

Modelado de Software

Titulación: Grado en Ingeniería del Software

Entrega Memoria Aplicación *“HOTEL MANAGER”*

Curso 2013-2014

Profesor: Antonio Navarro Martín

Participantes:

Álvarez Piñeiro, Emilio

Pérez Latorre, Álvaro

Pérez Valbuena, Juan Luis

Quesada Pimentel, Álvaro,

Serrano Torres, Daniel

Zabala Hidalgo, Alejandro

Indice

[1. Introducción a Hotel Manager 3](#_Toc374549886)

[2. Patrones Aplicados 3](#_Toc374549887)

[2.1 Integración 3](#_Toc374549888)

[2.2 Negocio 3](#_Toc374549889)

[2.3 Presentación 3](#_Toc374549890)

[3. Diagramas de aplicación 4](#_Toc374549891)

[3.1 Diagramas de Casos de Uso: 4](#_Toc374549892)

[3.2 Diagramas de Actividad: 5](#_Toc374549893)

[3.3 Diagramas de Secuencia: 5](#_Toc374549894)

[4. Base de datos relacional (MySQL) 6](#_Toc374549895)

[5. Gestión de Concurrencia 7](#_Toc374549896)

[6. Testing 7](#_Toc374549897)

# Introducción a Hotel Manager

*‘Hotel Manager’* es un Software de gestión hotelera realizada usando el Proceso Unificado de Rational por el grupo **‘*BSoD Software’*** de la Universidad Complutense de Madrid.

Se trata de una aplicación multicapa, realizada íntegramente en Java.

Esta entrega es la segunda y última realizada para la asignatura de *‘Modelado del Software’* del Grado de Ingeniería del Software.

Se añaden solo los cambios significativos con respecto a la anterior entrega. Para poder consultar cambios de la primera entrega, referirse a la Memoria de dicha entrega.

# Patrones Aplicados

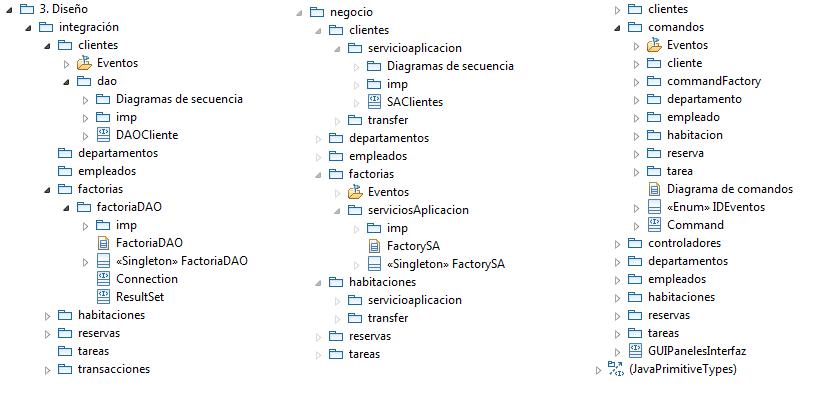
## Negocio

* **Patrón Objeto de Negocio:** *(Business Object*): Empleados, Tarea, Departamento.
* **Patrón Domain Store :** Servicios de Aplicación Empleados , Tarea y Departamento

# Diagramas de aplicación

Se ha decidido utilizar la herramienta CASE ‘*IBM Rational Software Architect 8.0.3’* para la realización de los diagramas necesarios para el desarrollo de esta aplicación.

El modelo se estructura acorde a la arquitectura multicapa. El primer nivel de paquetes se corresponde con las capas de integración, negocio y presentación en las que se divide la aplicación. Dentro de estos paquetes se encuentran los diagramas y objetos organizados por las entidades de la aplicación (Clientes, habitaciones y reservas). Aquellos objetos que hacen que hacen referencia a varias entidades, cómo las factorías y los controladores se encuentran incluidos en sus propios paquetes, y en casos especiales, dentro de los mismos se encuentra otra organización por entidades.



En concreto, los diagramas realizados son:

## Diagramas de Casos de Uso:

* + Empleado:
    - Alta Empleado
    - Consulta Empleado
    - Modificar Empleado
    - Consulta Todos los Empleados
    - Dar de Baja Empleado
    - Dar de Alta Tarea Empleado
    - Dar de Baja Tarea Empleado
  + Tarea:
    - Alta Tarea
    - Modificar Tarea
    - Consultar Tarea
    - Consultar Lista de Tareas
    - Dar de Baja Tarea
  + Departamento:
    - Dar de Alta Departamento
    - Dar de Baja Departamento
    - Modificar Departamento
    - Consultar Departamento
    - Consultar todas las Departamento

## Diagramas de Actividad:

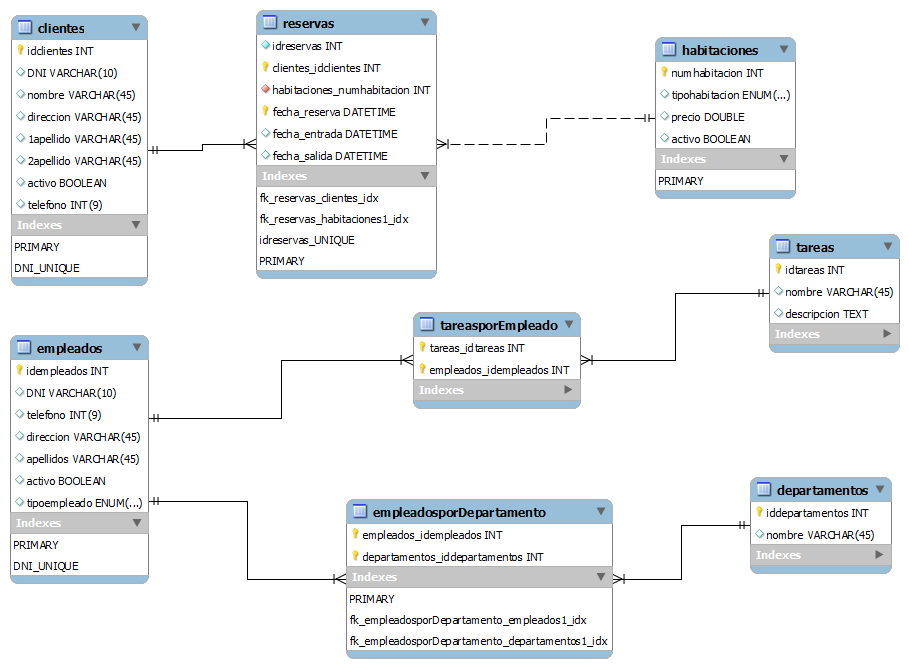
* + Empleado:
    - Alta Empleado
    - Consulta Empleado
    - Modificar Empleado
    - Consulta Todos los Empleados
    - Dar de Baja Empleado
    - Dar de Alta Tarea Empleado
    - Dar de Baja Tarea Empleado
  + Tarea:
    - Alta Tarea
    - Modificar Tarea
    - Consultar Tarea
    - Consultar Lista de Tareas
    - Dar de Baja Tarea
  + Departamento:
    - Dar de Alta Departamento
    - Dar de Baja Departamento
    - Modificar Departamento
    - Consultar Departamento
    - Consultar todas las Departamento

## Diagramas de Secuencia:

Se eligió un módulo y se realizaron los diagramas correspondientes. Se han realizado además los diagramas de las operaciones de otros módulos que entrañan mayor dificultad o complejidad.

# Base de datos relacional (MySQL)

Para la persistencia de los datos, se utilizado una base de datos relacional *‘MySQL Server 5.5’* en remoto, lo cualpermite el uso de esta aplicación desde cualquier computador con acceso a Internet. Se ha realizado el siguiente diseño usando *‘MySQL WorkBench 6.0’.*

**

# Gestión de Concurrencia

Se ha decidido utilizar una gestión de **concurrencia pesimista**. Se ha utilizado lo siguiente:

* Desactivar el *‘AUTOCOMMIT’* de nuestra base de datos. Con ello solo persistirán los cambios si se ejecuta posteriormente un *‘COMMIT’*. En caso contrario si queremos descartar dichos cambios se ejecutará un *‘ROLLBACK’.*
* Se ha añadido un nivel aislamiento de la base de datos ‘REPETABLE\_READ’.
* Se han añadido a las sentencias SQL de tipo SELECT la cláusula FOR UPDATE para leer el dato más actualizado disponible, estableciendo bloqueos exclusivos sobre cada fila leída. Es decir, el mismo bloqueo que haría UPDATE.

# Testing

Se ha implementado una suite de test de JUnit que prueba los casos de uso de forma general, probando que las salidas son iguales a las esperadas y que no se producen errores a la hora de inicializar los datos.

