

# Especificación de Requisitos de Software

IEEE Std. 830-1998

## **Proyecto Individual**

Autor: Luis Jaramillo Pulido

# ÍNDICE

## **1. Introducción**

- 1.1. Propósito
- 1.2. Alcance
- 1.3. Definiciones, Acrónimos y Abreviaturas

## **2. Descripción General**

- 2.1. Perspectiva del Producto
- 2.2. Funciones del Producto
- 2.3. Características de los Usuarios
- 2.4. Restricciones
- 2.5. Suposiciones y Dependencias

## **3. Requisitos Específicos**

- 3.1. Interfaces Externas
- 3.2. Funciones
  - 3.2.1 Empleados
  - 3.2.2 Departamentos
- 3.3. Requisitos de Rendimiento
- 3.4. Restricciones de Diseño
- 3.5. Atributos del Sistema

## **4. Apéndices**

# 1. Introducción

## 1.1. Propósito

El presente documento de Especificación de Requisitos Software tiene como propósito definir de manera clara y precisa las funcionalidades y restricciones que tendrá la aplicación que se desea construir.

## 1.2. Alcance

El documento cubrirá la especificación de empleados y departamento, que consiste en la implementación de un software de gestión administrativa. Se detallará cualquier transacción derivada de la gestión de cualquiera de los módulos.

Los datos estarán disponibles a través de la propia interfaz, pudiendo en cualquier momento mostrar la información que se precise. Dicha interfaz, será sencilla y de fácil manejo.

## 1.3. Definiciones, Acrónimos y Abreviaturas

**SRS:** Especificación de Requisitos Software (del inglés Software Requirements Specification); Este mismo documento es una SRS.

**IS:** Ingeniería del Software; el conjunto de buenas prácticas y hábitos en el desarrollo de productos software que permite garantizar la calidad y mantenibilidad del trabajo.

**IEEE:** Institute of Electrical and Electronics Engineers y, por metonimia, el conjunto de estándares que dicha organización redacta; Se trata de una de las referencias más importantes en la práctica de la IS.

**BD:** Base de datos; El sistema que gestionará la información que manejará la aplicación.

**SQA:** Software Quality Assurance; el conjunto de prácticas de IS centradas en la supervisión del trabajo de diseño e implementación. Suele ser llevado a cabo por un equipo independiente que no estará en este proyecto por falta de personal.

**IU:** Interfaz de Usuario; se trata de la parte visual de la aplicación con la que interactúa el usuario final.

## 2. Descripción General

### 2.1. Perspectiva del Producto

Se pretende implementar una aplicación, que gestione administrativamente la información relativa a sus empleados y departamentos en los que trabajan.

### 2.2. Funciones del Producto

Para cada módulo se van a implementar las siguientes funciones:

#### • Empleados

Habrà un subsistema que se encargará de la gestión de los empleados, así como su alta y baja del sistema. Se les identificarà por el DNI y habrá dos tipos de empleados: empleados a tiempo completo y empleados a tiempo parcial.

##### Casos de uso:

- Alta empleado a tiempo completo.
- Alta empleado a tiempo parcial.
- Eliminar empleados.
- Calculo nómina tiempo completo.
- Calculo nómina tiempo parcial.

#### • Departamentos

Dividirá el trabajo de los empleados según su función. Cada departamento contiene un nombre para su identificación y una lista de empleados.

##### Casos de uso:

- Alta departamento.
- Baja departamento.
- ModificarDepartamento
- Consultar departamento.
- Listar departamentos.
- Calcular nómina departamentos.

## 2.3. Características de los Usuarios

El único usuario será el **administrador (Usuario)**. El cual tendrá todos los privilegios sobre el software y gestionará las características del mismo.

## 2.4. Restricciones

- La Arquitectura será Multicapa.
- La aplicación será de escritorio.
- Se usará el software IBM RSA para el desarrollo de diagramas de Casos de Uso, de actividad, de secuencia, de clase.
- El código será escrito en JAVA y se usaran patrones de diseño previamente seleccionados.
- Para la gestión de la base de datos se usará Sqlite.

## 2.5. Suposiciones y Dependencias

Se considera que la suposición de que estos requisitos son estables, no se producirá ningún cambio a petición de este, no se añadirán nuevas funcionalidades.

## 3. Requisitos Específicos

### 3.1. Interfaces Externos

#### • Interfaz Gráfica

La interfaz gráfica con la que el usuario final interactúa deberá ser intuitiva de manera que, sin un manual de uso, el usuario identifique rápidamente los componentes y las secciones del sistema.

Las interfaces de usuario están relacionadas con los módulos descritos en la SRS. Para modificar los datos se deberá realizar por medio del teclado y el Mouse (ratón).

La interfaz estará formada por dos niveles de pantallas. El primer nivel contendrá una sola ventana de empleados donde aparecerá todas las posibles operaciones a realizar. El segundo nivel lo formarán la ventana de departamentos que contendrá las operaciones relacionadas con este módulo.

**La interfaz gráfica** seguirá el mismo tratamiento que la de usuario, de forma que esta ayude al usuario a manejar la aplicación sin necesidad de ningún tipo de ayuda externa como, por ejemplo, un manual de instrucciones.

El sistema se basará en un control de los empleados agrupados en departamentos.

Las siguientes pantallas son diferentes *mock up* (prototipo de pantalla) iniciales de la interfaz de usuario que tendrá la aplicación final.

#### • Interfaz hardware

Será necesario disponer de equipos de cómputo en un buen estado con las siguientes características:

- Adaptadores de red
- Teclado
- Ratón
- Impresora
- Memoria mínima interna de 1Gb

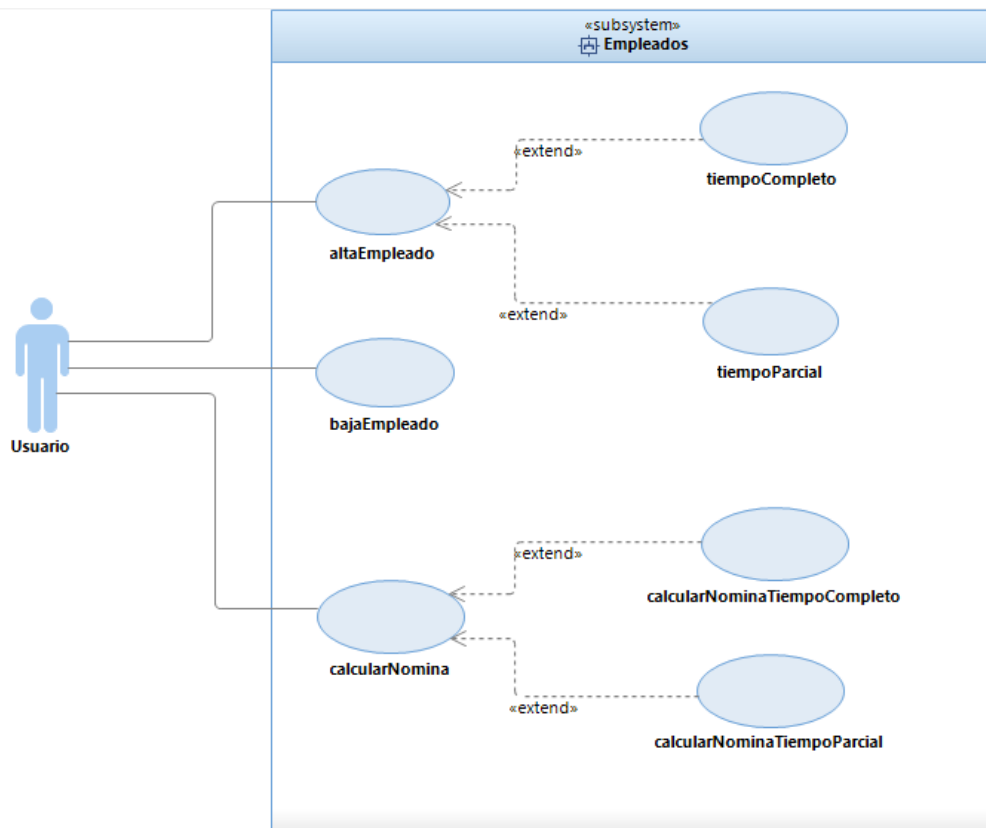
#### • Interfaz software

- Sistema operativo: Windows 7 o superior, distribución GNU/Linux compatible con

- Java 8 SE y MAC OS X Mountain Lion y superiores.  
- Aplicaciones adicionales: JAVA 8 SE.

## 3.2. Funciones

### 3.2.1 Empleados





## Caso de uso: Alta Empleado a Tiempo Completo

**Identificador:** EMP-1

**Objetivo en Contexto:** Dar de alta a un nuevo empleado a tiempo completo con todos los datos recibidos por la entrada, asignándole un departamento activo. Si el empleado existiera de antes, se activaría de nuevo.

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** Nombre, Apellidos, DNI, Departamento al que pertenecerá, Salario.

**Éxito:** Crea el empleado, guardando todos sus campos en la Base de Datos y permitiendo que el departamento lo pueda manejar.

**Fallo:** No se crea el empleado.

**Salida Datos:**

**Flujo principal**

1. El usuario pulsa en crear empleado.
2. El usuario selecciona a tiempo completo.
3. El usuario introduce el DNI del nuevo empleado.
4. El sistema comprueba que el formato del DNI es correcto.
5. El sistema comprueba que el DNI no esté en la BBDD.
6. El usuario introduce el resto de los datos.
7. El sistema introduce al empleado en la BBDD, asignándole un ID.

**Flujos secundarios**

1. a) El usuario pulsa el botón de cancelar, con lo que no se inicia la entrada de datos, no creándose el empleado, terminando el caso de uso.
4. a) Si no es correcto, se muestra un mensaje con el error y se vuelve a pedir el DNI.  
(*Pantalla: Mensaje de Error datos*).
- 5.a) Si el empleado ya existe, porque su DNI ya está registrado y activo, se muestra un mensaje de error y se vuelve al punto 3.
- 5.b) Si el empleado ya existe, porque su DNI ya está registrado e inactivo, se reactivará y mostrará un mensaje de activación.

## Caso de uso: Alta Empleado a Tiempo Parcial

**Identificador:** EMP-2

**Objetivo en Contexto:** Dar de alta a un nuevo empleado a tiempo parcial con todos los datos recibidos por la entrada, asignándole un departamento activo. Si el empleado existiera de antes, se activaría de nuevo. Si el empleado existiera de antes, se activaría de nuevo.

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** Nombre, Apellidos, DNI, Departamento al que pertenecerá, Salario por hora trabajada, número de horas trabajadas.

**Éxito:** Crea el empleado, guardando todos sus campos en la Base de Datos y permitiendo que el departamento lo pueda manejar.

**Fallo:** No se crea el empleado.

**Salida Datos:** Se le adjudica un ID al empleado.

### **Flujo principal**

1. El usuario pulsa en crear empleado.
2. El usuario selecciona a tiempo parcial.
3. El usuario introduce el DNI del nuevo empleado.
4. El sistema comprueba que el formato del DNI es correcto.
5. El sistema comprueba que el DNI no esté en la BBDD.
6. El usuario introduce el resto de datos.
7. El sistema introduce al empleado en la BBDD, asignándole un ID.

### **Flujos secundarios**

- 1.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la entrada de datos, no creándose el empleado, terminando el caso de uso con la post-condición de fallo.
- 4.a) Si no es correcto, se muestra un mensaje con el error y se vuelve a pedir el DNI.  
(Pantalla: *Mensaje de Error datos*).
- 5.a) Si el empleado ya existe, porque su DNI ya está registrado, se muestra un mensaje de error y se vuelve al punto 3.
- 5.b) Si el empleado ya existe, porque su DNI ya está registrado e inactivo, se reactivará y mostrará un mensaje de activación.

## **Caso de uso: Baja Empleado**

**Identificador:** EMP-3

**Objetivo en Contexto:** realiza la baja lógica del empleado seleccionado, que puede ser realizada por ID empleado o DNI.

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** ID.

**Éxito:** Elimina todos los datos del empleado.

**Fallo:** No se elimina al empleado.

**Salida Datos:** Muestra los datos.

### **Flujo principal**

1. El usuario pulsa en eliminar empleado.
2. El usuario introduce el ID del empleado a eliminar.
3. El sistema comprueba que el ID está en la BBDD.
4. El sistema elimina el empleado.

### **Flujos secundarios**

1.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la eliminación del empleado, terminando el caso de uso con la post-condición de fallo.

3.a) El empleado a eliminar no existe, porque el ID introducido no está en la BBDD, se muestra un mensaje de error y se vuelve al punto 2.

## Caso de uso: Calcular Nomina Empleado a Tiempo Completo

**Identificador:** EMP-4

**Objetivo en Contexto:** Calcular el montante total del salario que tiene que percibir el empleado a tiempo completo.

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** ID del empleado a calcular su nómina.

**Éxito:** Calcula la cantidad y la muestra.

**Fallo:** No se calcula la nómina.

**Salida Datos:** Cantidad total de salario del empleado.

### **Flujo principal**

1. El usuario pulsa en calcular nómina de un empleado a tiempo parcial.
2. El usuario introduce el ID del empleado.
3. El sistema comprueba que el dato es correcto (el ID existe en la BBDD y el empleado es a tiempo completo).
4. El sistema resta el salario del empleado con el cálculo del impuesto según su salario.

**Nomina = salario – (impuesto/100 \* salario)**

### **Flujos secundarios**

- 2.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la entrada de datos, no creándose el empleado, terminando el caso de uso con la post- condición de fallo.
- 3.a) Si algún dato no es correcto, se muestra un mensaje con el error y se vuelve al paso 2. (*Pantalla: Mensaje de Error datos*).

## Caso de uso: Calcular Nomina Empleado a Tiempo Parcial

**Identificador:** EMP-5

**Objetivo en Contexto:** Calcular el montante total del salario que tiene que percibir el empleado a tiempo parcial por las horas que ha trabajado.

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** ID del empleado a calcular su nómina.

**Éxito:** Calcula la cantidad y la muestra.

**Fallo:** No se calcula la nómina.

**Salida Datos:** Cantidad total de salario del empleado.

### **Flujo principal**

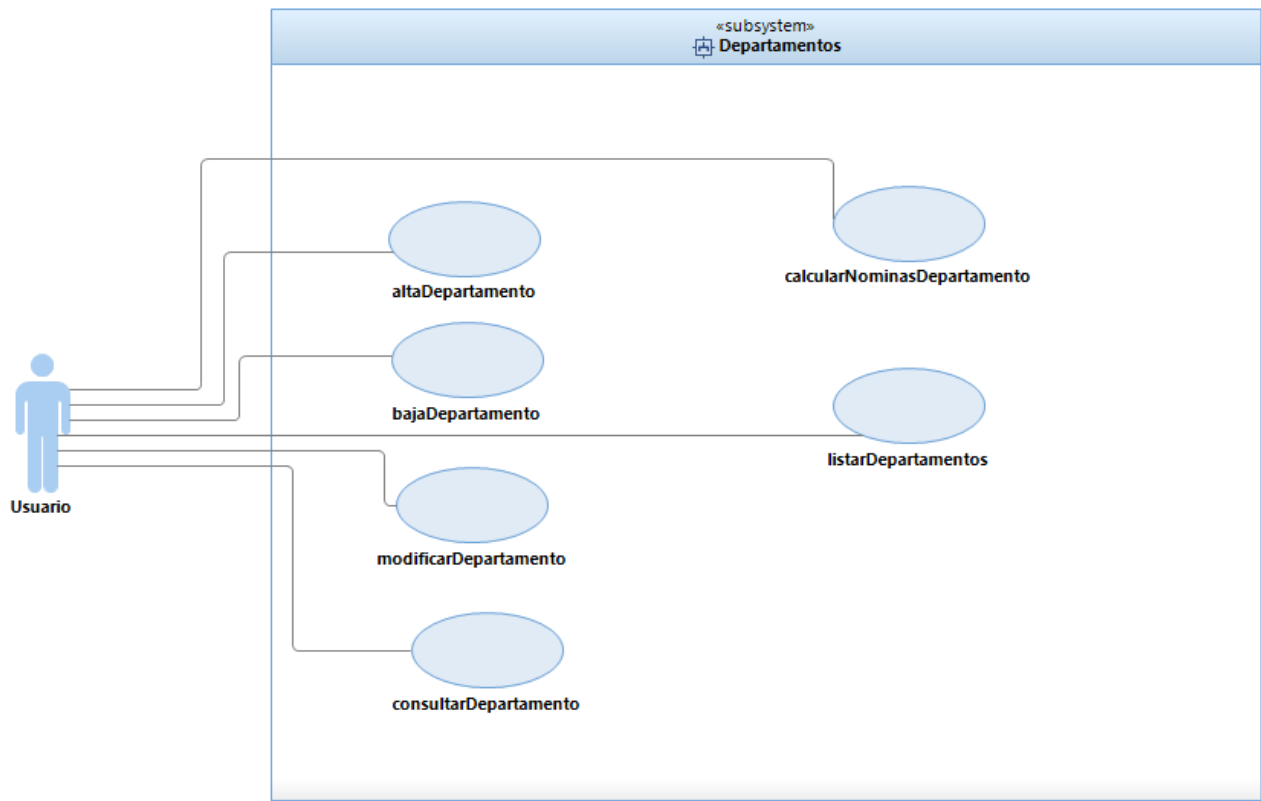
5. El usuario pulsa en calcular nómina de un empleado a tiempo parcial.
6. El usuario introduce el ID del empleado.
7. El sistema comprueba que el dato es correcto (el ID existe en la BBDD y el empleado es a tiempo parcial).
8. El sistema multiplica el salario del empleado por las horas trabajadas.

**Nomina = horas\*salarioPorHora**

### **Flujos secundarios**

- 2.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la entrada de datos, no creándose el empleado, terminando el caso de uso con la post- condición de fallo.
- 3.a) Si algún dato no es correcto, se muestra un mensaje con el error y se vuelve al paso 2. (*Pantalla: Mensaje de Error datos*).

### 3.2.2 Departamentos



## Caso de uso: Alta departamento

**Identificador:** DEP-1

**Objetivo en Contexto:** Dar de alta un nuevo departamento en el sistema con todos los datos recibidos por la entrada.

**Actor principal:** Administrador.

**Actores secundarios:** Bases de datos.

**Entrada Datos:** Nombre del departamento.

**Éxito:** Crea el departamento, guardando sus datos en la Base de Datos.

**Fallo:** No se crea el departamento y se informa al usuario.

**Salida Datos:** Se muestra una lista de departamentos con el nuevo incluido.

### **Flujo principal**

1. El administrador pulsa en Agregar departamento.
2. El administrador introduce todos los campos menos el ID de los datos de entrada
3. El sistema comprueba que los datos son correctos.
4. El sistema comprueba que no esté en la BBDD.
5. El sistema introduce al departamento en la BBDD, asignándole un ID automáticamente y un atributo "Activo" a true.

### **Flujos secundarios**

- 2.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la entrada de datos, no creándose el departamento, terminando el caso de uso con la post-condición de fallo.
- 4.a) Si el departamento ya está creado, se comprueba que no esté de baja. Si está de baja se da la opción de dar de alta, con lo que si el usuario acepta termina el caso de uso. En el caso en el que el departamento ya haya estado de alta se muestra un mensaje de error y se vuelve al paso 2.

## Caso de uso: Baja departamento

**Identificador:** DEP-2

**Objetivo en Contexto:** Da de baja (lógica) el departamento que el usuario indique.

**Actor principal:** Administrador. **Actores secundarios:** Base de datos. **Entrada Datos:** ID del departamento.

**Éxito:** Se da de baja el departamento.

**Fallo:** El departamento no se da de baja.

**Salida Datos:** Se muestra una lista de los departamentos activos restantes.

### **Flujo principal**

1. El administrador pulsa en Baja departamento.
2. El administrador introduce el ID del departamento a dar de baja.
3. El sistema comprueba que el ID está en la BBDD.
4. El sistema comprueba que el departamento esté activo.
5. El sistema comprueba que el departamento no tiene empleados.
6. El sistema da de baja el departamento.

### **Flujos secundarios**

- 1.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la acción, terminando el caso de uso con la post-condición de fallo.
- 3.a) El departamento no existe porque el nombre introducido no existe en la Base de Datos, se muestra un mensaje de error y se vuelve al punto 2.
- 4.a) El departamento está inactivo. Se muestra un mensaje por pantalla y termina la ejecución.
  - 5.a) El departamento tiene algún empleado. Se muestra un mensaje de error y se vuelve al punto 2.



## Caso de uso: Consultar departamento

**Identificador:** DEP-3

**Objetivo en Contexto:** Muestra los datos del departamento, así como la lista de empleados que pertenecen al mismo.

**Actor principal:** Administrador.

**Actores secundarios:** Base de datos.

**Entrada Datos:** ID del departamento.

**Éxito:** Muestra los datos del departamento.

**Fallo:** No se muestra nada.

**Salida Datos:** Se muestran los datos del departamento.

### **Flujo principal**

1. El administrador pulsa en Consultar departamento.
2. El administrador introduce el ID del departamento.
3. El sistema comprueba que el departamento está en la BBDD.
4. El sistema muestra los datos del departamento.

### **Flujos secundarios**

- 1.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la consulta del departamento, terminando el caso de uso con la post-condición de fallo.
- 3.a) Si el departamento no existe, porque el ID introducido no está en la Base de Datos, se muestra un mensaje con el error y se vuelve al punto 2.

## Caso de uso: Modificar departamento

**Identificador:** DEP-4

**Objetivo en Contexto:** Modificar todos o algún dato de un departamento seleccionado

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** ID del departamento.

**Éxito 1:** Modifica el nombre del departamento.

**Éxito 2:** Añade un empleado nuevo en la lista de empleados del departamento.

**Éxito 3:** Se elimina un empleado en la lista de empleados del departamento.

**Fallo:** No se hace ninguna modificación.

**Salida Datos:** Muestra los datos del departamento con los cambios.

### **Flujo principal**

1. El administrador pulsa en modificar departamento.
2. El administrador introduce el ID del departamento a modificar.
3. El sistema comprueba que el departamento existe en la Base de Datos.
4. El administrador escoge los datos que quiere modificar. Tiene tres opciones: cambiar el nombre del departamento, añadir un empleado o eliminar un empleado (éstas últimas siempre del departamento, no de la Base de Datos de empleados).
5. El sistema modifica el departamento.
  - a. El administrador escoge cambiar el nombre.
    - i. El administrador introduce el nuevo nombre del departamento.
    - ii. El sistema comprueba que no haya ya creado un departamento con el mismo nombre en la Base de Datos.
    - iii. El sistema modifica el departamento
  - b. El administrador escoge añadir un empleado al departamento.
    - i. El administrador introduce el DNI del empleado.
    - ii. El sistema comprueba que el empleado está en la Base de Datos.
    - iii. El sistema introduce el empleado en la Lista de empleados del departamento.
  - c. El administrador escoge eliminar un empleado del departamento
    - i. El administrador introduce el DNI del empleado.
    - ii. El sistema comprueba que el empleado está en la Lista de empleados del departamento.
    - iii. El sistema elimina el empleado de la Lista de empleados del departamento

### **Flujos secundarios**

- 1.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la modificación del departamento, terminando el caso de uso con la poscondición de fallo
- 2.a) El departamento consultado no existe porque el nombre introducido no está en la BBDD, se muestra un mensaje de error y se vuelve al punto 2
- 4.1.2a) El nombre del departamento ya está siendo usado por otro departamento. Se muestra un mensaje de error y se vuelve al punto 4.1.1.
- 4.2.2a) El empleado no está en la base de datos. Se muestra un mensaje de error y se da la opción de volver al punto 4.2.1 y de añadir el nuevo empleado a la Base de Datos haciendo uso del caso de uso Añadir Empleado.
- 4.3.2a) El empleado ya no está en la lista de empleados del departamento. Se muestra un mensaje de error y se vuelve al punto 4.3.1.

## Caso de uso: Listar departamentos

**Identificador:** DEP-5

**Objetivo en Contexto:** Muestra todos los departamentos.

**Actor principal:** Administrador.

**Actores secundarios:** Base datos.

**Entrada Datos:**

**Éxito:** Muestra una lista de los departamentos.

**Fallo:** No se muestra nada.

**Salida Datos:** Se muestra una lista de los departamentos.

**Flujo principal**

1. El administrador pulsa en Listar departamentos.
2. El sistema adquiere una lista de departamentos de la BBDD.
3. El sistema muestra la lista de departamentos.

**Flujos secundarios**

- 1.a) El usuario pulsa el botón de cancelar, terminando el caso de uso con la post-condición de fallo.
- 2.a) La BBDD no contiene ningún departamento, se muestra un mensaje de error y se termina el caso de uso con la post-condición de fallo.

## Caso de uso: Calcular Nóminas Departamento

**Identificador:** DEP-7

**Objetivo en Contexto:** Calcular el montante total del salario que tienen que percibir todos los empleados que pertenecen al departamento.

**Actor principal:** Administrador.

**Actores secundarios:** Base de Datos.

**Entrada Datos:** ID del departamento a calcular el montante de las nóminas.

**Éxito:** Calcula la cantidad y la muestra.

**Fallo:** No se calcula la nómina.

**Salida Datos:** Cantidad total de salarios del departamento.

### **Flujo principal**

1. El administrador pulsa en calcular nóminas de un departamento.
2. El administrador introduce el ID del departamento.
3. El sistema comprueba que el departamento existe en la base de datos y que cuenta con al menos un empleado.
4. El sistema multiplica el salario de cada empleado por sus horas trabajadas, sumando todas las cantidades totales de salario de cada empleado.

### **Flujos secundarios**

- 1.a) El usuario pulsa el botón de cancelar, con lo que no se inicia la entrada de datos, terminando el caso de uso con la post-condición de fallo.
- 3.a) Si algún dato no es correcto, se muestra un mensaje con el error y se vuelve a pedir los datos.

### **3.3. Requisitos de Rendimiento**

Solo podrá haber un tipo de usuario conectado al mismo tiempo al software.

Para el correcto funcionamiento de este software, tenemos que garantizar que la ejecución de consultas no afecte negativamente al rendimiento, pues la BBDD tiene que estar siempre operativa, las transacciones con la BBDD se harán de una en una para evitar posibles conflictos, usando el patrón de transacciones para mantener una orden en la ejecución que no afecte al rendimiento.

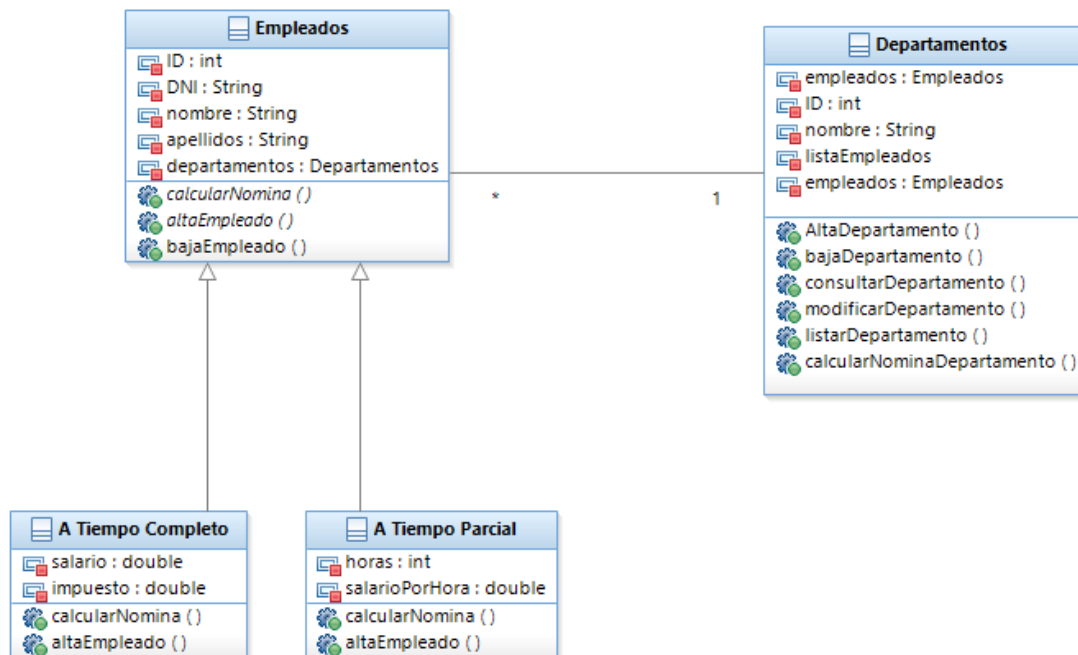
### 3.4. Restricciones de Diseño

El modelo UML 2.x de la aplicación, será en formato IBM Rational Software Architect 9.5.

- El lenguaje de implementación sera Java.
- La aplicación será de escritorio
- La persistencia de los datos debe hacerse en formato relacional
- La arquitectura de la aplicación será multicapa.
- Deben aplicarse los siguientes patrones obligatoriamente en la versión de la aplicación:
  - o Service to worker
  - o Transfer object
  - o Transfer Object Assembler
  - o Application Service
  - o Data Access Object
  - o Alguna query tal y como se vieron en clase.
- Además de estos (salvo Data Access Object y queries), deben aplicarse los siguientes patrones obligatoriamente en la segunda versión de la aplicación:
  - o Business Object
  - o Domain Store (implementado con JPA 2.x. Se recomienda usar EclipseLink 2.x).

#### Parte JPA

#### Modelo de Negocio:



### 3.5. Atributos del Sistema

Para cumplir con el estándar que se está siguiendo en la creación del programa a través de este documento, el sistema tiene que cumplir las siguientes características:

- **Fiabilidad:** Probabilidad de que un sistema, aparato o dispositivo cumpla una determinada función bajo ciertas condiciones durante un tiempo determinado.
- **Usabilidad:** Calidad del programa informático que son sencillos de usar porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.
- **Accesibilidad:** Posibilidad de acceder a cierta cosa o facilidad para hacerlo.
- **Mantenibilidad:** La **mantenibilidad** es la propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez se ha presentado un evento de falla.
- **Seguridad:** Ausencia de riesgo.
- **Extensibilidad:** Factor de calidad del software que consiste en la facilidad de adaptación del software a nuevos requisitos o cambios en la especificación.