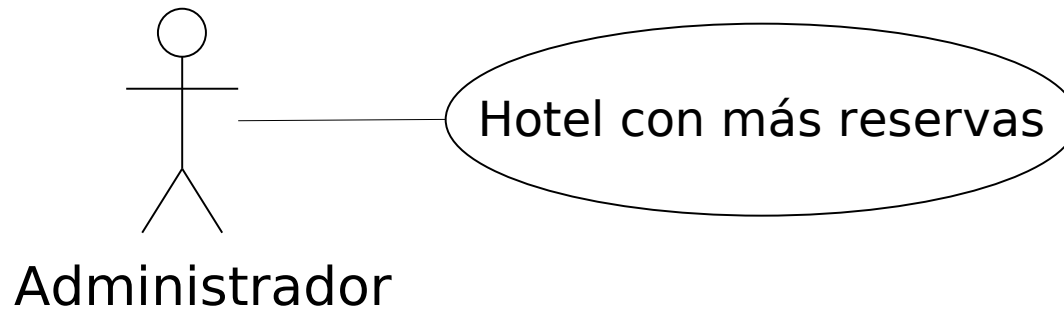


## Solución Examen Junio 2008

- Hotel más reservado (1h 20 min.)
  - Análisis (1,5 puntos):
    - Diagrama Secuencia Sistema + Contratos
  - Diseño (2,5 puntos):
    - Diagramas de Secuencia



**Caso de uso:** Hotel con más reservas

**Actores:** Administrador

**Descripción:** El administrador desea localizar los hoteles cuyos clientes han realizado más reservas en un periodo de tiempo, calculando el número medio de reservas por excursión.

**Escenario principal:**

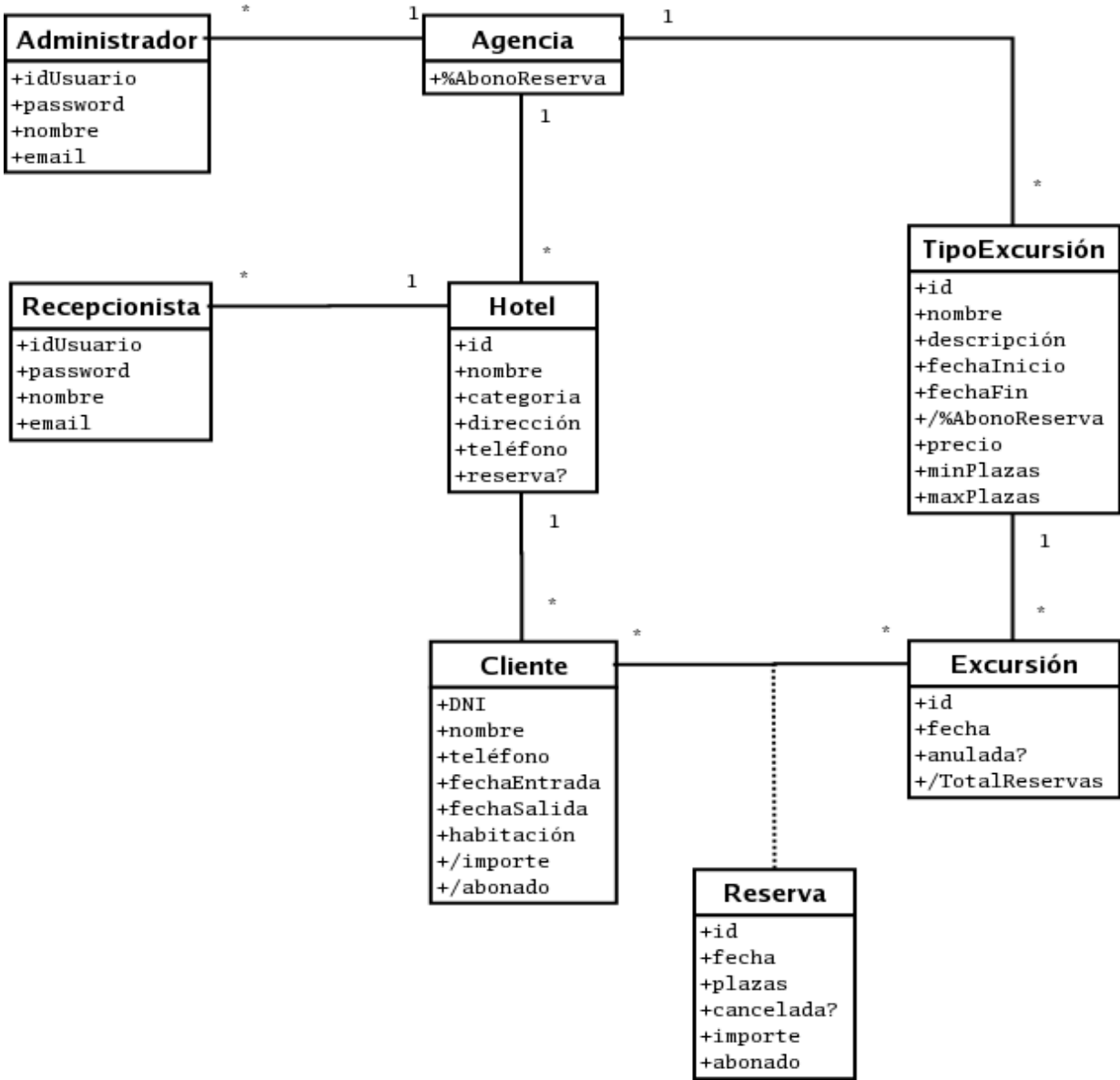
1.Administrador: El administrador introduce la fecha inicial y final del periodo.

2.Sistema: Presenta los hoteles con más reservas ordenados descendientemente por número de reservas totales realizadas en dicho periodo. Para cada hotel, se mostrará su identificador, nombre, categoría, número total de reservas, media de reservas por excursión e importe total de las reservas realizadas.

*El Administrador puede repetir los pasos 3 a 4 un número indeterminado de veces.*

3.Administrador: Selecciona un hotel.

4.Sistema: Muestra para cada día del periodo seleccionado, las reservas realizadas por dicho hotel y el importe total de las reservas.





## Contrato operación ObtenerHoteles

- **Name:** ObtenerHoteles(fechal, fechaF) : ListaInfoHoteles
- **Responsabilities**

Dada una fecha inicial y final de un periodo, obtener los hoteles ordenados descendientemente por número de reservas totales en dicho periodo
- **Preconditions**

fechal y fechaF són válidas  
fechal  $\leq$  fechaF
- **Postconditions**
- **Salida**

ListaInfoHoteles = Lista(<idHotel, nombre, categoria, NTR, MRE, IT>)  
donde  
NTR = Número Total de Reservas  
MRE = Media de Reservas por Excursión  
IT = Importe Total de las Reservas Realizadas

## Contrato operación ObtenerReservasHotel

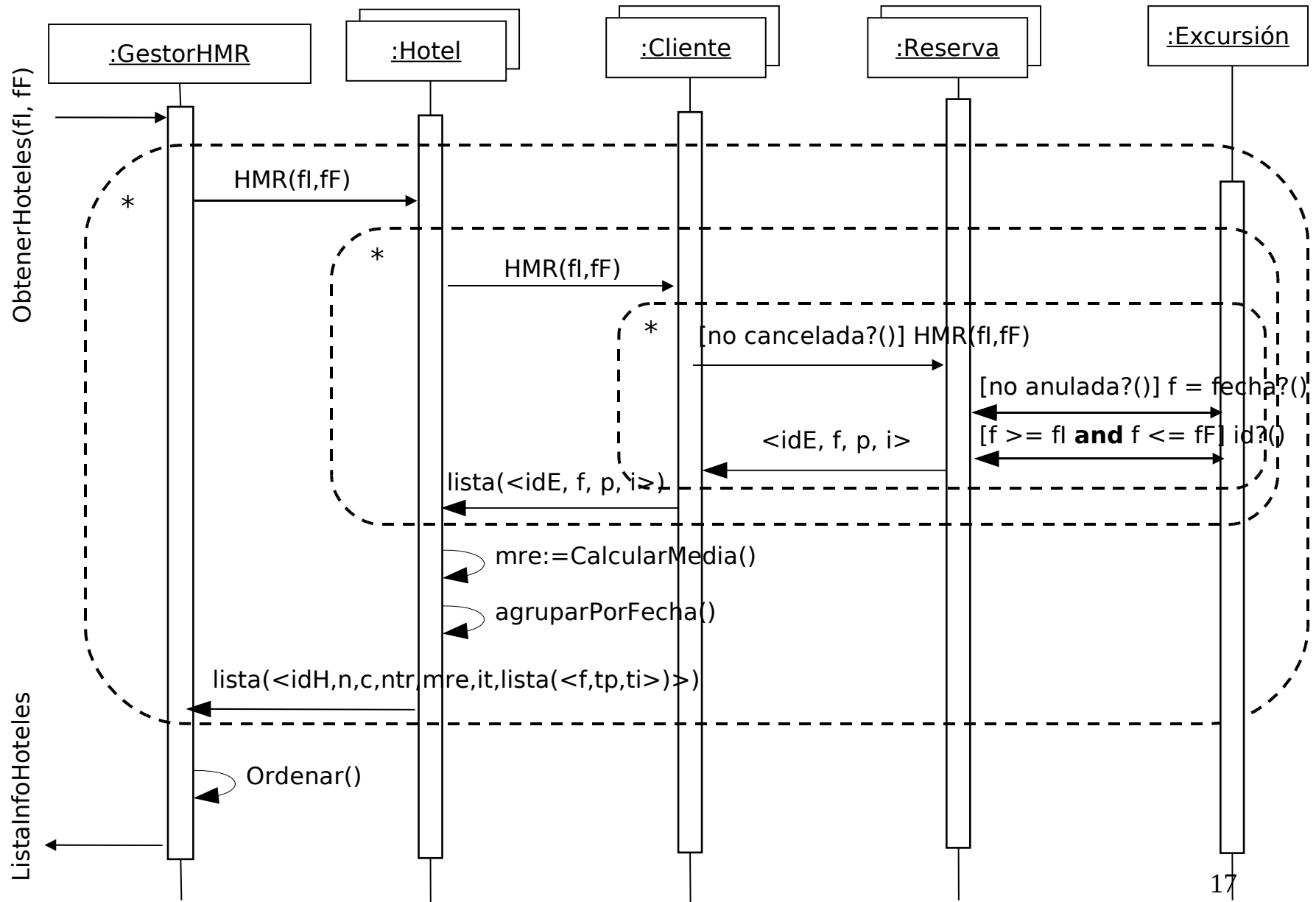
- **Name:** ObtenerReservasHotel(idHotel) : ListaInfoReservas
- **Responsabilities**

Dada un hotel, obtener para cada dia del periodo seleccionado, las reservas realizadas por dicho hotel y el importe total de las reservas
- **Preconditions**

fechaI y fechaF són válidas y conocidas por el sistema  
fechaI <= fechaF  
idHotel es válido
- **Postconditions**
- **Salida**

ListaInfoReservas = Lista(<fecha, NTR, IT>)  
donde  
NTR = Número Total de Reservas  
IT = Importe Total de las Reservas Realizadas

# Ingeniería del Software





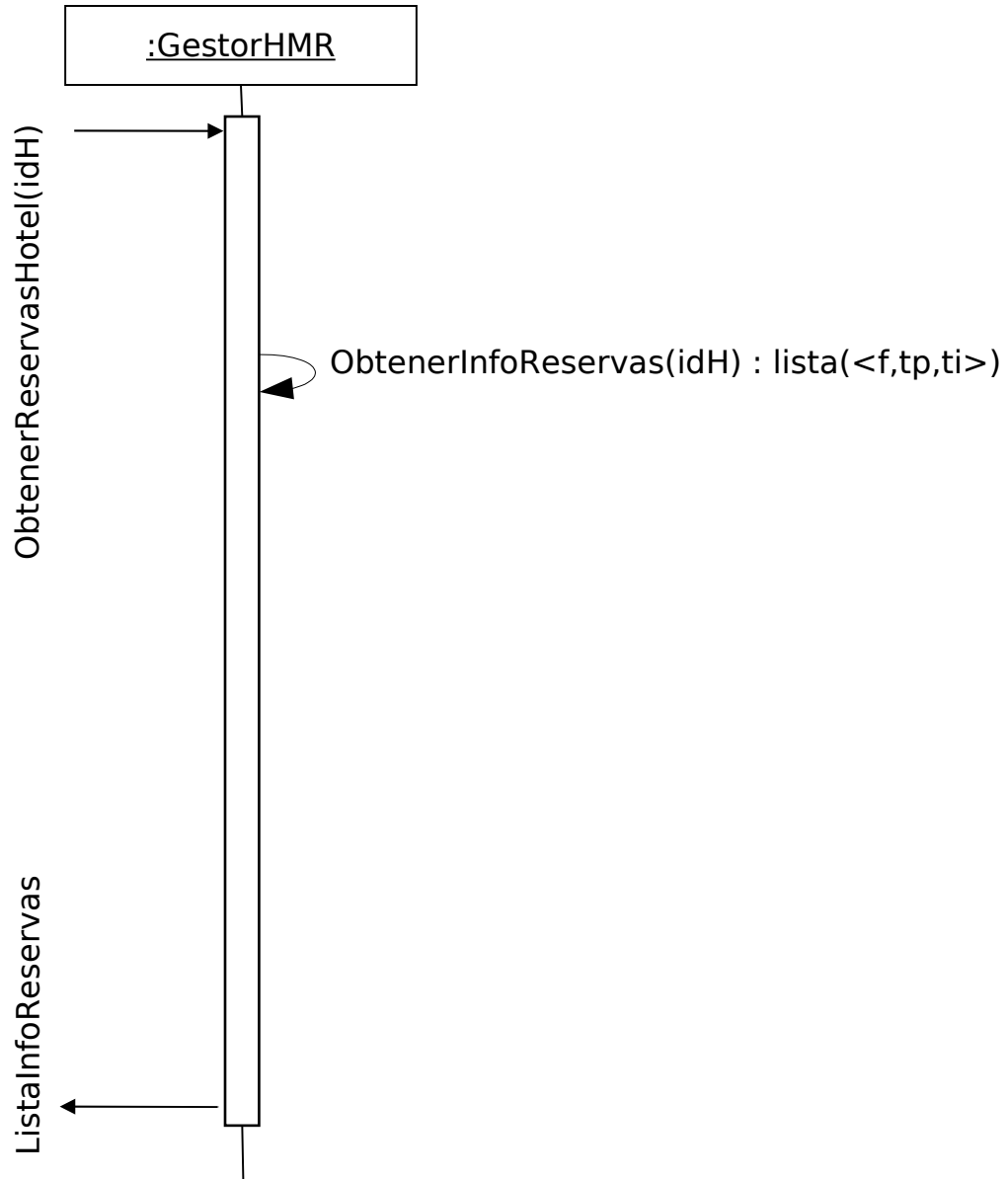
Escogemos el patrón controlador para gestionar el evento externo ObtenerHoteles. Aunque otras opciones son posibles, a falta de más información al tratarse de modelar un caso de uso, seleccionamos el controlador de caso de uso: GestorHMR. Además, esta clase artificial agrupa todos los hoteles y gestiona la estructura auxiliar ReservasHoteles. En esta estructura, el método ObtenerHoteles almacena para cada hotel y para cada día del periodo establecido, el total de reservas e importes. Esta estructura será consultada posteriormente por el evento externo del mismo caso de uso, ObtenerReservasHotel. Con ello pretendemos un diseño global eficiente con alta cohesión y bajo acoplamiento.

Por el patrón experto, el método HMR de Hotel obtiene para un periodo determinado por fl y fF, el identificador del Hotel, su nombre, su categoría, el número total de reservas (NTR), su media de reservas por excursión (MRE) y su importe total (IT). NTR se obtiene sumando todas las plazas de reservas realizadas por los clientes del hotel no canceladas, de excursiones no anuladas que hayan sido planificadas para una fecha del periodo. NRE se obtiene como el ratio entre el total de excursiones y sus plazas reservadas. IT se obtiene sumando los importes de las reservas. Los datos resultantes se ordenan de mayor a menor NTR.

Por el patrón experto, el método HMR de Cliente obtiene para un periodo determinado por fl y fF, la lista de todas las reservas no canceladas de ese cliente de excursiones no anuladas que hayan sido planificadas para una fecha del periodo. Para cada reserva se obtiene el identificador de la excursión, su fecha, el número de plazas reservadas y su importe.

Por el patrón experto, el método HMR de Reserva comprueba si la excursión no ha sido anulada por la agencia y si la fecha de la excursión pertenece al periodo determinado por fl y fF. En caso afirmativo, proporciona el identificador de la excursión, su fecha, el número de plazas reservadas y su importe.

Además, el método HMR de GestorHMR gestiona la estructura de datos ReservasHoteles para almacenar para cada hotel y día del periodo comprendido entre fl y fF, el total de plazas reservadas por el hotel y el importe total de las reservas.



Escogemos el patrón controlador para gestionar el evento externo `ObtenerReservasHotel`. Aunque otras opciones son posibles, a falta de más información al tratarse de modelar un caso de uso, seleccionamos el mismo el controlador de caso de uso que para la operación anterior: `GestorHMR`. Además, esta clase artificial agrupa también todos los Hoteles y gestiona la estructura auxiliar `ReservasHotel`. En esta estructura, el método `ObtenerHoteles` ha almacenado para cada hotel y para cada día del periodo establecido, el total de reservas e importes. Con ello pretendemos un diseño global eficiente con alta cohesión y bajo acoplamiento.

Por el patrón experto, la estructura `ReservasHotel` es consultada por el evento externo `ObtenerReservasHotel`. Dado un identificador de hotel, el método `ObtenerInfoReservas` del `GestorHMR` obtendrá el total de reservas e importes para cada día del periodo realizados por el hotel.