estar activo, no se crearía una nueva especie, mostrando un mensaje de error de que la especie ya existe. En caso de no existir, se introduce dentro de la base de datos la nueva especie y se muestra en pantalla el ID generado. En caso contrario se muestra un mensaje donde se informa del error.

3.2.5.2 Baja Especie

Concepto	Descripción
Identificador	M_ES_02
Nombre	Baja Especie
Autor	Airam Martín Peraza
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_especie
Salida	Mensaje de éxito/fracaso
Precondición	no hay identificadores repetidos en la base de datos

Postcondición	no hay identificadores repetidos en la base de datos
Descripción	Se comprueba que el Id introducido está en el formato correcto, luego comprueba que existe una especie activa en la base de datos con el ID introducido. En caso afirmativo, se comprueba si hay animales activos que estén asociados a el id de la especie, si hay algún animal activo, no dejará dar de baja a la especie. Caso contrario, daremos una baja lógica de especie. En caso de algún error, se informa de este correspondiente.

3.2.5.3 Mostrar Especie

Concepto	Descripción
Identificador	M_ES_03
Nombre	Mostrar Especie
Autor	Airam Martín Peraza
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_especie
Salida	Se muestra los datos de la especie y el id del hábitat
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Comprueba que el Id introducido está en el formato correcto y es de una especie existente. En caso afirmativo, se muestran los datos de la especie. En caso contrario, se informa del error.

3.2.5.4 Modificar Especie

Concepto	Descripción
-----------------	--------------------

Identificador	M_ES_04
Nombre	Modificar Especie
Autor	Airam Martín Peraza
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_especie Uno o más de los siguientes datos: <ul style="list-style-type: none"> • Nombre • id_habitat
Salida	Mensaje de éxito/fracaso
Precondición	no hay identificadores ni nombres repetidos en la base de datos
Postcondición	no hay identificadores ni nombres repetidos en la base de datos. Si éxito, especie actualizada
Descripción	<p>se comprueba que los datos a modificar son sintácticamente correctos. En caso afirmativo, se comprueba que la especie existe y se encuentra activo, lo mismo con el id de habitat. Si está activo, y si el campo a modificar es el nombre, se comprueba si no existe en la BD. En caso afirmativo se actualizan los datos del producto y se muestra un mensaje de éxito. En caso contrario se muestra un mensaje indicando el error producido.</p>

3.2.5.5 Listar Especies

Concepto	Descripción
Identificador	M_ES_05
Nombre	Listar Especie
Autor	Airam Martín Peraza
Prioridad	Alta
Estabilidad	Alta

Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Todas las especies que estan en la BD
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos
Descripción	Muestra todas las especies registradas que se encuentren en la BD. En caso de no haber, se mostrara una tabla vacía

3.2.5.6 Listar Especie por Habitat

Concepto	Descripción
Identificador	M_ES_06
Nombre	Listar Especie por Hábitat
Autor	Airam Martín Peraza
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_habitat
Salida	Todos los datos de las Especies asociadas a ese ID del Hábitat
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos
Descripción	Se verifica el formato del ID introducido. Si es correcto se comprueba que existe, mostrándose por pantalla un error en caso negativo. En caso positivo, se accede a los datos de la Especie de la BD mostrando una lista de todas aquellas especies con el ID de el habitat introducida

	asociada a ellos. En caso contrario, se muestra un mensaje indicando el error correspondiente
--	---

3.2.6 Animal

3.2.6.1 Alta Animal

Concepto	Descripción
Identificador	M_A_01
Nombre	Alta Animal
Autor	Sergi Cristóbal Gambau
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • Nombre • tipo={acuático, no acuático}} • tipo_agua(acuático) • temperatura(acuático) • num. patas(no acuático) • id_especie
Salida	Mensaje de éxito junto con el ID del nuevo animal o mensaje de fracaso indicando el error producido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si éxito, animal dado de alta
Descripción	Comprueba que los datos introducidos son correctos, si es así, se comprueba si el id de la especie existe en la BD y está activo. Si es correcto, comprobamos si el nombre, que debe ser único, es de un animal inactivo, en cuyo caso, se reactivaría actualizando sus atributos. En caso de ser repetido, no se crearía un nuevo animal, mostrando un mensaje de error de que el animal ya existe. En caso de no existir, se introduce dentro de la base de datos el nuevo animal y se muestra en pantalla el ID generado. En caso contrario se muestra un mensaje

	donde se informa del error.
--	-----------------------------

3.2.6.2 Baja Animal

Concepto	Descripción
Identificador	M_A_02
Nombre	Baja Animal
Autor	Sofía Postigo Ruiz
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_animal
Salida	Mensaje de éxito si el animal ya existe y se da de bajar correctamente. Mensaje de fracaso en caso contrario indicando el error producido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si éxito, animal dado de baja
Descripción	Comprueba que existe un animal activo en la base de datos con el ID introducido. En caso afirmativo, se realiza una baja lógica. En caso de que se produzca algún problema, muestra un mensaje de error.

3.2.6.3 Modificar Animal

Concepto	Descripción
Identificador	M_A_03
Nombre	Modificar Animal
Autor	Sergi Cristóbal Gambau
Prioridad	Baja

Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_animal <p>Uno o más de los siguientes datos:</p> <ul style="list-style-type: none"> • nombre • tipo={acuaticos,no acuaticos}} • tipo Agua(acuático) • temperatura (acuático) • Numero Patas (no acuático) • id_Especie
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.Se actualizan los atributos del animal.
Descripción	Se comprueba que los Id han sido ingresados en el formato correcto. Si es así, en caso de modificar el id especie, se comprueba que sea el id de una especie que exista en la BD y se encuentra activo. En caso correcto, pasa a comprobar que el animal existe y se encuentra activo, si está activo, se comprueba que los datos a modificar son sintácticamente correctos, y si el campo a modificar es el nombre, se comprueba si no existe en la BD. En caso afirmativo en base al tipo de animal se actualizan los datos del animal y se muestra un mensaje de éxito. En caso contrario, se muestra un mensaje indicando el error producido.

3.2.6.4 Mostrar Animal

Concepto	Descripción
Identificador	M_A_04
Nombre	Mostrar Animal
Autor	Sofia Postigo Ruiz
Prioridad	Media
Estabilidad	Alta
Origen	GUI

Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_animal
Salida	Atributos del animal correspondientes al ID introducido, es decir: nombre, edad, dieta, sexo, activo, tipo={acuático,no acuático}, tipo agua(acuático), temperatura(acuático), num. patas(no acuático), id especie
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Comprueba el formato del id introducido y si corresponde a un animal existente, en caso afirmativo mostrará una tabla con todos los atributos del animal junto con los atributos de su tipo correspondiente y si este está activo. En caso contrario, mandará un mensaje de error diciendo que el id introducido es incorrecto o indicando el error que se haya producido.

3.2.6.5 Listar Animales

Concepto	Descripción
Identificador	M_A_05
Nombre	Listar Animales
Autor	Sergi Cristóbal Gambau
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Muestra la lista de todos los animales registrados si no se encuentra ningún error.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se consulta en la BD y se devuelve una lista de todos los animales

	registrados con la BD con sus respectivos atributos. Se muestra además, si están activos o inactivos.
--	---

3.2.6.6 Listar Animales por Especie

Concepto	Descripción
Identificador	M_A_06
Nombre	Listar Animales por Especie
Autor	Sofía Postigo Ruiz
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none">id_especie
Salida	Muestra la lista de todos los animales vinculados con la especie a la que le corresponde el ID de entrada.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se verifica el formato de los ID introducidos. Si todo esto es correcto se comprueba la validez del ID introducido, mostrando por pantalla un mensaje de error en caso negativo. En caso positivo, se accede a los datos de Animal de la BD mostrando una lista de todos aquellos animales con el ID de la especie introducido asociado a ellos, mostrando sus atributos generales y el tipo de animal. En caso contrario, se mostrará un mensaje informando del error correspondiente.

3.2.7 Departamento

3.2.7.1 Alta Departamento

Concepto	Descripción
Identificador	M_D_01
Nombre	Alta Departamento
Autor	Carlos Mayorga Santiago
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • nombre
Salida	Mensaje de éxito/fracaso junto con el id del departamento
Precondición	No existen ids repetidos ni departamentos con el mismo nombre en la base de datos.
Postcondición	No existen ids repetidos ni departamentos con el mismo nombre en la base de datos. Si tiene éxito, se inserta un nuevo departamento en la base de datos.
Descripción	<p>Se comprueba que el nombre introducido es sintácticamente correcto. Se verifica si existe un departamento activo con el mismo nombre. Si es así, se muestra un mensaje de error indicando que el departamento ya existe.</p> <p>Si existe un departamento inactivo con dicho nombre, este se reactiva. Si no existe un departamento con el nombre introducido, se inserta un nuevo registro de departamento en la base de datos con el nombre proporcionado. Se genera un mensaje de éxito que confirma que el nuevo departamento se ha registrado con éxito y se muestra el ID del departamento.</p>

3.2.7.2 Baja Departamento

Concepto	Descripción
Identificador	M_D_02
Nombre	Baja Departamento
Autor	Carlos Mayorga Santiago
Prioridad	Baja

Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_departamento
Salida	Mensaje de éxito/fracaso
Precondición	No existen ids repetidos en la base de datos.
Postcondición	No existen ids repetidos en la base de datos. Si tiene éxito, se da de baja el departamento en la base de datos.
Descripción	<p>Se verifica que el id introducido sea sintácticamente correcto. Se comprueba si el departamento con el ID proporcionado existe en la base de datos, está marcado como activo y no tiene trabajadores asociados activos. Si no cumple con estas condiciones, se muestra un mensaje de error indicando que la operación no puede ser realizada. Si las cumple, se marca el departamento como inactivo en la base de datos. Se muestra un mensaje de éxito que confirma que el departamento se ha dado de baja con éxito.</p>

3.2.7.3 Modificar Departamento

Concepto	Descripción
Identificador	M_D_03
Nombre	Modificar Departamento
Autor	Carlos Mayorga Santiago
Prioridad	Baja
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_departamento • nombre

Salida	Mensaje de éxito/fracaso
Precondición	No existen ids repetidos en la base de datos.
Postcondición	No existen ids repetidos en la base de datos. Si tiene éxito, se actualiza el departamento
Descripción	Se comprueba que los datos sean sintácticamente correctos. Se verifica si el departamento con el ID proporcionado existe en la base de datos y está activo. Si no se cumple esta condición, se muestra un mensaje de error indicando que la operación no puede ser realizada. Si se cumple, se comprueba si el nuevo nombre pertenece a un departamento en la base de datos. Si no es así, se actualiza el nombre del departamento en la base de datos con el nuevo nombre proporcionado. Se genera un mensaje de éxito que confirma que el departamento se ha modificado con éxito. En caso de que alguna condición no se cumpla, se muestra un mensaje informando del error.

3.2.7.4 Mostrar Departamento

Concepto	Descripción
Identificador	M_D_04
Nombre	Mostrar Departamento
Autor	Carlos Mayorga Santiago
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_departamento
Salida	nombre
Precondición	No existen ids repetidos en la base de datos.
Postcondición	No existen ids repetidos en la base de datos
Descripción	Se comprueba que el id tenga formato correcto. Se verifica si el departamento con el ID proporcionado existe en la base de datos. Si no se cumple esta condición, se muestra un mensaje de error indicando que el departamento no existe. En caso contrario, se muestra por pantalla el nombre del departamento.

3.2.7.5 Listar Departamentos

Concepto	Descripción
Identificador	M_D_05
Nombre	Listar Departamentos
Autor	Carlos Mayorga Santiago
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Lista de departamentos activos e inactivos
Precondición	No existen ids repetidos en la base de datos.
Postcondición	No existen ids repetidos en la base de datos.
Descripción	Se muestra una lista con todos los departamentos almacenados en la base de datos.

3.2.7.6 Calcular Nomina Departamento

Concepto	Descripción
Identificador	M_D_06
Nombre	Calcular Nomina Departamento
Autor	Jorge Calvo Fernández
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_departamento

Salida	Suma total de los salarios de todos los trabajadores del departamento indicado.
Precondición	No hay identificadores repetidos en la BD
Postcondición	No hay identificadores repetidos en la BD
Descripción	<p>Se verifica la existencia y si está activo el departamento con el id_departamento proporcionado en la base de datos. Luego, se obtienen todos los trabajadores vinculados a dicho departamento. Una vez obtenidos, se comprueba que los trabajadores existen y están activos y posteriormente se suman todos los salarios de esos trabajadores.</p> <p>Para calcularlo se calculará: Nómina total del departamento = sum (salario individual de cada trabajador activo del departamento)</p> <p>Finalmente, la suma total calculada se muestra en la GUI. En caso de error en el cálculo o si el departamento está inactivo o no existe, se muestra un mensaje de error.</p>

3.2.8 Trabajador

3.2.8.1 Alta Trabajador

Concepto	Descripción
Identificador	M_EM_01
Nombre	Alta Trabajador
Autor	Álvaro Velasco
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	Base de datos
Entrada	<ul style="list-style-type: none"> • DNI • nombre • sueldo

	<ul style="list-style-type: none"> • teléfono • tipo={completo,parcial}} • h_extra(completo) • euros/h_extra(completo) • horas(parcial) • euros/h(parcial) • id_departamento
Salida	Mensaje de éxito con el ID del nuevo trabajador o mensaje de fracaso
Precondición	No hay identificadores repetidos ni trabajadores con el mismo DNI en la BD.
Postcondición	No hay identificadores repetidos ni trabajadores con el mismo DNI en la BD. Si tiene éxito, se inserta un nuevo trabajador en la base de datos.
Descripción	<p>Se comprueba que los datos sean sintácticamente correctos.</p> <p>Se verifica que el departamento con id dado exista y esté activo.</p> <p>Se verifica si existe un trabajador activo con el mismo DNI. Si es así, se muestra un mensaje de error indicando que el trabajador ya existe.</p> <p>Si existe un trabajador inactivo con dicho DNI, este se reactiva. Si no existe un trabajador con el DNI, se inserta un nuevo registro de trabajador . Se genera un mensaje de éxito que confirma que el nuevo departamento se ha registrado con éxito y se muestra el ID del departamento.</p>

3.2.8.2 Baja Trabajador

Concepto	Descripción
Identificador	M_EM_02
Nombre	<i>Baja Trabajador</i>
Autor	Álvaro Velasco
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	Base de datos
Entrada	<ul style="list-style-type: none"> • id_trabajador
Salida	Mensaje de éxito si el trabajador existe, está activo y se da de baja

	correctamente, o mensaje de fracaso, indicando el error producido, en caso contrario.
Precondición	No hay identificadores repetidos ni trabajadores con el mismo DNI en la BD.
Postcondición	No hay identificadores repetidos ni trabajadores con el mismo DNI en la BD. Si tiene éxito, será dado de baja el trabajador en la base de datos.
Descripción	Comprueba que el id está en formato correcto. Comprueba si existe un trabajador activo en la base de datos con el ID introducido. En caso afirmativo, pasa a estar inactivo. En caso contrario, se muestra un mensaje informando del error.

3.2.8.3 Modificar Trabajador

Concepto	Descripción
Identificador	M_EM_03
Nombre	<i>Modificar Trabajador</i>
Autor	Álvaro Velasco
Prioridad	Baja..
Estabilidad	Alta.
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • El <i>id_trabajador</i> <p>Uno o más de los siguientes datos:</p> <ul style="list-style-type: none"> • <i>DNI</i> • <i>nombre</i> • <i>sueldo</i> • <i>teléfono</i> • <i>tipo</i>{=[completo,parcial]} • <i>h_extra</i>(completo) • <i>euros/h_extra</i>(completo) • <i>horas</i>(parcial) • <i>euros/h</i>(parcial) • <i>id_departamento</i>

Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la base de datos.
Postcondición	No hay identificadores repetidos en la base de datos. Si hay éxito, se actualiza.
Descripción	Comprueba que el trabajador existe y se encuentra activo. En ese caso, comprueba que los datos introducidos son sintácticamente correctos. Si el campo a modificar es el DNI, se comprueba que no pertenezca a un trabajador en la BD, y si es el id_departamento, se comprueba que éste pertenezca a un departamento en la BD. Si esto se cumple, en base al tipo de Trabajador se actualizan los datos del trabajador correspondiente y se muestra un mensaje de éxito. En caso de que alguna condición no se cumpla, se muestra un mensaje informando del error.

3.2.8.4 Mostrar Trabajador

Concepto	Descripción
Identificador	M_EM_04
Nombre	Mostrar Trabajador.
Autor	Álvaro Velasco
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_trabajador
Salida	Se muestran los atributos del trabajador con id_trabajador.
Precondición	No hay identificadores repetidos en la base de datos.
Postcondición	No hay identificadores repetidos en la base de datos.
Descripción	Comprueba si el ID introducido está en el formato correcto, corresponde con algún trabajador existente en la base de datos, ya esté activo o inactivo. En caso afirmativo, mostrarán por pantalla sus atributos. En caso contrario, se muestra un mensaje informando del error.

3.2.8.5 Listar Trabajadores

Concepto	Descripción
Identificador	M_EM_05
Nombre	Listar Trabajadores
Autor	Álvaro Velasco
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Muestra una lista de todos los trabajadores existentes, junto con sus atributos si no encuentra ningún error.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se consulta en la BD y se muestran todos los trabajadores registrados junto con sus atributos. Se muestra además si están activos o inactivos. En caso de no existir ninguno, no muestra nada.

3.2.8.6 Listar Trabajadores por Departamento

Concepto	Descripción
Identificador	M_EM_06
Nombre	Listar Trabajadores por Departamento
Autor	Álvaro Velasco
Prioridad	Alta

Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_departamento
Salida	Lista de todos los trabajadores vinculados con el departamento del ID introducido.
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Se verifica que el formato y validez del ID introducido y que este sea >0. Muestra todos los trabajadores que trabajan en el departamento introducido y están registrados, junto con sus atributos, ya estén activos o inactivos. En caso de no existir ninguno, no muestra nada. En caso de que alguna de las verificaciones vaya mal se muestra un mensaje de error.

3.2.8.7 Calcular Sueldo Trabajador

Concepto	Descripción
Identificador	M_EM_07
Nombre	Calcular Sueldo Trabajador
Autor	Jorge Calvo Fernández
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_trabajador
Salida	Muestra el sueldo total calculado del trabajador.
Precondición	No hay identificadores repetidos en la base de datos..
Postcondición	No hay identificadores repetidos en la base de datos.

Descripción	<p>Se consulta en la base de datos el trabajador con el id_trabajador proporcionado. Luego, se obtienen los siguientes datos del trabajador: tipo (completo/parcial), sueldo, h_extra y euros/h_extra (si es completo), horas y euros/h (si es parcial).</p> <p>Dependiendo del tipo de trabajador (completo o parcial), se procede a calcular el sueldo:</p> <p>Si el trabajador es de tipo completo:</p> <ul style="list-style-type: none"> • $\text{Sueldo total} = \text{sueldo} + (\text{h_extra} * \text{euros/h_extra})$ <p>Si el trabajador es de tipo parcial:</p> <ul style="list-style-type: none"> • $\text{Sueldo total} = \text{horas} * \text{euros/h}$ <p>Finalmente, el sueldo total calculado se modificara en el campo sueldo del trabajador y a continuación se muestra en la GUI. En caso de error en el cálculo o si el trabajador no existe, se muestra un mensaje de error.</p>
--------------------	---

3.2.9 Venta

3.2.9.1 Abrir Venta

Identificador	M_V_01
Nombre	Abrir Venta
Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_trabajador
Salida	Vista de los casos de Uso M_V_02, M_V_03 y M_V_04
Precondición	No hay identificadores repetidos en la BD.
Postcondición	No hay identificadores repetidos en la BD. Si tiene éxito, se abre la

	venta
Descripción	Comprobamos que el ID del Trabajador ha sido ingresado en el formato correcto. En caso afirmativo, carga la interfaz donde se podrán añadir/quitar productos a la nueva venta, y cerrarla. En caso negativo, informamos del error

3.2.9.2 Añadir producto a venta

Identificador	M_V_02
Nombre	Añadir producto a venta
Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_producto • unidades del producto
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Se comprueba que los datos ingresados están en el formato correcto y que el número de unidades no es superior a 100. Si esto se cumple, se comprueba si el producto a añadir no se encuentra ya en la venta, en caso de ya encontrarse, se actualiza su número de unidades y se comprueba que el total de unidades no sea superior a 100. En caso de ser un nuevo producto, se lo añade a la venta con su respectivo número de unidades. En caso contrario, se informa del error.

3.2.9.3 Quitar producto de venta

Identificador	M_V_03
Nombre	Quitar producto de Venta

Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_producto • número de unidades
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Comprueba que el Id está escrito en el formato correcto, caso correcto entonces comprueba que el ID del producto se encuentra dentro de la venta y que el número de unidades a quitar no es superior al número de unidades que fueron solicitadas al añadir el producto.. En caso afirmativo, se actualiza la venta y se informa de que se ha eliminado las unidades del producto. Además, si se eliminan todas las unidades del producto, éste se elimina de la venta En caso contrario, se informa del error

3.2.9.4 Cerrar Venta

Identificador	M_V_04
Nombre	Cerrar Venta
Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	N/A
Salida	Mensaje de éxito/fracaso

Precondición	No hay identificadores repetidos en la BD
Postcondición	No hay identificadores repetidos en la BD. Si hay éxito, nueva venta insertada en la BD
Descripción	Se realiza varias comprobaciones, primero se comprueba que el id del trabajador pertenece a un trabajador activo registrado en la BD. a continuación se comprueba que los productos añadidos se encuentren activos o existan, luego comprueba si la compra contiene al menos un producto y por último comprueba si el número de unidades de cada producto es menor o igual al stock del producto. En caso afirmativo, se actualiza el stock de cada producto. Una vez realizado esto, calculamos el precio total de la venta y se muestra un mensaje de éxito con el ID asignado a la compra y el precio total calculado. Caso contrario, se muestra un mensaje informando del error correspondiente.

3.2.9.5 Mostrar Venta

Identificador	M_V_05
Nombre	Mostrar Venta
Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_venta
Salida	<i>precio_total, fecha, metodo_pago, id_trabajador, nombre</i> (Trabajador), lista de productos asociados a ese id_venta, cada uno con su <i>precio</i> y <i>uds</i>
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Comprueba que el id se encuentra en el formato correcto. Si es así, comprueba que el ID introducido pertenece a una venta introducida en la BD, ya esté activa o inactiva. En caso afirmativo, se muestran los atributos de la venta y su listado de productos. En caso contrario, se

	informa del error.
--	--------------------

3.2.9.6 Listar Ventas

Identificador	M_V_06
Nombre	Listar Ventas
Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Todas las ventas registradas con los atributos fecha, precio_total, método de pago, id_Venta y id_Trabajador
Precondición	No hay identificadores repetidos en la BD
Postcondición	No hay identificadores repetidos en la BD.
Descripción	Muestra todas las ventas registradas, activas o inactivas, con sus atributos. En caso de no existir ninguna, se muestra una tabla vacía

3.2.9.7 Listar Ventas por Empleado

Identificador	M_V_07
Nombre	Listar ventas por Empleado
Autor	Steven Pérez Salinas
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI

Entrada	<ul style="list-style-type: none"> • id_trabajador
Salida	Todas las ventas registradas con los atributos fecha, precio_total, método de pago, id_Venta y id_Trabajador
Precondición	El empleado existe en la base de datos.
Postcondición	El id_trabajador existe en la BD. Si existe, se muestra la información de todas las ventas asociadas a ese trabajador.
Descripción	Comprueba que el ID introducido está en el formato correcto. En caso correcto, comprueba que existe un trabajador activo en la BD con el ID introducido. En caso afirmativo, muestra todas las ventas, activas o inactivas, realizadas por el trabajador (si no hay muestra una tabla vacía). En caso contrario, se informa del error.

3.2.9.8 Devolución Venta

Identificador	M_V_08
Nombre	Devolver Venta
Autor	Steven Pérez Salinas
Prioridad	Baja
Estabilidad	Media
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_venta • id_producto • cantidad a devolver en el producto
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Se comprueba que el Id es ingresado en el formato correcto. Si es correcto, se pasa a comprobar que el ID es el de una venta que se encuentra en la BD y que está activa, también se comprueba que el producto existe en la BD. A continuación, se procede a actualizar el stock de el productos asociado a la venta, en caso de devolver unidades de un producto que ya estaba dado de baja, se comprueba si

	<p>está dado de baja en ese caso, se actualiza su stock y se reactiva. A continuación se analiza si se ha devuelto una parte de la venta o toda la venta, debido a que si se devuelve una venta completa, esta pasa a estar inactiva y su precio total pasa a ser 0. En caso de no devolver una venta completa, se procede a calcular el nuevo precio total de la venta y para finalizar se muestra un mensaje de éxito indicando que se ha devuelto la venta. En caso de error, se muestra un mensaje informando del error correspondiente</p>
--	---

3.2.9.9 Modificar Venta

Identificador	M_V_09
Nombre	Modificar Venta
Autor	Steven Pérez Salinas
Prioridad	Baja
Estabilidad	Media
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_venta <p>Uno o más de los siguientes datos:</p> <ul style="list-style-type: none"> • id_trabajador • método de pago • fecha
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	<p>Comprueba que el Id está escrito en el formato correcto. Si es correcto, se pasa a comprobar que el ID es el de una venta que se encuentra en la BD y que está activa, Seguidamente se comprueba que el ID del trabajador se encuentre dentro de la BD y esta activo. Si esto es correcto, se pasa a comprobar el formato de los datos introducidos, si son correctos, se actualiza los datos correspondientes de la venta. En caso de algún error, se muestra un mensaje informando del error correspondiente</p>

3.2.10 Producto

3.2.10.1 Alta de Producto

Concepto	Descripción
Identificador	M_PR_01
Nombre	Alta Producto
Autor	Mariana Cucicea
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	Base de datos
Entrada	<ul style="list-style-type: none"> • Nombre • Precio • Stock • id_marca_producto • tipo={alimentación, merchandasing}} • tipo(alimentación) • categoria(merchandasing) • ed.Limitada(merchandasing)
Salida	Mensaje de éxito con el ID del nuevo producto o mensaje de fracaso indicando el error producido.
Precondición	No hay identificadores ni nombres repetidos en la base de datos
Postcondición	No hay identificadores ni nombres repetidos en la base de datos; si éxito, nuevo producto insertado
Descripción	Comprueba que los datos introducidos son correctos, si es así, se comprueba que exista el id de marca de producto y esté activo. Se comprueba si el nombre, que debe ser único, es de un producto inactivo, en cuyo caso, se reactivaría actualizando sus atributos. En caso de ser repetido y estar activo, no se crearía un nuevo producto,

	mostrando un mensaje de error de que el producto ya existe. En caso de no existir, se introduce dentro de la base de datos el nuevo producto y se muestra en pantalla el ID generado. En caso contrario se muestra un mensaje donde se informa del error.
--	---

3.2.10.2 Baja Producto

Concepto	Descripción
Identificador	M_PR_02
Nombre	Baja Producto
Autor	Mariana Cucicea
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_producto
Salida	Mensaje de éxito si el producto existe y se da de baja correctamente. mensaje de fracaso en caso contrario, indicando el error producido.
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos; si éxito, producto dado de baja
Descripción	Se comprueba que el Id introducido está en el formato correcto, el id introducido existe y está activo. En caso de éxito, se realiza una baja lógica del producto y se muestra un mensaje indicando que se ha dado de baja. En caso contrario, se mostrará un mensaje indicando el error producido.

3.2.10.3 Mostrar Producto

Concepto	Descripción
Identificador	M_PR_03

Nombre	Mostrar Producto
Autor	Mariana Cucicea
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_producto
Salida	id_producto, activo, nombre, precio, stock, id_marca_producto
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Se comprueba que el Id introducido está en el formato correcto y es de un producto existente, en cuyo caso se mostrarán sus atributos. En caso contrario, se mostrará un mensaje indicando el error.

3.2.10.4 Modificar Producto

Concepto	Descripción
Identificador	M_PR_04
Nombre	Modificar Producto
Autor	Mariana Cucicea
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_producto <p>Uno o más de los siguientes datos:</p> <ul style="list-style-type: none"> • nombre • Precio • Stock • id_marca_producto

	<ul style="list-style-type: none"> • tipo={alimentación, merchandasing}} • tipo(alimentación) • categoria(merchandasing) • ed.Limitada(merchandasing)
Salida	Mensaje de éxito/fracaso
Precondición	no hay identificadores ni nombres repetidos en la base de datos
Postcondición	no hay identificadores ni nombres repetidos en la base de datos; si éxito, producto actualizado
Descripción	se comprueba que los datos a modificar son sintácticamente correctos. En caso afirmativo, se comprueba que el producto existe y se encuentra activo, lo mismo con el id de marca de producto. Si está activo, y si el campo a modificar es el nombre, se comprueba si no existe en la BD. En caso afirmativo en base al tipo de producto se actualizan los datos del producto y se muestra un mensaje de éxito. En caso contrario se muestra un mensaje indicando el error producido.

3.2.10.5 Listar Productos

Concepto	Descripción
Identificador	M_PR_05
Nombre	Listar Producto
Autor	Mariana Cucicea
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	N/A
Salida	Lista de todos los productos registrados con atributos id, activo, nombre, precio, stock, id_marca_producto
Precondición	no hay identificadores repetidos en la base de datos
Postcondición	no hay identificadores repetidos en la base de datos
Descripción	Se consulta en la BD y se devuelve una lista de todos los productos

	datos de alta con sus respectivos atributos.
--	--

3.2.10.6 Listar Productos por Marcas de Producto

Concepto	Descripción
Identificador	M_PR_06
Nombre	Listar Productos por Marcas de Producto
Autor	Mariana Cucicea
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_marca_producto
Salida	Lista de productos agrupados por marca de producto
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Se verifica el formato del ID introducido. Si es correcto se comprueba que existe, mostrándose por pantalla un error en caso negativo. En caso positivo, se accede a los datos de Producto de la BD mostrando una lista de todos aquellos productos con el ID de la marca introducida asociada a ellos.

3.2.11 Marca de Producto

3.2.11.1 Alta Marca de Producto

Concepto	Descripción
----------	-------------

Identificador	M_MP_01
Nombre	Alta Marca del Producto
Autor	Carlo Dubini
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	Base de datos
Entrada	<ul style="list-style-type: none"> • Nombre
Salida	Mensaje de éxito/fracaso junto con el id de la marca
Precondición	No hay identificadores ni nombres repetidos en la base de datos
Postcondición	No hay identificadores ni nombres repetidos en la base de datos; si éxito, nueva marca de producto insertada
Descripción	<p>El usuario ingresa el nombre de la nueva marca en la interfaz gráfica de usuario.</p> <p>Se verifica si existe una marca del producto activo con el mismo nombre. Si se cumple esta condición, se muestra un mensaje de error indicando que ya existe.</p> <p>Si la marca del producto tiene el mismo nombre, pero está inactiva se procede con la reactivación.</p> <p>Si no existe una marca del producto con el mismo nombre, se inserta un nuevo registro en la base de datos con el nombre proporcionado.</p> <p>Se genera un mensaje de éxito que confirma que la nueva marca del producto se ha registrado con éxito y se muestra su ID.</p>

3.2.11.2 Baja Marca de Producto

Concepto	Descripción
Identificador	M_MP_02
Nombre	Baja Marca del Producto
Autor	Carlo Dubini

Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_marca_producto
Salida	Mensaje de éxito/fracaso junto con el id de la marca. En el caso de que fracase, se explica la razón de porque no se ha podido dar de baja.
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos; si éxito, marca de producto dado de baja
Descripción	El usuario ingresa el id de la marca en la interfaz gráfica de usuario. Comprueba que la marca del producto existe, está activa y no tiene productos activos. Si se cumple este caso, se hará una baja lógica la marca en la BD y se desvincula automáticamente la marca del producto con los proveedores. En el caso contrario, se mostrará un mensaje mostrando la razón del error.

3.2.11.3 Modificar Marca de Producto

Concepto	Descripción
Identificador	M_MP_03
Nombre	Modificar Marca del Producto
Autor	Carlo Dubini
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_marca_producto nombre
Salida	Mensaje de éxito/fracaso junto con los datos modificados

Precondición	No hay identificadores ni nombres repetidos en la base de datos
Postcondición	No hay identificadores ni nombres repetidos en la base de datos; si éxito, marca de producto actualizada
Descripción	En ese caso, comprueba que los datos introducidos son sintácticamente correctos. Comprueba si la marca existe y está activa, y que el nuevo nombre no esté repetido con los demás datos de la BD. Si se cumple, se modifica la marca del producto en la BD. En caso contrario, se lanza un mensaje informando del error.

3.2.11.4 Mostrar Marca de Producto

Concepto	Descripción
Identificador	M_MP_04
Nombre	Mostrar Marca del Producto
Autor	Carlo Dubini
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> • id_marca_producto
Salida	nombre de la marca junto con su id
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Comprueba si el ID introducido es sintácticamente correcto y si se encuentra en la BD de Marca del Producto. En caso afirmativo, mostrará en pantalla sus atributos. En caso contrario, se muestra un mensaje informando del error.

3.2.11.5 Listar Marca de Producto

Concepto	Descripción
Identificador	M_MP_05
Nombre	Listar Marca de Producto
Autor	Carlo Dubini
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Todas las Marcas registradas con el atributo nombre
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Muestra todas las marcas registradas y sus atributos. En caso de no existir ninguna, se muestra una tabla vacía

3.2.11.6 Listar Marcas por Proveedor

Identificador	M_MP_06
Nombre	Listar Marcas por Proveedor
Autor	Carlo Dubini
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_proveedor
Salida	Todas las Marcas vinculados con el proveedor con sus respectivos atributos

Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Comprueba que el ID introducido esta en el formato correcto. En caso correcto, comprueba que existe un proveedor en la BD con el ID introducido. En caso afirmativo, muestra todas las marcas vinculadas al proveedor (si no hay muestra una tabla vacía). En caso contrario, se informa del error.

3.2.12 Proveedor

3.2.12.1 Alta Proveedor

Concepto	Descripción
Identificador	M_P_01
Nombre	Alta Proveedor
Autor	Sergi Cristóbal Gambau
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • Nombre • Teléfono • CIF
Salida	Mensaje de éxito/fracaso junto con el id del nuevo proveedor.
Precondición	No hay identificadores ni CIFs repetidos en la base de datos
Postcondición	No hay identificadores ni CIFs repetidos en la base de datos; si éxito, nuevo proveedor insertado
Descripción	El usuario ingresa el CIF del nuevo proveedor en la interfaz gráfica de usuario.

	<p>Se comprueba que los datos sean sintácticamente correctos.</p> <p>Se verifica si existe un proveedor activo con el mismo CIF. Si se cumple esta condición, se muestra un mensaje de error indicando que el proveedor ya existe.</p> <p>Si el proveedor tiene el mismo CIF, pero está inactivo se procede con la reactivación.</p> <p>Si no existe un proveedor con el mismo CIF, se inserta un nuevo registro de departamento en la base de datos con el CIF proporcionado.</p> <p>Se genera un mensaje de éxito que confirma que el nuevo proveedor se ha registrado con éxito y se muestra el ID del proveedor .</p>
--	---

3.2.12.2 Baja Proveedor

Concepto	Descripción
Identificador	M_P_02
Nombre	Baja Proveedor
Autor	Sofía Postigo Ruiz
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> id_proveedor
Salida	Mensaje de éxito si el proveedor existe y se da de baja correctamente. Si se produce algún problema se muestra un mensaje indicando el error producido.
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos; si éxito, proveedor dado de baja
Descripción	El usuario ingresa el ID del proveedor que desea dar de baja en la interfaz gráfica de usuario.

	<p>Se comprueba si el proveedor con el ID proporcionado existe en la base de datos y está marcado como activo. Si no cumple con estas condiciones, se muestra un mensaje de error indicando que la operación no puede ser realizada.</p> <p>En caso afirmativo, se marca el proveedor como inactivo en la base de datos y se desvinculan las marcas de producto asociadas a él.</p> <p>Se muestra un mensaje de éxito que confirma que el proveedor se ha dado de baja con éxito.</p>
--	---

3.2.12.3 Modificar Proveedor

Concepto	Descripción
Identificador	M_P_03
Nombre	Modificar Proveedor
Autor	Sergi Cristóbal Gambau
Prioridad	Baja
Estabilidad	Alta
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_proveedor <p>Uno o más de los siguientes datos:</p> <ul style="list-style-type: none"> • CIF • nombre • teléfono
Salida	Mensaje de éxito/fracaso
Precondición	No hay identificadores ni CIF repetidos en la base de datos
Postcondición	No hay identificadores ni CIF repetidos en la base de datos; si éxito, proveedor modificado
Descripción	Comprueba que los datos introducidos son sintácticamente correctos. Comprueba si el proveedor existe y está activo. En caso de modificar el CIF, verifica que no esté repetido con los demás datos de la BD. Si se cumple, se modifica el proveedor en la BD. En caso contrario, se lanza un mensaje informando del error.

3.2.12.4 Mostrar Proveedor

Concepto	Descripción
Identificador	M_P_04
Nombre	Mostrar Proveedor
Autor	Sofía Postigo Ruiz
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_proveedor
Salida	En caso de operación exitosa la salida sería el id, activo, teléfono, nombre y CIF del proveedor cuyo id ha sido introducido.
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	<p>El usuario ingresa el ID del proveedor que desea ver en la interfaz gráfica de usuario.</p> <p>Se comprueba que los datos sean sintácticamente correctos.</p> <p>Se verifica si el proveedor con el ID proporcionado existe en la base de datos. Si no se cumple esta condición, se muestra un mensaje de error indicando que el proveedor no existe.</p> <p>En caso afirmativo, se muestra el id, activo, teléfono, nombre y CIF del proveedor</p>

3.2.12.5 Listar Proveedores

Concepto	Descripción
Identificador	M_P_05

Nombre	Listar Proveedores
Autor	Sergi Cristóbal Gambau
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	N/A
Salida	Lista de proveedores activos e inactivos
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	<p>El usuario inicia la operación desde la interfaz gráfica de usuario sin necesidad de proporcionar ninguna entrada.</p> <p>La operación recupera todos los proveedores almacenados en la base de datos.</p> <p>Se muestra en la pantalla del usuario una lista de todos los proveedores con su id, activo, nombre, teléfono y CIF</p>

3.2.12.6 Listar Proveedores por Marca de Producto

Concepto	Descripción
Identificador	M_P_06
Nombre	Listar Proveedores por Marca de Producto
Autor	Sofía Postigo Ruiz
Prioridad	Alta
Estabilidad	Alta
Origen	GUI
Destino	GUI
Entrada	<ul style="list-style-type: none"> id_marca_producto

Salida	Lista de proveedores con su marca de producto
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos
Descripción	Comprueba que el ID introducido esta en el formato correcto. En caso correcto, comprueba que existe una marca de producto en la BD con el ID introducido. En caso afirmativo, muestra todas las marcas vinculadas al proveedor (si no hay muestra una tabla vacía). En caso contrario, se informa del error.

3.2.12.7 Vincular Proveedor con Marca de Producto

Concepto	Descripción
Identificador	M_P_07
Nombre	Vincular Proveedores con Marca de Producto
Autor	Sofia Postigo Ruiz
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_proveedor • id_marcaProducto
Salida	Mensaje de éxito si se vincula correctamente o mensaje de fracaso indicando el error producido.
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos; si éxito, tendremos marca de producto vinculada a proveedor
Descripción	Se verifica el formato de los ID introducidos. Si es correcto se verifica que tanto el proveedor como la marca de producto existen y están activos. Si es así se vincula la marca de producto al proveedor. Se devuelve un mensaje confirmando la vinculación o indicando el error producido.

3.2.12.8 Desvincular Proveedor con Marca de Producto

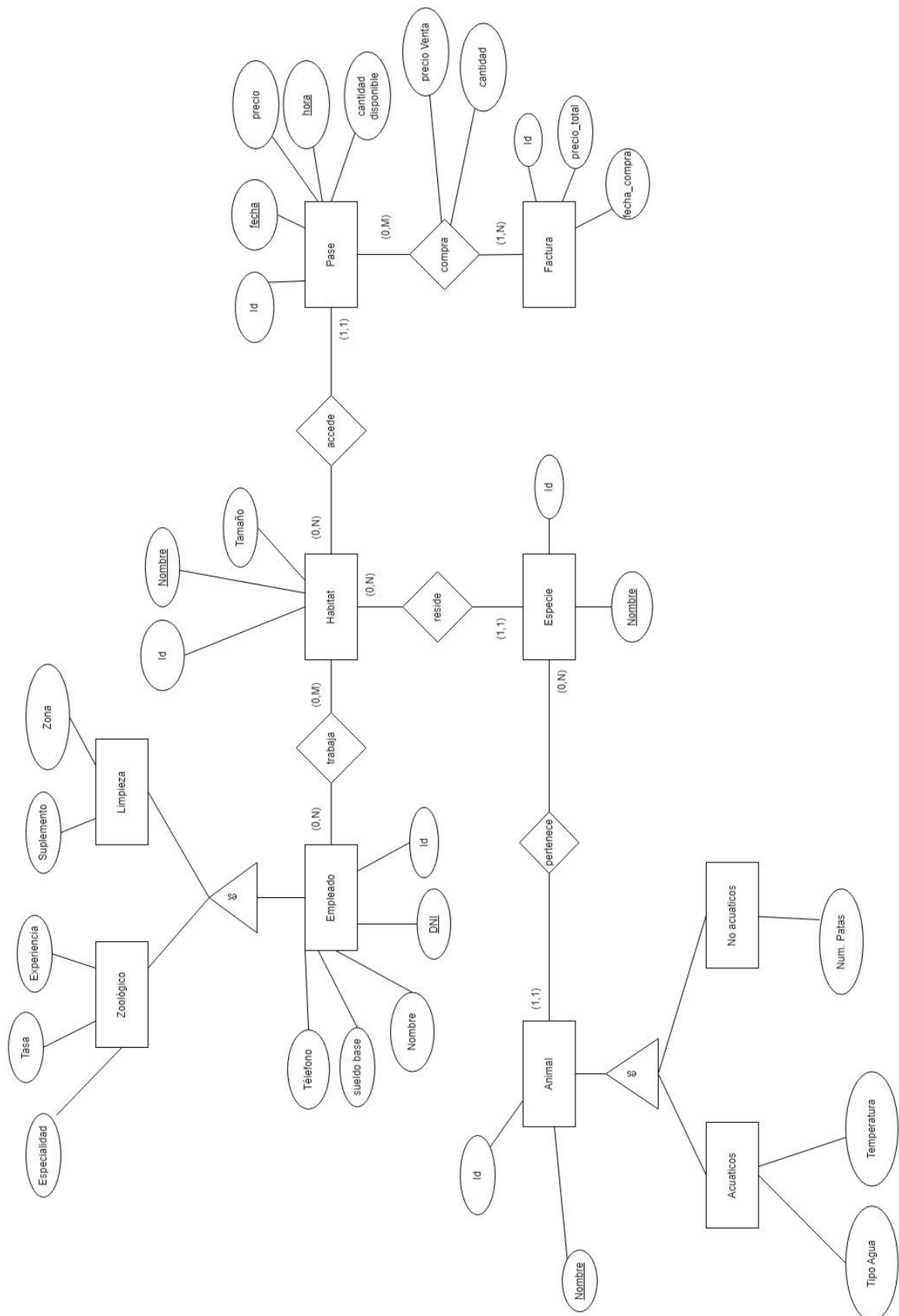
Concepto	Descripción
Identificador	M_P_08
Nombre	Desvincular Proveedor con Marca de Producto
Autor	Sofia Postigo Ruiz
Prioridad	Alta
Estabilidad	Media
Origen	GUI
Destino	BD
Entrada	<ul style="list-style-type: none"> • id_proveedor • id_marcaProducto
Salida	Mensaje de éxito si se desvincula correctamente o mensaje de fracaso
Precondición	No hay identificadores repetidos en la base de datos
Postcondición	No hay identificadores repetidos en la base de datos; si éxito, marca de producto estará desvinculada de proveedor
Descripción	Se verifica el formato de los ID introducidos. Si es correcto se verifica que tanto el proveedor como la marca de producto existen, están activos y que la marca de producto está vinculada del proveedor. Se devuelve un mensaje confirmando la desvinculación o indicando el error producido.

3.3 Requisitos de rendimiento

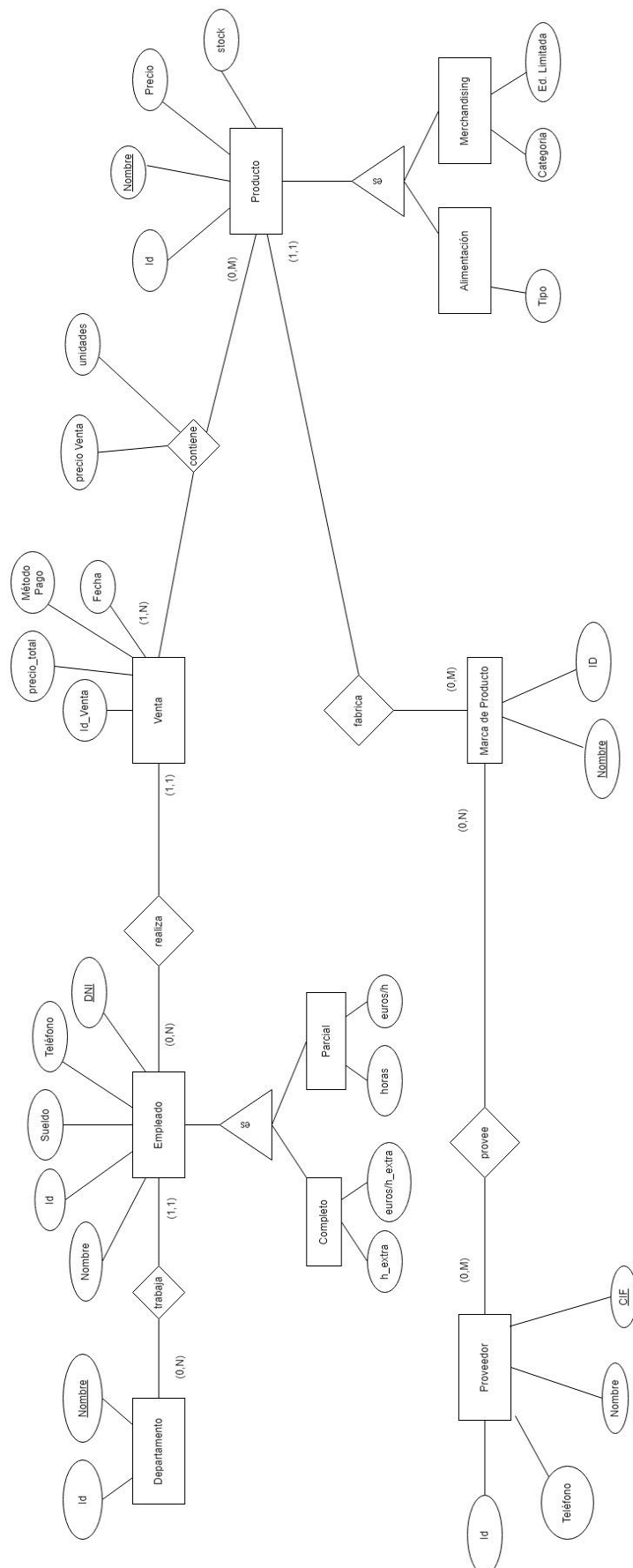
Es necesario contar con una conexión sólida hacia la base de datos que se aloja en las máquinas locales. Debe ser capaz de manejar la demanda de todos los usuarios que intenten conectarse al mismo tiempo, teniendo en cuenta que esta demanda no será muy elevada, dado que se trata de un cine local. Sin embargo, debe garantizar que no se generen conflictos de ningún tipo y también debe gestionar la concurrencia de manera adecuada.

3.4 Modelo de Dominio

3.4.1. Modelo de Dominio DAO



3.4.2 Modelo de Dominio JPA



3.5 Restricciones de diseño

Algunas de las restricciones del diseño de la aplicación se mencionan en el apartado de restricciones, pero otras de ellas serían:

- Este sistema de gestión de zoológicos debe necesariamente utilizar los siguientes patrones de diseño:
 - Transfer Object
 - Transfer Object Assembler (TOA)
 - Application Service (SA)
 - Data Access Object (DAO)
 - Service to Worker
- El sistema gestor de la tienda hará uso de los mismos patrones exceptuando el Data Access Object (DAO), añadiendo además los siguientes:
 - Domain Store
 - Business Store
- Los diagramas de clase se realizan usando el lenguaje UML de modelado.
- Se utilizará la arquitectura multicapa para ambos el gestor de la tienda y el zoológico.
- El código de la aplicación se implementará usando el lenguaje de programación Java.
- Se utilizarán queries para la primera parte del proyecto.
- Se utilizará el polimorfismo en ambas partes del proyecto.
- La persistencia de los datos se hará en formato relacional.
- MySQL será el sistema gestor de base de datos (SGBD) utilizado para la base de datos de la aplicación.
- Se utilizará GitHub como sistema de control de versiones, con dos repositorios diferentes para gestionar la documentación y el código del proyecto por separado, utilizándolo también a través de IBM RSAD.

3.6 Atributos del sistema

La funcionalidad del programa estará acorde a los casos de uso en todo momento.

- **Fiabilidad:** El sistema debe asegurar de que las funcionalidades se ejecuten correctamente y que la base de datos funcione correctamente.
- **Mantenibilidad:** El sistema debe de estar desarrollado para poder permitir la corrección de errores de una manera sencilla y eficiente. Por lo que para conseguir eso, se deberá diseñar de forma modular para poder aislar los fallos y evitar así que no se propaguen.
- **Usabilidad:** El sistema debe proveer al usuario de una interfaz clara e intuitiva para poder facilitar el uso de la aplicación.

Especificación de Requisitos Software

- **Escalabilidad:** El sistema debe de tener la capacidad de añadir nuevas funcionalidades.
- **Testabilidad:** El sistema debe de permitir la ejecución de pruebas para poder facilitar la búsqueda de errores dentro del proyecto.